对象检测——基于深度学习

丁雨晖，zjhndyhnba@hotmail.com

1. 简介

对象检测在计算视觉领域是研究的热点。对象检测的目标是将图像中的对象进行定位和分类，这与视频分析和图像理解有着密切的关联。传统的对象检测方法，主要基于特征提取和模板匹配。近年来随着深度学习的发展，新的基于深度学习的方式开始涌现，相比传统方法取得了大幅度的提高，出现了从最初的两阶段模型到当下的端到端模型。

对象检测使用的主流数据集包括PASCAL VOC以及COCO。PASCAL VOC总共有大约一万张图片作为训练集和验证集，其中包含了20个分类，该数据集标注了图片中物体的边框（bounding box）以及分类信息。COCO数据集可用于不同的挑战，包括标题生成、对象检测、关键点检测以及图像语义分割。该数据集包含超过120000张图片用于训练和验证，超过四万张用于测试，其中包含80个分类。

下文将介绍深度学习下对象检测的主要框架，并对SSD框架做重点讲解。

2. 基于深度学习的对象检测

深度学习方法主要分为两大类，分别是基于候选区的方式和基于回归分类的方式。基于候选区的方式是两阶段的，首先通过某些启发式算法选取一系列候选框，通常数目在几百或上千，完成候选框提取后对所有候选框缩放到同一尺寸，然后通过CNN网络进行分类。此外还需要通过回归对候选框进行精修，以逼近真实值。基于回归和分类的方式是端到端的模型，输入图像后可以直接对位置进行回归和分类。图2-1展示了这两类框架以及相应的论文。

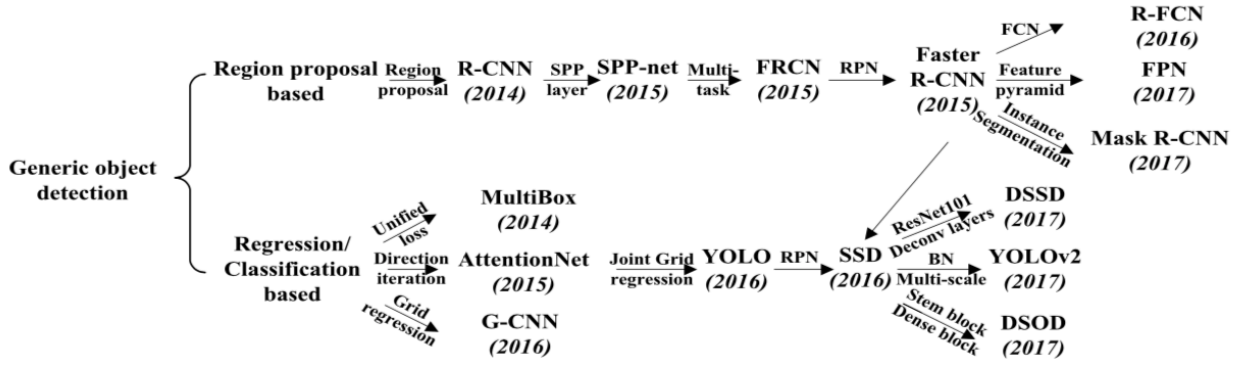


图2-1 两类对象检测框架

2.1. 基于候选区的对象检测

基于候选区的框架是一个两阶段的过程。某种程度，这是一种符合人类大脑注意力机制的方式，它先通过粗略的扫描整个场景，然后聚焦到感兴趣的区域（Regions of Interest: RoIs）。具有代表性的算法包括R-CNN、SPP-net以及Fast(er) R-CNN等。

2.1.1.R-CNN

R-CNN是对象检测领域中具有里程碑意义的研究成果，该算法第一次使用深度学习用于对象检测，在PASCAL VOC数据集上的平均精度均值（mean Average Precision: mAP）达到了53.3%，比之前最好的算法提升了30%。如图2-2展示了该算法的流程，可以分为如下三个阶段：

