基于深度学习的目标检测算法综述（三）

vision
算法攻城狮

关注他

97 人赞同了该文章

基于深度学习的目标检测算法综述（一）

基于深度学习的目标检测算法综述（二）

基于深度学习的目标检测算法综述（三）

本文内容原创，作者：美图云视觉技术部 检测团队，转载请注明出处

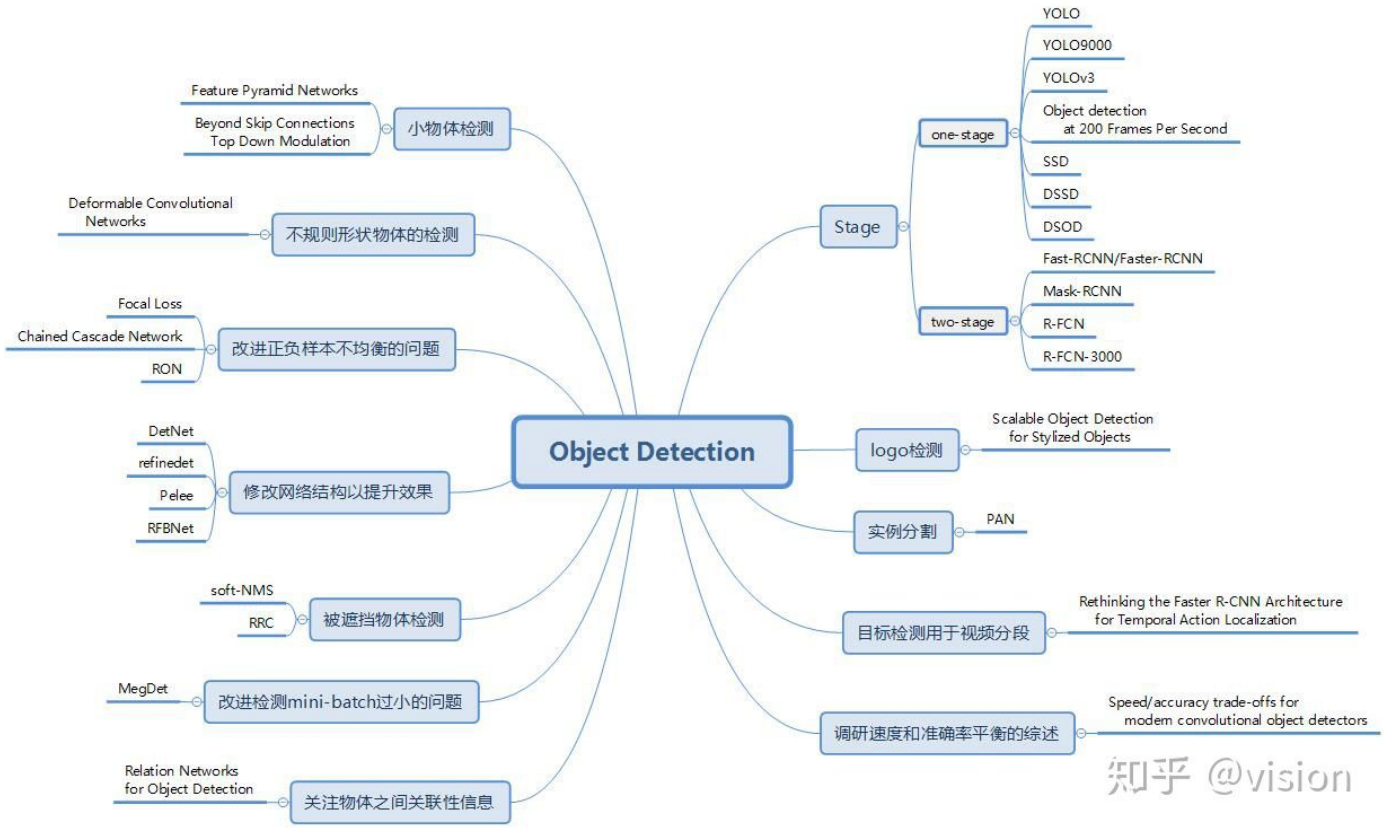
基于深度学习的目标检测算法综述分为三部分：

1. Two/One stage算法改进。这部分将主要总结在two/one stage经典网络上改进的系列论文，包括Faster R-CNN、YOLO、SSD等经典论文的升级版本。

2. 解决方案。这部分我们归纳总结了目标检测的常见问题和近期论文提出的解决方案。

3. 扩展应用、综述。这部分我们会介绍检测算法的扩展和其他综述类论文。

在上篇文章（基于深度学习的目标检测算法综述（二））我们归纳总结了目标检测的常见问题和近期论文提出的解决方案。本篇文章是基于深度学习的目标检测算法综述的最后部分。在这里，我们会介绍目标检测算法的扩展应用，每个方向给出1篇代表论文，同时，最后给出google发表在CVPR2017上的目标检测算法效果和性能评估论文。



3. 扩展应用、综述

目标检测在很多计算机视觉领域中已经有了很多成熟的应用，如人脸检测、行人检测、图像检索和视频监控等。而目标检测算法不仅可以应用在普通物体的分类和定位上，在近年也有了很多扩展。我们会在后文中介绍其中三篇论文，如logo检测论文“**Scalable Object Detection for Stylized Objects**”、通过改进的Top-Down结构提升识别和分割效果的论文“**Path Aggregation Network for Instance Segmentation**”以及把目标检测思想用于视频分段的“**Rethinking the Faster R-CNN Architecture for Temporal Action Localization**”。

