文章编号：2095-4980(2023)01-0095-07

利用通道剪枝技术的实时实例分割方法

宁　欣[1]，刘江宽[2,3]，李卫军[*1]，石　园[2,3]，支金林[2,3]，南方哲[4]

(1.中国科学院 半导体研究所，北京 100083；2.威富集团形象认知计算联合实验室，北京 102200；3.深圳市威富世界有限公司，广东 深圳 518102；4.新疆大学 软件学院，新疆 乌鲁木齐 830091)

摘　要：随着实例分割技术在各种场景中的应用越来越广泛，运行速度和硬件资源占用是该技术在应用中需要考虑的2个重要因素。最近提出的基于图像原型掩码系数的实例分割网络(YOLACT)在运行速度方面做得很好，但是需要设置较大的特征提取网络才能保证分割精确度，这就导致了模型占用的硬件资源较多，同时运行速度也受到了限制。在YOLACT的基础上，提出一种新的模型，对实例分割的特征提取网络进行了优化，先使用基于批量归一化层放缩因子的通道剪枝方法对YOLACT网络进行压缩，然后对压缩后的卷积层和批量归一化层进行融合，最后，在COCO val2017上对本文提出的方法进行了评估。实验结果表明，相比原始的YOLACT网络，该方法的模型文件大小可以减少56.9%，运行速度提升28.6%，运行时显存占用也降低了13.6%，有效地减少了硬件资源占用，并且提升了运行速度。

关键词：实例分割；模型压缩；通道剪枝；运行效率

中图分类号：TP391　　**文献标志码：**A　　**doi：**10.11805/TKYDA2020452

Real-time instance segmentation using channel pruning

NING Xin[1]，LIU Jiangkuan[2,3]，LI Weijun[*1]，SHI Yuan[2,3]，ZHI Jinlin[2,3]，NAN Fangzhe[4]

(1.Institute of Semiconductors，Chinese Academy of Sciences，Beijing 100083，China；2.Cognitive Computing Technology Joint Laboratory，WAVE Group，Beijing 102200，China；3.Shenzhen Wave Kingdom Co. LTD，Shenzhen Guangdong 518102，China；4.School of Software，Xinjiang University，Urumqi Xinjiang 830091，China)

Abstract：With the application of instance-segmentation in various scenarios, the running speed and the utilization of hardware resources are two important factors to be considered in the application of instance segmentation. Recently, a instance segmentation network named You Only Look At Coefficients (YOLACT) bears a high processing speed. However, YOLACT needs to set a large feature extraction network to ensure the segmentation accuracy, which leads to high resource occupancy and limited running speed. Based on the YOLACT, a new model is proposed, and the feature extraction of network segmentation is optimized. Firstly, a channel pruning method based on batch normalized scale factor is utilized to compress YOLACT network, then the convolution layer and batch normalization layer are fused. Finally, the proposed approach is evaluated on Common Objects in Context(COCO) val2017. The experimental results show that, the model size of the method can be reduced by 56.9% and the running speed can be improved by 28.6% compared with that of the original YOLACT network. This approach can reduce the hardware resource consumption and can improve the running speed.

Keywords：instance-segmentation；model-compression；channel-pruning；running efficiency

实例分割是计算机视觉中一项极具挑战性的任务，它结合了目标检测(object detection)和语义分割(semantic segmentation)，实例分割不仅需要检测图像中的不同个体，还要对不同个体进行像素级别的标注[1]。随着实例分割技术在实际场景中的应用越来越广泛，除了对分割精确度有较高的要求，对运行速度、模型文件大小、硬件资源占用等方面也提出了新的要求。

在卷积神经网络(Convolutional Neural Network，CNN)大规模应用在机器视觉任务之前，实例分割多采用经典的目标检测与语义分割方法进行串接实现。经典目标检测是基于图像特征加机器学习的模式，如Haar Like

收稿日期：2020-09-11；**修回日期：**2020-12-07

基金项目：国家自然科学基金资助项目(61901436)

通信作者：李卫军　email: wjli@semi.ac.cn.