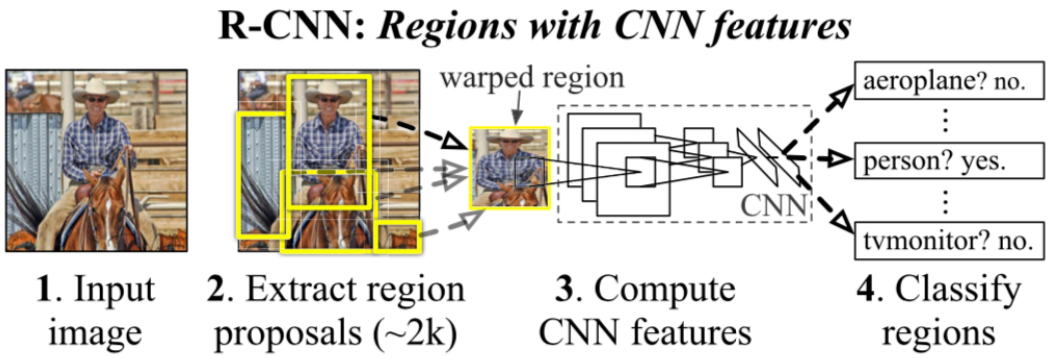


图2-2 R-CNN流程图

a) 候选区域生成：R-CNN采用selective search生成约2000个候选区域。selective search算法先生成一个区域集合S，通过颜色和纹理相似度对相邻区域进行匹配和合并，并将合并区域加入到集合R，然后反复用集合R与集合S中的区域匹配合并，将新区域加入集合R，旧区域从集合S中移除，直至集合S为空集。

b) 基于CNN的深度特征提取：得到候选区域后，先将这些区域缩放成同一个尺度，然后使用CNN网络提取到4096维度的特征。

c) 分类和定位：使用线性SVM作为分类层，为每个区域在类别集合上打分，包括背景类。这些区域使用边框回归（bounding box regression）进行进一步调整，以逼近真实值。具体而言需要训练得到一组参数对图片的输出特征进行映射，得到一组修正值，用来修正候选区。最后通过非最大化抑制（Non-Maximum Suppression: NMS）生成最终的边框。

尽管相比传统方法带来巨大提升，R-CNN有许多缺点。由于全连接层的存在，CNN网络的输入必须固定到统一尺度。这样的形变操作可能会影响到识别的准确率。因此有了SPP-Net的提出。

2.1.2.SPP-Net

SPP-Net的提出者认为对每个候选框区域进行变形会使图片信息丢失，从而影响识别准确率，但是CNN特征提取模块最终必须提取到全连接层用于分类和回归，因此需要一种方法既不改变原图的尺寸，又能提取到统一尺度的全连接层，于是提出了空间金字塔池化算法。如图2-3是池化过程的示意图。

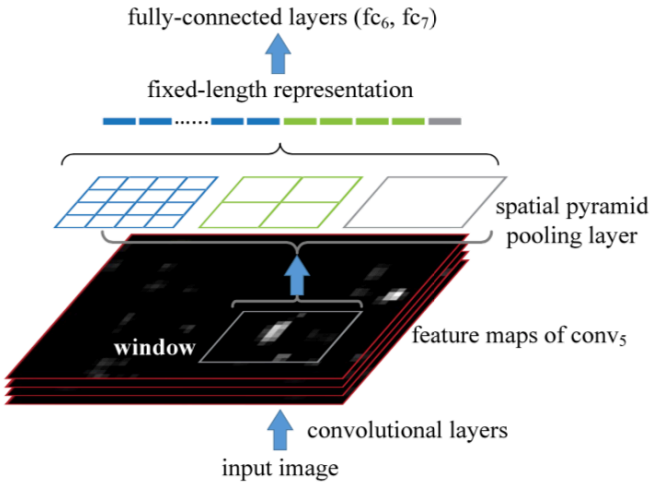


图2-3 SPP-Net架构图

图像进过CNN模块得到大小不同的特征图（feature map），然后对每个特征图分别划分成4*4、2*2以及1*1大小的单元，然后对每个单元进行池化，这样不管特征图的大小，最后经过池化都统一到21的向量。

2.1.3.Fast R-CNN

SPP-Net从精度上对R-CNN进行了提升，但速度上并没有，它没有解决R-CNN训练时多阶段的问题，包含了特征提取、网络微调（fine-tuning）、SVM训练以及边框回归的过程。Fast R-CNN的提出了从候选区域作为输入，分类和定位作为输出的端到端训练方式。如图2-4所示为Fast R-CNN的架构图，其中RoI池化层是SPP层的特例，输出的分类损失和回归损失进行了合并，即论文中所说的多任务损失（multitask loss），从而达到了端到端的目的。