论文"**Mimicking Very Efficient Network for Object Detection**"借鉴知识蒸馏思想，在训练过程中用训练好的大网络作为监督网络指导小网络的参数学习，实现从零开始训练目标检测网络。

本综述中的最后部分我们还介绍了Google发表于CVPR2017的平衡速度和准确率的论文"**Speed/accuracy trade-offs for modern convolutional object detectors**"，该文用非常大量详实的实验探讨了常用目标检测算法及其各参数设置对于目标检测speed和accuracy的影响，对如何平衡速度和准确率提出了建议。

3.1 logo检测：Scalable Object Detection for Stylized Objects

论文链接：arxiv.org/pdf/1711.0982

开源代码：无

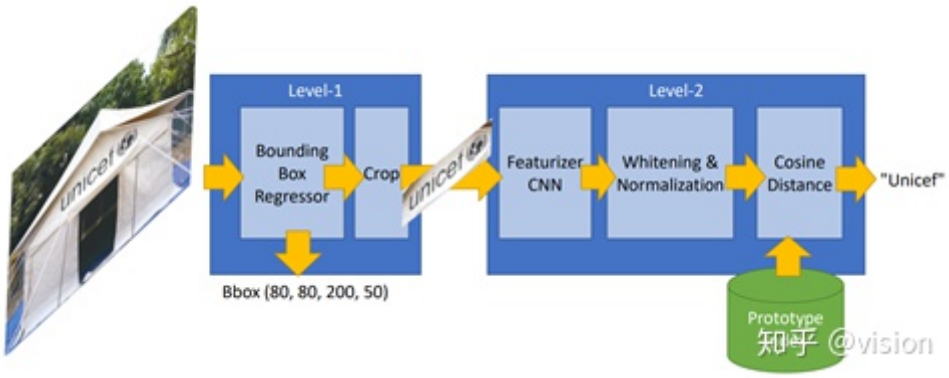
录用信息：无

论文目标：

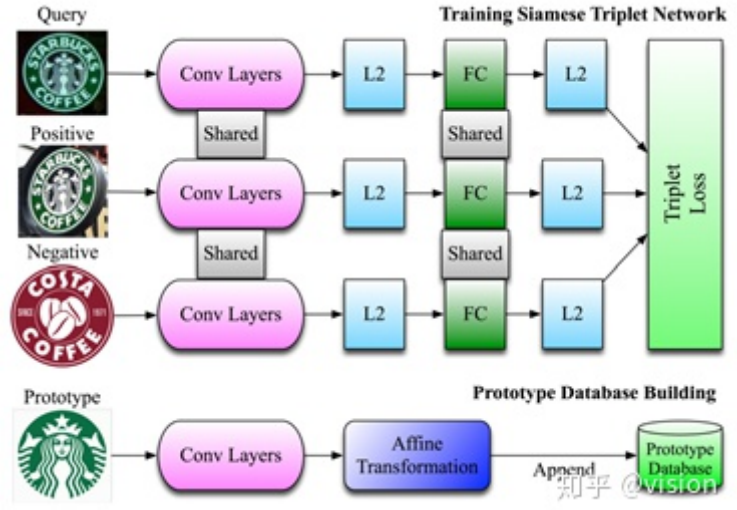
论文旨在解决对logo的检测问题。作者将e2e的检测模型分为两步，1. 使用Single Shot Multibox Detector（文中使用YOLO v2）检测可能的logo位置。2. 使用Triplet Loss训练的deep image-similarity network模型对logo进行检索。

核心思想：

Logo检测和普通目标检测的区别，以及分成两步检测的原因有：1. Logo种类可能很多，如果用Softmax分类几千种物体效果会很差。随着数量增多，识别效果会很快变差，由于每个分类器不仅要识别是哪个物体而且要识别Proposal中是否包含待检测目标。2. Logo更新很快，如果用普通目标检测，添加新数据后需要重新训练。3. Logo在图片中一般都有固定的字体、特性，而且相对普通物体更醒目易于观察。4. Logo相比普通物体和背景的关系不大。作者用人脸检测来做示例，人脸和logo检测类似，都是大量实例检测。每张人脸都是不同的，但是如果用目标检测来区分每个人脸是不现实的。只能先做人脸检测，然后再做人脸检索。



如上图所示，本文使用的网络为2-layer网络。Layer1是YOLOv2，通过识别Logo来提取框，然后crop+resize为224*224。 作者提出使用Single Shot Multibox Detector而非proposal-based网络的原因为logo检测需要使用物体的全局信息。Layer2是训练检索网络。训练为典型的triplet loss训练过程：使用两组相同logo，一组不同logo，通过缩小类内loss、增大类间loss来促进收敛网络。Layer2网络通过CNN提取特征然后经过L2归一化和FC层，最终提取feature vector。检索数据集的准备过程在训练完成后进行，使用特定logo经过特征提取和仿射变换操作，制作数据集。最终使用feature vector的余弦距离来进行实例检索。



算法效果：

由于这篇论文针对的是logo检测的改进，所以测评都是在logo数据集上。Flickrlogos-32为开源的logo数据集，MSR1k和Synth9K为作者自己造的logo数据集。

	Flickrlogos-32	MSR1k-Test	Synth9k
FasterRCNN-Resnet-101	0.80	0.58	*N.A.*
FasterRCNN-Inception	0.82	0.61	*N.A.*
Two-Layer (Ours^2)	0.75	0.52	0.58