Feature SVM[2]、HOG SVM[3]，语义分割多采用条件随机场的方法，如条件随机场(Conditional Random Field，CRF)[4]、马尔可夫随机场(Markov Random Field，MRF)[5]。传统方法实现的实例分割，在分割精确度和效率上都较低。

近年来，得益于卷积神经网络(CNN)检测技术的发展[6]，基于CNN的实例分割也得到了广泛的研究。相比经典的实例分割算法，基于CNN的实例分割方法表现出很好的性能，已经是目前实例分割的主流研究方向。但目前实例分割的工作更多注重于分割精确度的提升，对运行速度、硬件资源占用问题的关注相对较少[7]。

尽管在目标检测领域，已经有YOLO[8−9]、SSD(Single Shot MultiBox Detector)[10]等实时检测算法，在语义分割领域也有ERFNet[11]、ENet[12]、LEDNet[13]等网络可以做到实时语义分割，但在实例分割领域，对运行效率和硬件资源占用的研究还相对较少。

最近提出的实例分割网络YOLACT[14]考虑了分割的实时性需求，可以在单图形处理器(Graphics Processing Unit，GPU)上达到30 fps左右的处理速度，同时在COCO[15]验证集上的实例分割平均精确度(mean Average Precision，mAP)[15]为30.6，在实时性和分割精确度上做到了很好的平衡。但YOLACT需要设置较大的特征提取网络(Backbone)才能保证较高的分割精确度，这会导致模型的权重文件较大，占用较多的运行显存和硬件存储空间，同时较高的模型参数量也限制了速度的提升。

为了解决上述问题，本文在YOLACT基础上，提出了一种运行效率更高、资源占用更少的利用通道剪枝技术的实例分割方法。具体地，是利用通道剪枝技术对YOLACT网络进行压缩和加速，主要针对YOLACT中耗时较高的特征提取网络(Backbone)部分进行优化。首先，对模型的Backbone部分进行通道剪枝[16]的参数压缩，然后将剪枝后网络的卷积层和批量归一化层融合为一个等效计算层，进一步提升运行速度，最后在COCO val2017上对该方法的性能进行了评估。实验结果表明，在设置Backbone为ResNet101时，本文方法的模型文件大小为84 MB，比YOLACT的模型文件小56.9%，实例分割速度可以达到36.1 fps，相比YOLACT提升了28.6%，而分割精确度mAP只降低了7.3%。

1 相关工作

1.1 实例分割

早期，基于CNN的实例分割算法大多分两阶段(two−stage)结构，先使用区域候选网络(Region Proposal Network，RPN)[17]产生候选区域，然后在每个感兴趣区域(Region of Interest，ROI)[17]上利用全卷积网络(Fully Convolutional Networks，FCN)[18]生成分割掩码。Deepmask[19]通过输入图像中出现的实例来预测候选掩码，但对边界分割的准确度较低。Sharpmask[20]通过生成不同深度的图像特征得到更高质量掩码来改进Deepmask。InstanceFCN[21]提出了位置敏感评分图，以保证分割任务的平移可变性；FCIS[22]通过对位置敏感的得分图进行改进，提高了实例分割的精确度；Mask RCNN[1]改进了损失函数，并加入了RoIAlign层，在具有挑战性的COCO数据集上获得了很高的精确度。上述方法对运行效率和资源占用的问题探讨得较少。

最近，单阶段(one−stage)实例分割工作得到越来越多的重视和研究。Chen等提出的BlendMask[23]通过更合理的特征融合模块结合高层次和低层次的语义信息来提取更准确的实例分割特征，但需要为每个ROI提取特征，导致计算量较大。Wang等提出的按位置分割对象(Segmenting Objects by Locations，SOLO)[24]把实例分割问题转化为分类问题，但是由于预选框的网格数量较大，导致整体计算耗时较多，也无法做到实时处理。尽管上述的one−stage实例分割方法概念上比two−stage的方法要快，但由于引入了计算量比较大的操作，也无法做到实时。

还有一些方法，在目标检测后串接语义分割网络[25]，使用像素聚类[26−27]形成最后的实例掩码。上述方法需要多阶段计算或者计算量较大的聚类操作，限制了运行速度的提升。