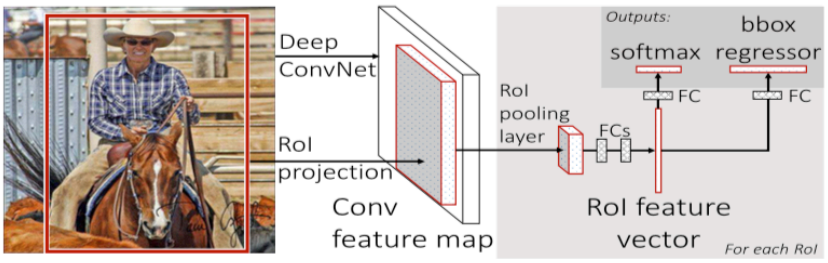


图2-4 Fast R-CNN架构图

公式2-1展示了多任务损失函数，其中$L_{\text{cls}}(p,u)=-\log p_u$计算了在真实类别$u$上的对数损失，$p=(p_0,\cdots,p_C)$表示对象在$C+1$个类上的概率，其中第0个类代表背景。$t^u=\left(t_x^u,t_y^u,t_w^u,t_h^u\right)$表示对象真实边框与候选框的偏移，$v=\left(v_x,v_y,v_w,v_h\right)$表示预测的偏移值，指示函数$[u\geq 1]$用于去除背景区域，因为背景不需要计算定位损失。

$$L\left(p,u,t^u,v\right)=L_{\text{cls}}(p,u)+\lambda[u\geq 1]L_{\text{loc}}\left(t^u,v\right)$$

公式2-1 多任务损失

2.1.4.Faster R-CNN

对于以上的几种方法，候选区域生成过程是制约计算效率的瓶颈。Faster R-CNN的作者提出了使用候选区域生成网络（Region Proposal Network: RPN）用于生成候选框。如图2-5是Faster R-CNN的架构图。

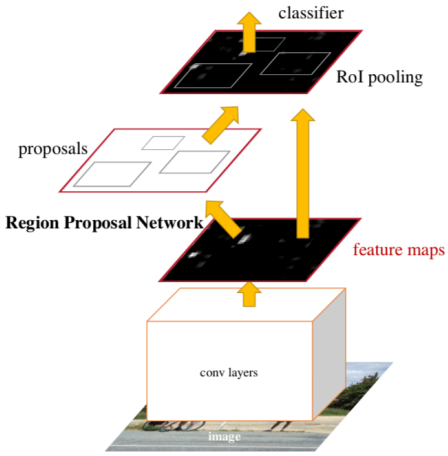


图2-5 Faster R-CNN架构图

Faster R-CNN的作者提出了锚点框（anchor box）的概念。如图2-6中使用3×3的滑动窗口卷积特征图，每个滑动窗口生成k个锚点框。对于宽高分别为W和H的特征图，总共生成WHk个某点图，k的大小由缩放尺度和纵横比决定，文中设定3种缩放尺度以及3种纵横比，因此每个位置有9个锚点框。RPN的损失函数与公式2-1类似，不同的是类别只有两种，即背景和非背景。

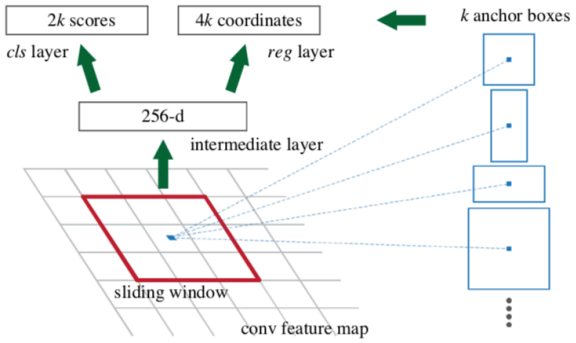


图2-6 RPN架构图

2.1.5.R-FCN

Fast R-CNN中的RoI池化层将网络分割成两部分，RoI池化层之前的是共享的全卷积（fully convolutional）网络，之后的是不共享子网络，如果RoI池化层后的网络过于深会导致计算效率下降。因此，R-FCN的作者认为需要将RoI池化层往后挪，以完全去除后面的不共享网络。在基于ResNet-101网络的对象检测算法中，R-CNN的非共享层层数为101，Faster R-CNN为10，而R-FCN为0。但这也存在问题，随着RoI池化层越来越靠后，平移可变性（Translation variance）被打破。所谓平移可变性指的是随着网络变深，特征图变小，小特征图对候物体的偏移会感知不到。也就是RoI越靠前，定位越准确，越靠后，定位越不精确。因此作者提出了位置敏感得分图（position-sensitive score map）。如图2-7所示为R-FCN的架构图。