3.2 实例分割：Path Aggregation Network for Instance Segmentation

论文链接：arxiv.org/abs/1803.0153

开源代码：无

录用信息：CVPR2018

论文目标：

我们在综述的第二部分提到，FPN是目标检测Top-Down结构的一种常见形式。这篇论文通过改进主干网络FPN的结构，缩短了从低层特征到高层特征之间的路径，进一步减少了定位信息流动的难度，从而同时提升了目标检测和实例分割的效果。

核心思想：

网络结构如下图。

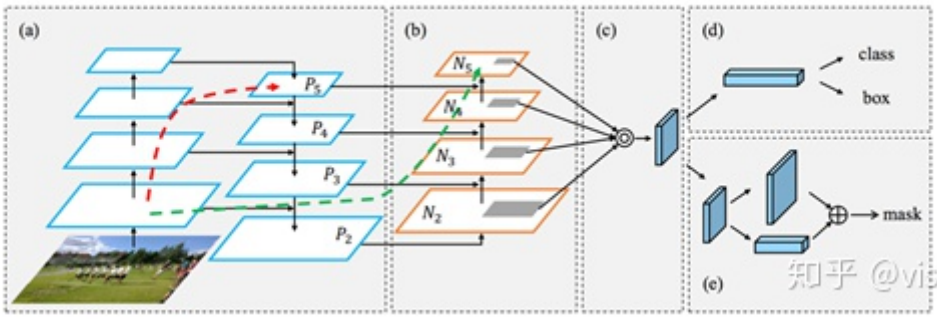


图1 PANet的框架

PANet的框架如图1所示，Bottom-up Path用于增强Low-layer的信息传播。Adaptive Feature Pooling允许每个候选区域可以获取所有特征层次的信息，以此用于预测。添加了一个互补分支用于预测mask。

PANet的创新点可以总结如下：

a.创建Bottom-up增强路径缩短信息路径，利用Low-level 特征中存储的精确定位信号，提升特征金字塔架构。

b.创建Adaptive Feature Pooling合并所有特征层次上每个proposal的特征信息，避免任意分配结果。

c.使用小型Fc层补充mask预测，捕获每个proposal的不同视图，以此与原始的Mask R-CNN互补。通过融合这两个视图，增加信息多样性，能更好的预测mask。

具体如下：

1. Bottom-up Path Augmentation

高层的神经元主要响应整个物体，其他神经元更倾向于响应局部纹理信息。该网络结构通过传播low-level的强响应，增强了整个特征结构的定位能力。因为对边缘和实例部分的强响应对于精确定位实例是强指向标。因此论文使用横向连接将低层和高层的信息连接起来。使用bottom-up path将底层信息传递到决策层只需要不到十层（如图1绿色虚线所示），但是FPN则需要通过backbone，走完整个ResNet基础网络，需要走一百多层。具体的Augmented Bottom-up Structure 如图2所示。

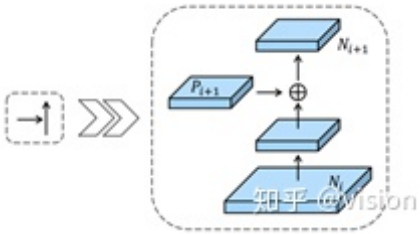


图2 Augmented Bottom-up Structure

网络结构在FPN{P2,P3,P4,P5}后接{N2,N3,N4,N5}（图1 a,b所示）,其中N2就是P2没有经过任何操作。如图2所示，每个模块通过横向连接将较高分辨率的Ni和低分辨率的Pi+1横向连接，产生Ni+1。

2. Adaptive Feature Pooling

FPN的proposals根据其大小分配给不同的特征层次。这样尺寸小的proposal被分配给Low-level，而尺寸大的分配给High-level，但这可能会产生非最优结果。例如两个只相差10个像素的proposals会被分给不同的特征层次，但实际上它们很相似。因此，特征的重要性与其所属的特征层次并不是强相关。高层次的特征由更大的感受野产生，获取了更丰富的语义信息。小型proposals获取这些特征可以更好地使用语义信息做预测。另一方面，低层次特征有许多细节和定位信息。基于此，论文提出对每个proposal池化所有层次的特征，然后融合它们做预测。其实现过程如图3所示。

将每个候选区域映射到不同的特征层次，用RoIAlign池化不同层次的特征网格，进一步融合不同层次的特征网格（逐像素求最大或者求和）。

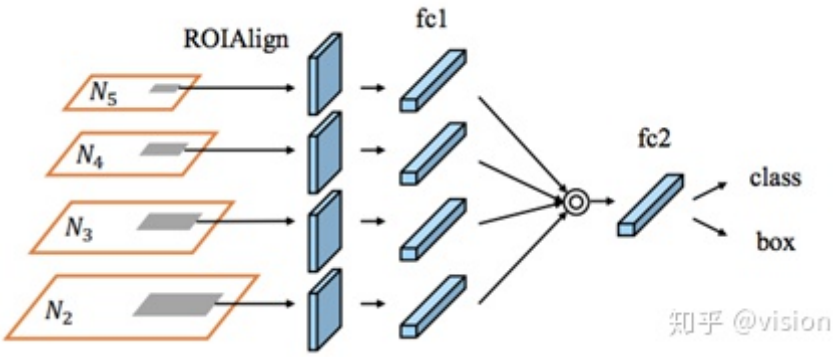


图3 Adaptive Feature Pooling Structure

3. Fully-connected Fusion

Fc与FCN有不同的属性，由于局部感受野和共享参数，FCN有像素级预测，而Fc对不同空间的预测均是通过一组可变参数实现，因此它的位置敏感度较高。另外Fc预测不同空间位置均通过全部proposals的全局信息实现，这对于区分不同实例和识别属于同一对象的分离部分很有效。因此考虑将这两种层的预测结果融合可以达到更好的预测。其实现如图4所示。