1.2 实时实例分割

文献[28]指出可以通过学习形状编码实现30 fps的实时处理，但精确度远不如当前主流的实例分割算法。Box2Pix[29]依靠轻量级的主干网络结合手工设计的算法，在Cityscapes数据集上达到10.9 fps，在KITTI数据集上达到35 fps，但在更有挑战性的COCO数据集上没有相关的实验结果。Lee等基于目标检测Anchor−Free的思想提出CenterMask[30]实例分割框架，并设计了一个轻量级的网络CenterMask−Lite，CenterMask−Lite使用轻量级主干网络时，能够达到实时处理的结果，文献[30]中的结果表明在TiTan XP GPU设备上CenterMask−Lite与YOLACT网络耗时相同(都为23 ms)，但是CenterMask的模型文件较大(153 MB)，同样会占用较多的存储空间和运行时的显存，不利于实际部署，还有进一步压缩模型的空间。

Daniel Bolya等提出的one−stage实例分割网络YOLACT，将实例分割任务拆分成2个并行的子任务：一个分

支去生成一系列的原型掩码；另一个分支预测每个实例的掩码系数，通过线性组合2个分支得到实例分割的结果。YOLACT可以做到实时的实例分割，但需要设置较大的特征网络(Backbone)才能获得较高的分割精确度，这会导致模型权重文件过大，耗时较高，也会占用较多的运行显存、存储空间等硬件资源，不利于在实际场景的应用，而这正是本文所解决的问题。

2　提出的方法

本文的模型以YOLACT为基础架构，先使用通道剪枝方法对它的特征提取网络(Backbone)进行压缩，然后将剪枝后网络中的卷积层和批量归一化层(Batch Normalization，BN)融合为一个等效计算层，实现对模型的压缩和加速，降低硬件资源的占用，提升模型的处理速度。

2.1 基于通道剪枝技术加速实例分割网络

LIU Zhuang等提出了一种基于BN层放缩因子的通道剪枝的方法[16]，对常见的分类网络(ResNet[31]、VGG(Visual Geometry Group)[32]等)进行了剪枝处理，在ImageNet、CIFAR等分类数据集上取得了非常好的效果。但通道剪枝的方法在实例分割任务上的应用还比较少，本文尝试将该方法应用在实例分割任务上，以期获得压缩模型大小，提升运行速度的效果。

针对YOLACT网络需要设置较大的Backbone保证分割精确度的同时，会带来较高的资源占用、运行速度损失的问题，提出了改进的模型，主要是对YOLACT的Backbone部分进行通道剪枝处理。提出的方法具体如下。

首先，设置模型的Backbone(如ResNet101)初始化网络，并设置初始的损失函数与YOLACT的损失函数一致，损失函数如式(1)：

$$L\mathrm{oss}=L_{\mathrm{cls}}+L_{\mathrm{boxes}}+L_{\mathrm{mask}} \tag{1}$$

式中L_{cls}、L_{boxes}、L_{mask}分别为模型的分类、目标检测、分割掩码3个分支的损失函数，与文献[13]中的设置一致。初始化网络和损失函数后，在训练集上开始第一阶段的训练，当损失函数不再下降，保存模型文件。下一步，在损失函数中加入L_1正则化项：

$$L_1=\lambda\sum_{\gamma\in\Gamma}|\gamma| \tag{2}$$

式中：λ为超参数；γ为模型中Backbone部分的BN层放缩因子，是可以学习的参数；Γ为Backbone中所有通道BN层放缩因子的集合。加入L_1正则化项后，模型的损失函数如下：

$$L\mathrm{oss}=L_{\mathrm{cls}}+L_{\mathrm{boxes}}+L_{\mathrm{mask}}+L_1 \tag{3}$$

在损失函数中加入L_1正则化项后，读取第一阶段训练后保存的模型文件，在训练集上再对网络进行第二阶段的稀疏化训练，当损失不再下降时停止训练，保存稀疏化后的模型文件。稀疏化训练的过程中，模型中一些通道的BN层的放缩因子γ会趋于0，这些通道对后续计算结果的贡献度比较小，剪去这些通道对实例分割精确度的影响也比较小。因此，在模型稀疏化训练完成后，按照自定义设置的剪枝比例消去一定百分比的γ较小的通道，不仅可以减少模型中的参数量和计算量，加速计算，同时对精确度的影响比较小。通道剪枝的原理如图1所示。

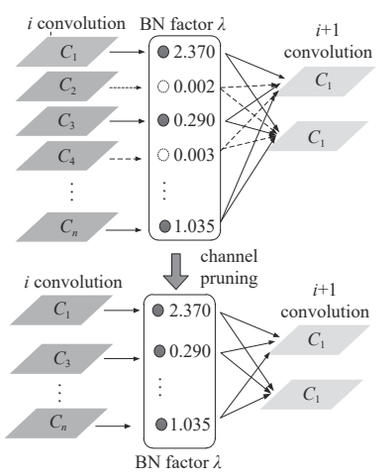

Fig.1 Channel pruning

图1 通道剪枝

但是，消去过多的通道也会导致误差的不断累积，从而对实例分割结果产生比较大的影响。因此也需要根据实际测试结果设置合理的通道剪枝比例，获得运行速度和实例分割精确度的平衡。关于剪枝比例对分割精确度、推理速度的影响，本文通过测试不同剪枝比例下的实测结果进行评估。该部分的理论依据，比如针对不同实例分割网络剪枝比例对分割精确度、推理速度的理论影响曲线，还在进一步探索和验证中。并且由于神经网络的可解释性理论还不完备，当前的方法还是以实际测试结果作为评估的主要方式。

最后，为了减少剪枝操作带来的精确度损失，需要再对剪枝后的网络进行权重参数微调。具体的做法是：去掉网络损失函数中的正则化项，读取保存的剪枝后模型文件，将剪枝后的网络在训练集上继续训练，微调模型中的权重参数。对剪枝后的模型微调后，保存最终的网络模型文件，完成对实例分割网络的剪枝和压缩处理。

2.2 融合网络中的卷积层和 BN

对于具有 BN 层的卷积神经网络，在网络推理阶段，将卷积层和 BN 层融合为一个等效的计算层[33]，可以有效减少网络中的计算量，提高运行速度，如图 2 所示。本文将剪枝后的实例分割模型中剩余通道的卷积层和 BN 层融合为一个等效的计算层，进一步提高模型的运行速度。该过程是基于等式的等效转换，理论上对实例分割的精确度没有影响。但是，由于实际实施过程中，浮点数计算的不精确性，以及对过多的网络层融合会引入累积误差，因此最终的结果也会对实例分割精确度 mAP 产生一些影响。

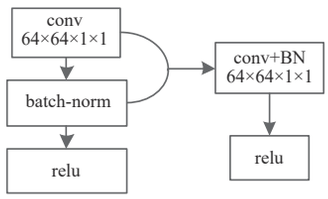

Fig.2 Combined BN and convolution
图 2 融合卷积层和 BN 层

3 实验

在 COCO train2017 上对本文的实例分割模型进行了训练，并在 COCO val2017 上进行评估，使用平均精确度 mAP 度量实例分割的精确度。测试的硬件环境为：1080Ti 单 GPU(8 GB 显存)、Intel Xeon E5−2683 CPU、128 GB 内存。

3.1 模型的实例分割结果测试和对比

以 ResNet101 作为模型的特征提取网络(Backbone)为例，实验细节如下：设置模型的 Backbone 为 ResNet101，Batchsize 设为 8，初始学习率为 0.001，分别在[20,40,50] epochs 后降低 10 倍的学习率，训练 64 epochs 后停止训练；稀疏化训练时，L_1 正则化项中的超参数 λ 取 10^{-5}，Batchsize 设为 8，学习率固定为 10^{-5}，训练 12 epochs 后停止稀疏化训练；下面的测试表格数据是在设置通道剪枝比例为 50% 时，剪枝并微调完成后，在 COCO val2017 上的评估结果。与其他实例分割模型的运行速度 FPS(Frames Per Second)、分割精确度 mAP、模型文件大小、显存占用等结果对比见表 1(SOLO 和 BlendMask 暂未开源，无法测试显存占用和模型文件大小，故未标注)。