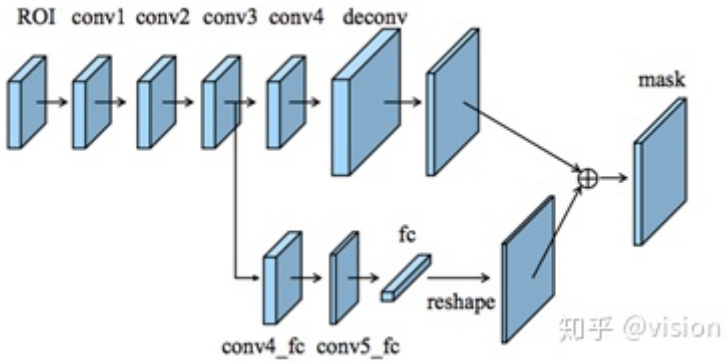


图4 Fully-connected Fusion Structure

算法效果：

在实例分割方面，使用相同的初始化模型，PANet比Mask R-CNN好了将近3个点；在目标检测方面，使用renet50当基础网络的PANet比coco2016年的冠军高了0.9%，它的backbone是2×ResNet-101+ 3×Inception-ResNet-v2和推理tricks。

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Backbone
Champion 2016 [33]	37.6	59.9	40.4	17.1	41.0	56.0	6×ResNet-101
Mask R-CNN [21]+FPN [35]	35.7	58.0	37.8	15.5	38.1	52.4	ResNet-101
Mask R-CNN [21]+FPN [35]	37.1	60.0	39.4	16.9	39.9	53.5	ResNeXt-101
PANet / PANet [ms-train]	36.6 / 38.2	58.0 / 60.2	39.3 / 41.4	16.3 / 19.1	38.1 / 41.1	53.1 / 52.5	ResNet-50
PANet / PANet [ms-train]	40.0 / **42.0**	62.8 / **65.1**	43.1 / **45.7**	18.8 / **22.4**	42.3 / **44.7**	57.2 / **58.1**	ResNeXt-101

Method	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{bb}_S	AP^{bb}_M	AP^{bb}_L	Backbone
Champion 2016 [27]	41.6	62.3	45.6	24.0	43.9	55.2	2×ResNet-101 + 3×Inception-ResNet-v2
RentinaNet [36]	39.1	59.1	42.3	21.8	42.7	50.2	ResNet-101
Mask R-CNN [21]+FPN [35]	38.2	60.3	41.7	20.1	41.1	50.2	ResNet-101
Mask R-CNN [21]+FPN [35]	39.8	62.3	43.4	22.1	43.2	51.2	ResNeXt-101
PANet / PANet [ms-train]	41.2 / 42.5	60.4 / 62.3	44.4 / 46.4	22.7 / 26.3	44.0 / 47.0	54.6 / 52.3	ResNet-50
PANet / PANet [ms-train]	45.0 / **47.4**	65.0 / **67.2**	48.6 / **51.8**	25.4 / **30.1**	48.6 / **51.7**	59.1 / **60.0**	ResNeXt-101

图5 PANet在COCO上的结果图，其中上侧为分割的结果对比表，下侧检测的结果对比表

3.3 目标检测用于视频分段：Rethinking the Faster R-CNN Architecture for Temporal Action Localization

论文链接：arxiv.org/abs/1804.0766

开源代码：无

录用信息：CVPR2018

论文目标：

本文借鉴了目标检测经典算法Faster R-CNN的思想，实现了对视频动作的定位与识别。

1. 通过multi-tower network和 dilated temporal convolutions方法，提高了动作定位精度（Temporal Action Localization）。

2. 通过扩展候选片段感受野，提高视频中的关键动作的识别效果(Temporal context)。

核心思想：

1.借鉴了Faster R-CNN的思想，将目标检测的思想从空间域转化为时间域。采用类似RPN层的结构对视频中动作做初步的定位与识别，然后采用类似RCNN的结构对动作实现精确定位及分类。

2. 由于视频时长从1s到200s不等，跨度非常大。本文通过multi-tower network和 dilated temporal convolutions解决了视频动作分布时间跨度较大难以定位的问题，同时通过扩展生成候选片段和动作分类的感受野，更有效的利用了视频上下文时序信息提升了动作识别效果。

算法概述

1） Temporal Action Localization

即：给定一段未分割的长视频，算法需要检测视频中的行为开始时间、结束时间及其类别。

action recognition与temporal action Localization之间的关系同 image classfication与 object detection之间的关系非常像。本文主要为了解决Temporal Action Localization问题，借鉴了Faster R-CNN的结构。