表 1 不同实例分割架构的对比

Table1 A comparison of different methods

method	*t*/ms	FPS	mAP	size/MB	GPU usage/MB
FCIS[22]	151.5	6.6	29.5	223	3 376
Mask R–CNN[1]	116.3	8.6	35.7	246	3 574
MS R–CNN[34]	116.3	8.6	38.3	305	4 352
SOLO[24]	96.2	10.4	37.8	—	—
BlendMask[23]	101.8	9.8	38.4	—	—
YOLACT[14]	35.48	28.2	30.62	195	1 377
Ours*	29.14	34.3	28.64	84	1 189
Ours*(fused BN)	27.48	36.1	28.42	84	1 189

表 1 中对比的几种实例分割网络，特征提取网络都为 ResNet101，Ours*为只剪枝压缩处理，Ours*(fused BN)为剪枝后再融合 bn 层，剪枝比例 50%。除了 YOLACT 和本文的方法之外，其他模型的速度基本都在 10 fps 以下，达不到实时处理。在同样的实验环境下，本文的模型耗时相比 YOLACT 减少了 8 ms，耗时缩短了 22.5%，本模型的处理速度可以达到 36 fps，相比 YOLACT 的 28 fps 提升了 28.6%。模型文件的大小相比 YOLACT 大小减少了 111 MB，压缩了 56.9%。另外，显存的占用也降低了 13.6%，同时 mAP 为 28.32，比 YOLACT 低 7.3%，与 FCIS 的精确度差距在 3.6%，分割精确度达到了目前主流实例分割网络的水平[7]。由于剪枝操作不仅会影响当前通道的数量，也会消去后续与该通道连接的卷积，因此剪枝比例设为 50% 时，实际的模型压缩效果会大于 50%。通道剪枝消去的是对后续实例分割计算影响较小的通道，不仅可以减少网络的计算量，对分割精确度影响也比较小，实验结果也验证了方法的有效性。

另外，对比表 1 中只进行剪枝处理以及剪枝后再融合 BN 层的结果，发现在融合 BN 层后，运行速度提升了 2 ms 左右，对模型文件大小几乎没有影响。因为 BN 层的参数量相对卷积层来说比较小，对模型文件大小的影响可以忽略，但是融合 BN 层减少了网络的计算流程和步骤，能够提升运行速度。卷积层和 BN 层的融合过程在概念上是等效计算，但是较多的网络层数的融合也会引入累积误差，从而对分割精确度产生一些影响，因此融合 BN 层后的模型的 mAP 降低了 0.2。这可能也是由于在实现卷积层和 BN 层的融合时未处理好累积误差，需要在后续工作中继续完善。

3.2 实际分割效果图对比

本文模型在设置 50% 剪枝比例时的实例分割效果与原始 YOLACT 分割的对比如图 3 所示，二者初始 Backone 都设为 ResNet101。直观可见，本文模型分割的整体效果与 YOLACT 相差不大，仅在单个物体的分割细节和边缘

上有一定的损失，这是由于剪去的通道对实例分割结果产生的误差所引起的。根据表 1，本文的方法相比 YOLACT 的分割精确度只下降了 7%，通过图 3 实际分割效果图也可以直观地看出，本文的方法在简单场景中能获得比较好的分割效果。但本模型相比 YOLACT 能够降低 56.9% 的模型文件大小，提升 28.6% 的运行速度，降低 13.6% 的显存占用。在对精确度要求不高的场景下，本模型能够获得较好的分割效果，同时具有运行速度更快、节省资源等优点，更有利于实际场景的部署和应用。

Fig.3 Comparison of mask details, YOLACT(upper) and our methods(lower)
图 3　YOLACT(上)与本文方法(下)分割实际效果图

3.3　不同 Backbone 对结果的影响

作为对比，针对本文的实例分割模型，还测试了设置 Backbone 为 ResNet50、ResNet18 时对运行效率、分割精确度的影响，方法同前面设置 ResNet101 时的实验步骤一致，为了对比，统一设置通道剪枝比例为 50%，剪枝完成后融合卷积层和 BN 层。测试结果与相同 Backbone 的 YOLACT 网络进行了对比，见表 2。对于相同的 Backbone，本文的方法相比于 YOLACT 模型，都能够有效地压缩模型的大小，提升运行速度。在设置 ResNet18 作为 Backbone 时，YOLACT 的分割精确度为 20.43，本文压缩加速后的模型的分割精确度为 17.12，分割精确度都比较低，与 ResNet101 作为 Backbone 时的分割精确度相比都降低了 35% 左右。过低的分割精确度在实际场景中的实用价值也比较低。因此，在实际应用中需要考虑实际需求，设置合理的 Backbone，才能获得分割精确度和运行速度上的平衡。

表 2　设置不同 Backbone 时的结果对比

Table2 A comparison of different Backbone values

method	Backbone	*t*/ms	FPS	mAP	size	GPU usage
YOLACT[14]	ResNet101	35.48	28.2	30.62	195	1 377
YOLACT[14]	ResNet50	24.84	40.3	29.66	120	1 283
YOLACT[14]	ResNet18	20.63	48.5	20.43	83	1 246
Ours*(fused BN)	ResNet101	27.48	36.1	28.32	84	1 189
Ours*(fused BN)	ResNet50	19.39	51.6	26.76	49	1 104
Ours*(fused BN)	ResNet18	16.72	59.8	17.12	38	1 089

3.4　剪枝比例对实例分割结果的影响

另外，分析了不同的剪枝比例对分割精确度 mAP 和运行时间的影响，不同剪枝比例下的分割精确度和运行时间的关系如图 4 所示(Backbone 设置 ResNet50)。横坐标是剪枝比例的百分比，随着剪枝比例的增大，网络的耗时(单位 ms)在不断下降，同时实例分割的精确度(mAP)也在下降。当剪枝比例设置较小时，剪去的主要是通道放缩因子 γ 最小的部分通道，因此对实例分割结果影响也较小，mAP 的下降不明显。随着剪枝比例的不断增大，分割精确度的下降会越来越快，这是

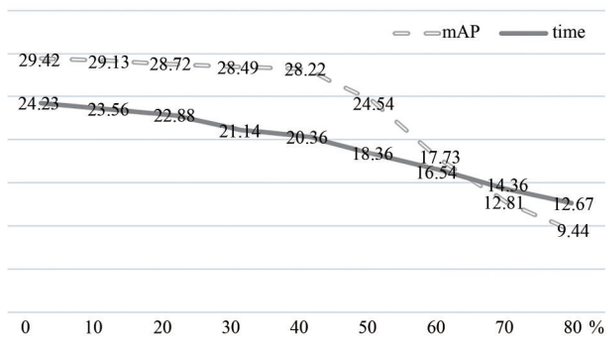

Fig.4 Different pruning percentages on mAP and running time
图 4　不同剪枝百分比对 mAP 和运行时间的影响

由于通道放缩因子γ较大的部分通道也被剪去了，从而对实例分割结果产生了较大的影响。因此随着剪枝比例不断增大，分割精确度下降的趋势也越来越大。实际应用中，设置合适的通道剪枝比例，才能获得分割精确度和运行效率的平衡。

关于设置合适的通道剪枝比例的相关理论依据，由于当前的一些理论工具还不完善，如神经网络的可解释性理论等还不完备，针对实例分割剪枝比例这个问题，在进一步探索和实验中，也在尝试对其他实例分割网络进行通道剪枝实验，建立统一的理论。当前的方法还是以实际测试不同剪枝比例下的结果作为指导实际部署时的参考。剪枝比例对精确度和速度的影响本身是一个二者权衡的(trade-off)的问题，剪枝越多，运行速度越快，但是分割精确度就会越低。本文表1和表2的数据选择的是在剪枝比例大约为50%时的测试结果。由于剪枝操作不仅会影响当前通道的数量，也会消去后续与该通道连接的卷积层，因此剪枝比例设为50%时，实际的模型压缩效果可能会大于50%，因此设定的剪枝比例也是一个近似的比例。

4 结论

随着实例分割技术的应用越来越广泛，提高运行速度，降低资源占用会更加有利于实例分割模型的部署和应用。针对这个问题，提出了一种简单且有效的实例分割方法，该方法在实时实例分割网络YOLACT的基础上，使用通道剪枝技术优化改进了特征提取网络部分(Backbone)，然后融合模型中的卷积层和BN层，进一步提高模型的运行速度。