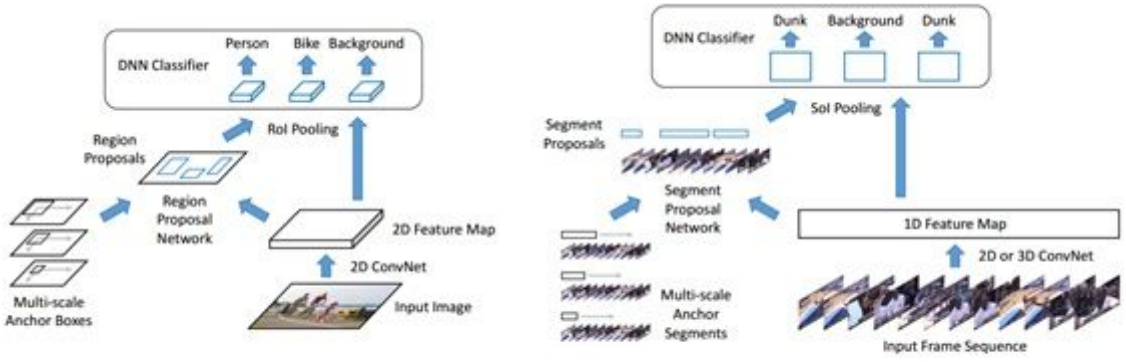


Figure 1: Contrasting the Faster R-CNN architecture for object detection in images [33] (left) and temporal action localization in video [15, 9, 16, 51] (right). Temporal action localization can be viewed as the 1D counterpart of the object detection problem.

与目标检测类似，它包含两个阶段：

1.生成动作候选区域

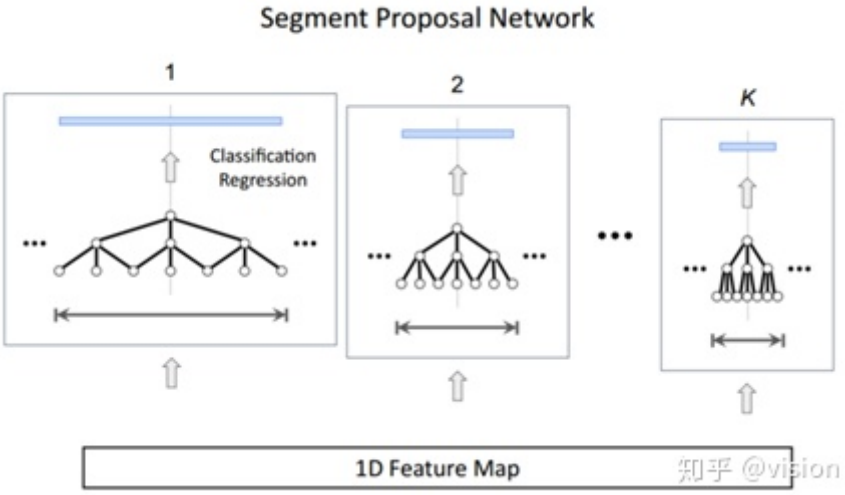
给定一组帧序列，通常通过二维或者三维卷积网络提取出一组一维特征图。之后，将该特征图传输给一维网络（类似RPN层，Segment Proposal Network），返回一组候选片段。

2.对候选区域分类

接着，对于每个候选片段，本论文计算动作类别的概率，并进一步对片段边界进行回归。在这一步，首先使用一维的SoIPooling（时间维度上做Pooling，类似空间维度的RoIPooling），接着使用 DNN 分类器来实现。

Segment Proposal Network：

文中作者采用了multi-tower 架构。每个 anchor 大小都有一个具备对齐后的感受野的相关网络。

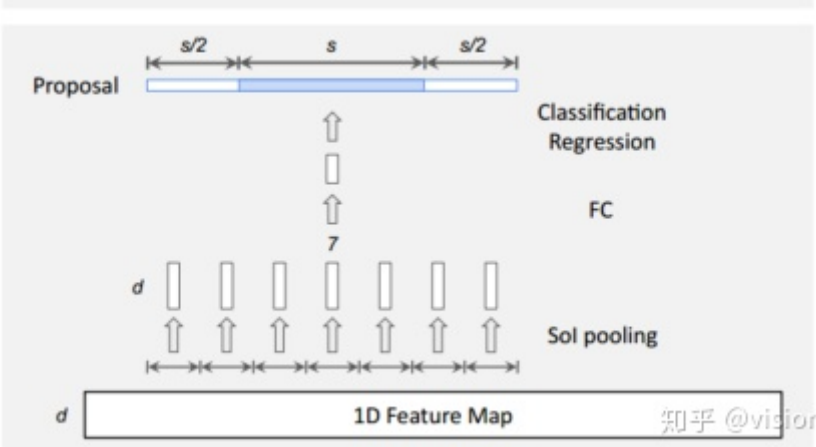


可以理解为对于尺度Feature map做相应Pooling或者空洞卷积（dilated temporal convolutions），作者经过实验后认为空洞卷积效果更好，时序定位更加准确。