本文模型在COCO数据集上进行了训练和测试。在设置特征提取主干网络(Backbone)为ResNet101时，模型文件大小为84 MB，比原始的YOLACT模型文件大小减少111 MB，模型的大小压缩了56.8%，可以有效减少硬件存储空间的占用，同时分割处理速度达到了36.1 fps，运行速度提升了28.6%，并且运行时显存占用也降低了13.6%，分割精确度只下降了7.3%，更加有利于实例分割技术的部署和应用。

参考文献：

[1] HE K,GKIOXARI G,DOLLÁR P, et al. Mask r-cnn[C]// IEEE International Conference on Computer Vision. Venice,Italy:IEEE, 2017:2961-2969.

[2] MOHAN A,PAPAGEORGIOU C,POGGIO T. Example-based object detection in images by components[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2001,23(4):349-361.

[3] DALAL N,Triggs B. Histograms of oriented gradients for human detection[C]// IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR'05). San Diego,CA,USA:IEEE, 2005:886-893.

[4] LADICKÝ L, RUSSELL C, KOHLI P, et al. Associative hierarchical crfs for object class image segmentation[C]// IEEE 12th International Conference on Computer Vision. Kyoto,Japan:IEEE, 2009:739-746.

[5] LIU Z,LI X,LUO P,et al. Semantic image segmentation via deep parsing network[C]// IEEE International Conference on Computer Vision. Santiago, Chile:IEEE, 2015:1377-1385.

[6] 牟效乾,陈小龙,苏宁远,等. 基于时频图深度学习的雷达动目标检测与分类[J]. 太赫兹科学与电子信息学报, 2019,17(1):105-111. (MOU Xiaoqian,CHEN Xiaolong,SU Ningyuan,et al. Radar detection and classification of moving target using deep convolutional neural networks on time-frequency graphs[J]. Journal of Terahertz Science and Electronic Information Technology, 2019,17(1):105-111.)

[7] MINAEE S,BOYKOV Y Y,PORIKLI F,et al. Image segmentation using deep learning:a survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021(1):1-22.

[8] REDMON J, FARHADI A. YOLO9000:better,faster,stronger[C]// IEEE Conference on Computer Vision and Pattern Recognition. Honolulu,HI,USA:IEEE, 2017:7263-7271.

[9] REDMON J,FARHADI A. Yolov3:an incremental improvement[J]. Computer Vision and Pattern Recognition, 2018. doi:10.48550/arXiv.1804.02767.

[10] LIU W,ANGUELOV D,ERHAN D,et al. Ssd:single shot multibox detector[C]// European Conference on Computer Vision. [s.l.]:Springer,Cham, 2016:21-37.

[11] ROMERA E,ALVAREZ J M,BERGASA L M,et al. ERFNet:efficient residual factorized convnet for real-time semantic segmentation [J]. IEEE Transactions on Intelligent Transportation Systems, 2018,19(1):263-272.

[12] PASZKE A,CHAURASIA A,KIM S,et al. Enet:a deep neural network architecture for real-time semantic segmentation[J]. Computer Vision and Pattern Recognition, 2016. doi:10.48550/arXiv.1606.02147.

[13] WANG Y,ZHOU Q,LIU J,et al. Lednet: a lightweight encoder-decoder network for real-time semantic segmentation[C]// IEEE International Conference on Image Processing(ICIP). Taipei,Taiwan,China:IEEE, 2019:1860-1864.

[14] BOLYA D,ZHOU C,XIAO F,et al. Yolact:real-time instance segmentation[C]// IEEE/CVF International Conference on Computer Vision. Seoul,Korea(South):IEEE, 2019:9157-9166.