2）如何有效利用时序上下文（temporal context）

作者在生成动作候选区域及动作种类识别时，认为动作前后信息对精确定位和动作分类有很大的意义。故强制将前面一段时间和后面一段时间加入候选区。

如图：

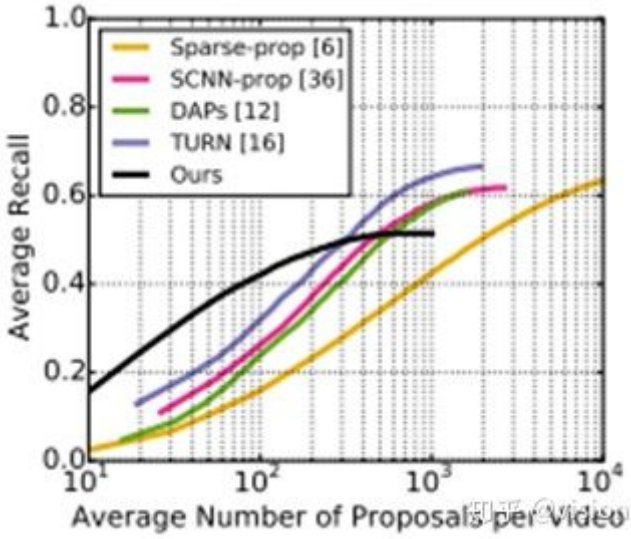


虽然候选区是中间的S区域，但是它依然会强制把前s/2和后s/2的区域加入proposal

文中表示： 推理过程中以0.7为阈值的IoU的proposals做NMS。

在训练过程中阈值大于0.7选为正样本，小于0.3为负样本。用Adam优化器，学习率0.0001

算法效果：



可以看到该方法在提取的片段较少时，有最好的表现。其他方法在提取片段较多时才会有较好表现。

3.4 从零训练目标检测网络： Mimicking Very Efficient Network for Object Detection

论文链接：openaccess.thecvf.com/c

开源代码：没找到

录用信息：CVPR2017

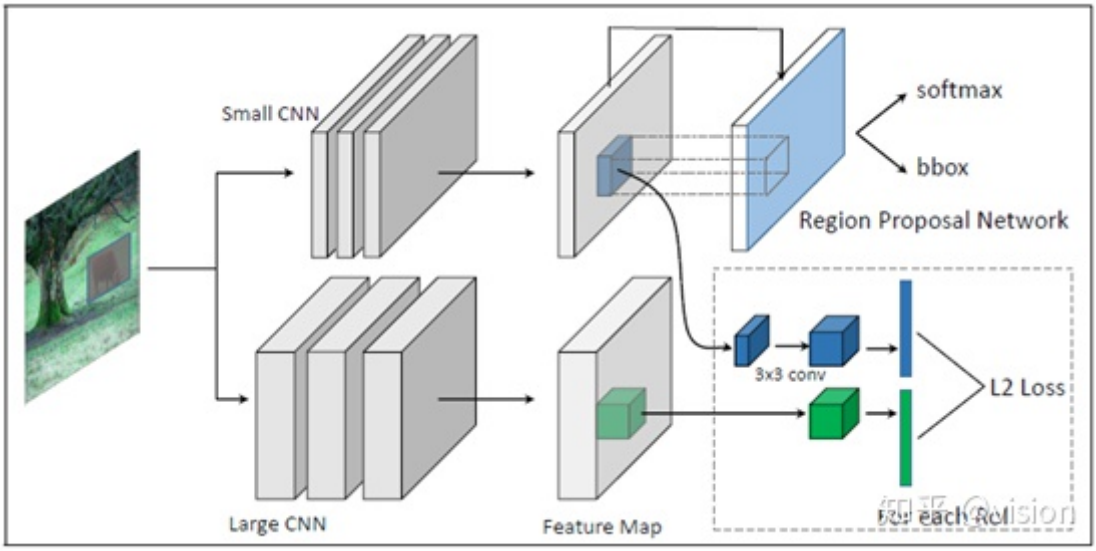
论文目标：

这篇论文与本综述（一）中介绍的DSOD的目标一致，都是解决如何有效地“从0开始训练检测网络”的问题。不同的是，DSOD旨在提出一种可以实现从头训练的检测网络结构。而本文借鉴知识蒸馏思想，在训练过程中，用训练好的大网络作为监督网络指导小网络的参数学习，小网络无需用预训练模型初始化。训练得到的小网络模型用于测试。

核心思想：

知识蒸馏，用大网络监督小网络的学习，常被用于解决分类任务中的模型压缩和加速问题。mimic论文作者将此思想应用于检测网络的学习任务。

论文以faster rcnn为例，介绍了论文的网络结构和训练过程。



上图示意了训练时的网络结构，上半部分是要训练的目标网络，通常是一个精简的小网络，后面加上RPN网络；下半部分是大网络，大网络用已经训练好的大检测网络模型进行参数初始化。论文将训练过程中的监督信息分成两部分：来自大网络的监督信息 mimic supervision 和训练数据的标注信息ground-truth supervision。

作者认为检测过程分为两个部分feature extractor和feature decoder，feature 对于大网络和小网络分别作为basenet的faster rcnn，二者的区别仅在于feature extractor不同（分别是大网络和小网络），而在提取到特征后的feature decoder是一致的。因此，论文将来自大网络的mimic supervision应用于feature extractor部分，而将ground-truth supervision应用于feature decoder部分。

论文将训练过程分为两个阶段，下面分别介绍这两个阶段的参数学习方法：

1）第一个阶段应用mimic supervision学习图中上半部分的目标小网络；小网络随机初始化参数，大网络用已经训练好的大检测网络模型进行参数初始化。训练过程中，固定大网络参数，使用以下loss指导小网络进行参数学习。

$$\mathcal{L}_m(W) = \frac{1}{2N}\sum_i \frac{1}{m_i}\|u^{(i)} - r(v^{(i)})\|_2^2,$$

其中，u^i为大网络的RPN输出的第i个候选框的特征，这个特征是从候选框的对用的feature map中直接提取得到的；v^i是小网络的RPN对应的第i个候选框的特征；变换操作r()是用于将v^i变换到与u^i相同的维度；m_i是第i个候选框特征的维度。

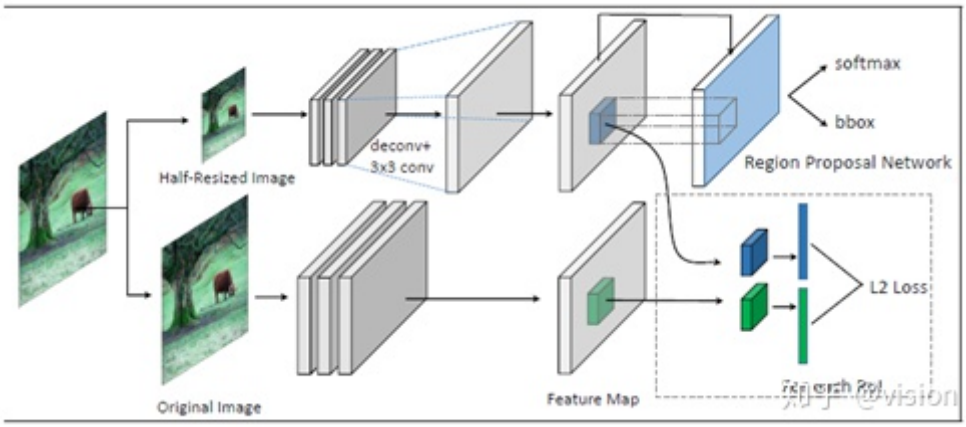
最小化以上loss的过程也就是将大网络的特征提取能力迁移到小网络的过程。

2）第二个阶段应用ground-truth supervision进行检测网络（detection network）的学习和RPN网络的精修。对于faster rcnn，检测网络跟RPN网络共享部分特征，对共享的特征层，使用第一阶段学到的参数初始化；对检测网络新增加的卷积层，随机初始化参数。

论文指出，第二阶段如果只用ground-truth supervision精修，会损害第一阶段学到的网络参数。因此，作者提出使用softmax层之前的那层特征（classification logits)进行mimic supervision。即上面公式中的u^i和v^i分别为大小网络的第i个proposal的classfication logits，这个阶段的loss计算包括两部分L_m(W)和L_gt(W)，如下所示，其中L_gt(W)代表分类和位置预测误差。

$$\mathcal{L}(W) = \lambda_1 \mathcal{L}_m(W) + \mathcal{L}_{gt}(W),$$

以上介绍了大网络到小网络的特征提取和检测能力迁移。除此以外，本论文思想还可用于接受大分别率输入图像的网络的检测能力迁移至小分辨率输入图像的网络，提升小分辨率输入图像的检测效果。如下图所示，基本思想即在最后一个feature map输出让RPN网络之前添加一个decovolution层增加feature map的分辨率，训练时使用支持大分辨率输入的网络进行监督。



算法效果：

下表中MR_-2为Miss Rate on False Positive Per Image，数值越小代表效果越好。其他细节可参见原论文。

Method	MR_{-2}	Parameters	test time (ms)
Inception R-FCN	7.15	2.5M	53.5
1/2-Inception Mimic R-FCN	7.31	625K	22.8
1/2-Inception finetuned from ImageNet	8.88	625K	22.8

Table 1: The parameters and test time of large and small models. Tested on TITANX with 1000×1500 input. The 1/2-Inception model trained by mimicking outperforms that fine-tuned from ImageNet pre-trained model. Moreover, it obtains similar performance as the large Inception model with only 1/4 parameters and achieves a 2.5× speed-up.

3.5 调研速度和准确率平衡的综述：Speed/accuracy trade-offs for modern convolutional object detectors

论文链接：arxiv.org/abs/1611.1001

开源代码：无

录用信息：CVPR2017

论文目标：

通过大量实验对比三种主流检测算法在各个情况下的表现。

核心思想：

在我们日常的视觉任务中往往有一个大家都会有一个经验，就是当提高一个网络的复杂度能得到更好的效果，而去精简一个网络的同时准确度自然会有所损失。更进一步来讲，相比于分类任务，目标检测任务在实际应用的时候更容易受到计算复杂度，内存要求等因素的限制。本文用非常大量详实的实验探讨了最常用的一些目标检测算法以及其中各个参数的设置对于目标检测的speed和accuracy的影响，以及对于如何以较好的性价比平衡速度和准确率给予了自己的结论。

看这篇论文的作者参与数量就能感觉到做完这些大量的实验，尤其是在像coco这样很大的数据集上，是非常耗时且花费精力的。在这里感谢作者的工作为没有资源和时间去做这些试验的人给出了很好的指导性的建议。我认为这些建议很多时候在实用的角度来讲贡献一点也不亚于提出一个新的方法。

首先先说一下试验的背景：用不同的深度学习框架，不同的网络初始化方法，不同的数据增强方法等等训练相同的算法，其性能和最终准确率都会有所不同。为了去掉这些因素的影响，作者统一在tensorflow的框架上实现了Faster R-CNN，SSD,RFCN算法，并以此为基础进行试验，保证了试验的公平性，其中有部分的参数初始化方法以及优化方法略有不同，在论文中均作了说明。同时为了保证基础特征的公平性和丰富程度，作者分别在VGG16, ResNet-101, Mobilenet, Inception-V2, Inception-V3, Inception-ResNet上进行试验，使得结果具有更高的普适性，同时也能从基础网络的角度看到了不同网络对于不同算法准确率和速度的影响。作者还对比了不同图像输入尺寸在以上这些网络上的表现。最后作者还对比了各个算法不同的预选框的设置方法，某些网络特定层的特征分辨率的不同等等设置对于网络的影响。可以说作者整个的试验设计非常系统，非常严谨。通过作者的试验结果，可以从各个角度对于当前目标检测的网络选择和参数设置有一个很直观的认识。