[15] LIN T Y,MAIRE M,BELONGIE S,et al. Microsoft coco:common objects in context[C]// European Conference on Computer Vision. [s.l.]:Springer,Cham, 2014:740-755.
[16] LIU Z,LI J,SHEN Z,et al. Learning efficient convolutional networks through network slimming[C]// IEEE International Conference on Computer Vision. Venice,Italy:IEEE, 2017:2736-2744.
[17] REN S,HE K,GIRSHICK R,et al. Faster r-cnn:towards real-time object detection with region proposal networks[J]. Advances in Neural Information Processing Systems, 2015(28):91-99.
[18] LONG J,SHELHAMER E,DARRELL T. Fully convolutional networks for semantic segmentation[C]// IEEE Conference on Computer Vision and Pattern Recognition. [s.l.]:IEEE, 2015:3431-3440.
[19] PINHEIRO P O,COLLOBERT R,DOLLÁR P. Learning to segment object candidates[J]. Advances in Neural Information Processing Systems, 2015(2):1990-1998.
[20] PINHEIRO P O,LIN T Y,COLLOBERT R,et al. Learning to refine object segments[C]// European Conference on Computer Vision. [s.l.]:Springer,Cham, 2016:75-91.
[21] DAI J,HE K,LI Y,et al. Instance-sensitive fully convolutional networks[C]// European Conference on Computer Vision. [s.l.]:Springer, Cham, 2016:534-549.
[22] LI Y,QI H,DAI J,et al. Fully convolutional instance-aware semantic segmentation[C]// IEEE Conference on Computer Vision and Pattern Recognition. Honolulu,HI,USA:IEEE, 2017:2359-2367.
[23] CHEN H,SUN K,TIAN Z,et al. Blendmask:top-down meets bottom-up for instance segmentation[C]// IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, USA:IEEE, 2020:8573-8581.
[24] WANG X,KONG T,SHEN C,et al. Solo:segmenting objects by locations[C]// European Conference on Computer Vision. [s.l.]:IEEE, Springer,Cham, 2020:649-665.
[25] KIRILLOV A,LEVINKOV E,ANDRES B,et al. Instancecut:from edges to instances with multicut[C]// IEEE Conference on Computer Vision and Pattern Recognition. Honolulu,HI,USA:IEEE, 2017:5008-5017.
[26] BAI M,URTASUN R. Deep watershed transform for instance segmentation[C]// IEEE Conference on Computer Vision and Pattern Recognition. Honolulu,HI,USA:IEEE, 2017:5221-5229.
[27] LIANG X, LIN L, WEI Y, et al. Proposal-free network for instance-level object segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017,40(12):2978-2991.
[28] JETLEY S,SAPIENZA M,GOLODETZ S,et al. Straight to shapes:real-time detection of encoded shapes[C]// IEEE Conference on Computer Vision and Pattern Recognition. Honolulu,HI,USA:IEEE, 2017:6550-6559.
[29] UHRIG J,REHDER E,FRÖHLICH B,et al. Box2pix:single-shot instance segmentation by assigning pixels to object boxes[C]// IEEE Intelligent Vehicles Symposium(IV). Changshu,Jiangsu, China:IEEE, 2018:292-299.
[30] LEE Y, PARK J. Centermask: real-time anchor-free instance segmentation[C]// IEEE/CVF Conference on Computer Vision and Pattern Recognition. [s.l.]:IEEE, 2020:13906-13915.
[31] HE K,ZHANG X,REN S,et al. Deep residual learning for image recognition[C]// IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas,NV,USA:IEEE, 2016:770-778.
[32] SIMONYAN K,ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. Computer Vision and Pattern Recognition, 2014. doi:10.48550/arXiv.1409.1556.
[33] CHEN T, MOREAU T, JIANG Z, et al. TVM: end-to-end optimization stack for deep learning[J]. Computer Vision and Pattern Recognition, 2018. doi:10.48550/arXiv.1802.04799.
[34] HUANG Z,HUANG L,GONG Y,et al. Mask scoring r-cnn[C]// IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach,CA,USA:IEEE, 2019:6409-6418.

作者简介：

宁　欣(1989-)，男，博士，副研究员，主要研究方向为计算机视觉、深度学习、形象认知计算技术. email:ningxin@semi.ac.cn.

刘江宽(1990-)，男，硕士，工程师，主要研究方向为计算机视觉、模型加速技术.

李卫军(1975-)，男，博士，研究员，主要研究方向为计算机视觉、模式识别.

石　园(1989-)，男，硕士，工程师，主要研究方向为计算机视觉、模型加速技术.

支金林(1997-)，男，学士，工程师，主要研究方向为算法加速与部署技术.

南方哲(1994-)，男，硕士，工程师，主要研究方向为计算机视觉、图像生成.