整个论文有非常多的试验结果和结论，如果对于哪方面感兴趣的读者建议还是去仔细查看原论文的相应部分的图表和论述。本文这里就不一一拿出来讨论，只抽出一部分试验结果进行描述。

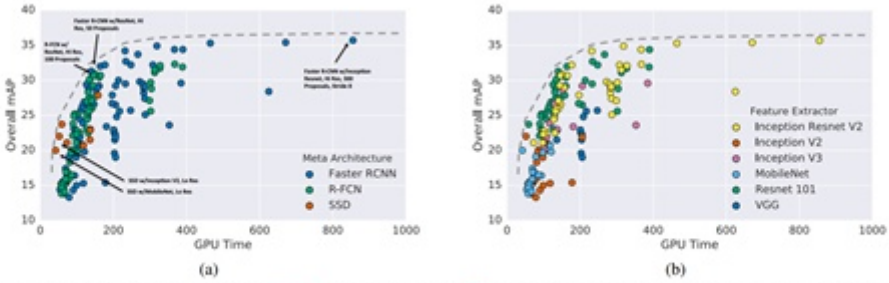


Figure 2. mAP vs gpu wallclock time colored by (a) meta-architecture and (b) feature extractor. Each (meta-architecture, feature extractor) pair can correspond to multiple points on this plot due to changing input sizes, stride, etc.

首先整体对比一下不同的网络接入三种检测算法后的表现，其中每种网络和算法的组合对又根据不同的图像输入尺寸，stride size画出了多个点。可以看到ResNet101，和Inception - ResNet101（在输入尺寸等比较小的情况下）是性价比比较高的网络，而Inceptionv2-3网络表现平平。也许像Inceptionv3这种网络是对于分类任务高度定制的，多尺寸的卷积和的融合影响了物体的位置信息的表达，不太适合于目标检测任务。

而检测算法方面，Faster R-CNN系列在mAP上稳稳胜出。但是速度相较于SSD R-FCN来讲也明显慢一些，但是通过一些参数的设置FASTER R-CNN也可以达到很快的速度。

作者对比了输入不同图像尺寸对网络的影响，可以看到，输入更大的图片确实可以明显的提升mAP但是也降低了网络的速度。作者还统计了不同算法和网络对于不同尺寸的目标预测的表现。结论是SSD对于小物体表现非常差，远低于其他算法。而在大物体上的表现几乎一样，甚至在一些网络上表现更好。所以如果检测任务当中大部分目标均为大物体，SSD绝对是一个非常好的选择，可以达到有快又准的效果。但是如果存在大量的小物体，为了提高召回率还是建议选择两阶段的检测算法比较好。具体试验结果请查看原始论文。

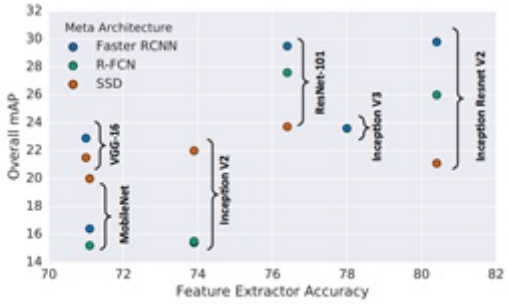


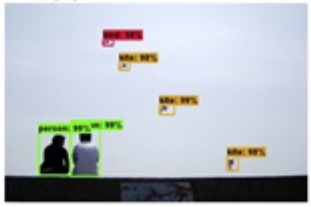
Figure 5. mAP vs. Top-1 Accuracy of the Feature Extractor on Imagenet (to avoid crowding the plot, we show only the low resolution models).

上图的实验结论非常超出我的认知，图中对比了不同网络在不同算法上的表现。可以看到SSD算法对于基础网络的特征非常不敏感，从结构、计算复杂度、深度差别很大的网络接入SSD后mAP的变化非常不明显。而两阶段的检测器，对于不同的网络mAP变化非常大。当基础网络复杂度比较低的时候SSD算法的性价比要更高一些。而另一方面增大SSD基础网络的复杂度，收益也相当有限，无法达到二阶段算法的准确率。可能SSD算法的主要瓶颈在于算法结构上，所以提高送入feature的质量并没有特别改善SSD的表现。

还有一个很有指导意义的结论是当要优化算法的速度的时候，减少proposal 的数量是一个很好的选择。可以很大程度的加快网络的速度同时精度受到的影响比较小。一个例子是当Faster R-CNN + Inception-ResNet组合时，只用50个proposals 与用300个proposals相比，mAP相差无几，但是速度提升了三倍之多，这里原文中有更详细的讨论。

结论：

(c) FRCNN+Resnet101, 100 proposals (d) RFCN+Resnet10, 300 proposals



(e) FRCNN+IncResnetV2, 300 proposals

本文通过大量实验对比了影响目标检测算法速度和准确性各项因素的作用，可以帮助从业者在实际应用中选择更合适的目标检测算法。