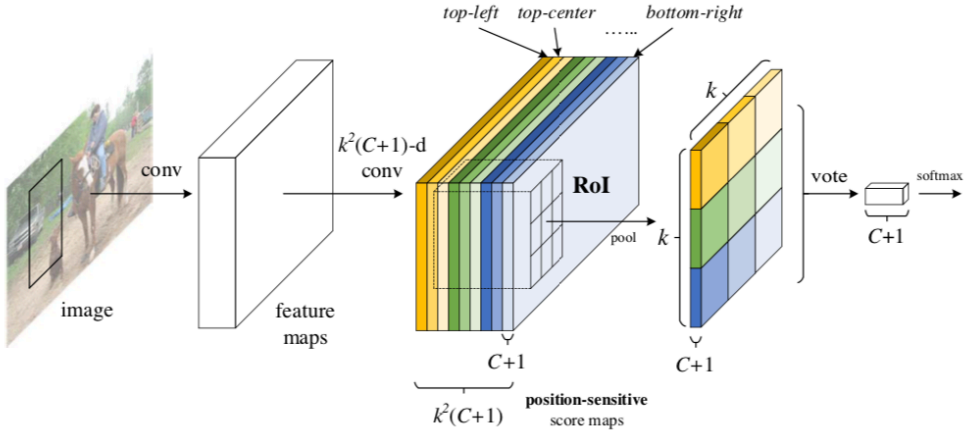


图2-7 R-FCN架构图

R-FCN通过全卷积网络得到通道数为$k^2(C+1)$的RoI层，其中C为类别数，(+1)为背景。候选区域被划分成了$k \times k$个单元，经过池化得到大小为$k \times k$的特征图，通道数为$C+1$，在特征图的每个位置都给出了$C+1$个类别的得分，通过投票机制判定该候选框内物体所属类别。直观上的理解就是候选框包含对象的成分越多，判定为该对象的概率越大。除此之外，边框回归仍然需要，与用于分类的特征图对应，该模型还会有一个大小为$k \times k$，通道数为4的特征图，通过计算平均值将$k \times k$个单元聚合为一个，得到长度为4的向量，即边框回归所需的4个偏移量。R-FCN与之前的模型最大的不同在于其不再保留全连接层，而是用了全卷积的网络。图像经过卷积得到的热区图的每个位置反映出对物体的敏感程度，表现在得分的高低。不管是结果还是直观理解上，这种方式相比含有全连接的网络能保留更多的位置信息。

2.2. 基于回归和分类的对象检测

基于候选区域的框架由若干相关阶段组成，包括候选区域生成，CNN特征提取，分类和边框回归，这些阶段通常是单独训练的。即使在最近的端到端模块Faster R-CNN中，仍然

需要替代训练来获得RPN和检测网络之间的共享卷积参数。因此，处理不同部分所花费的时间成为实时应用程序的瓶颈。单阶段的检测算法中，YOLO和SSD是最具代表性的算法。

2.2.1.YOLO

YOLO使用顶层的多个特征图预测多个类别的置信度以及边框。YOLO的基本思想如图2-8所示，YOLO将输入图像分成$S \times S$网格，每个网格单元负责预测该网格单元中心的对象。每个网格单元预测边框及其对应的置信度分数。置信度分数可以定义为$\Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$，代表有多少可能存在一个物体（$\Pr(\text{Object}) \geq 0$），同时它的预测具有一定的置信度（$\text{IOU}_{\text{pred}}^{\text{truth}}$）。同时，无论边框的个数，每个网格都需要预测$C$个类别的条件概率（$\Pr(\text{Class}_i | \text{Object})$）。

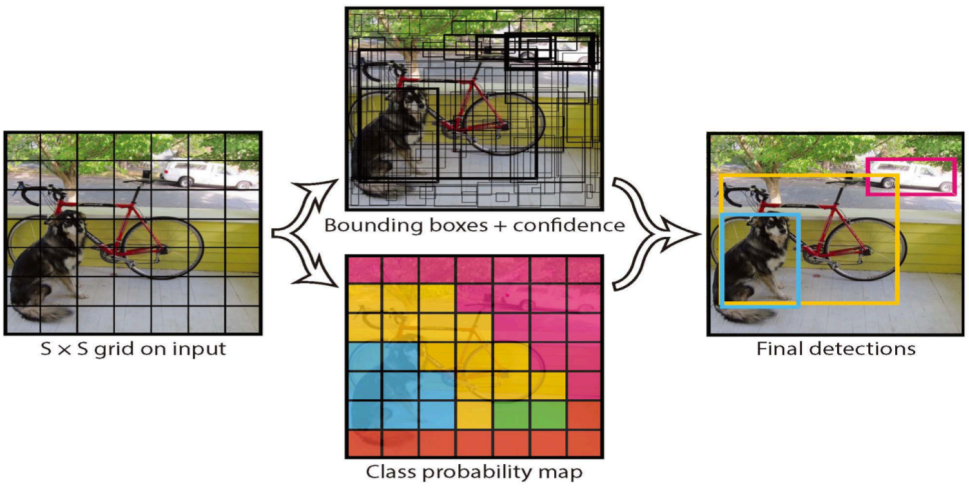


图2-8 YOLO原理示意图

在测试时，通过将单个框置信度预测与类别条件概率相乘，实现每个框的特定类别置信度分数，如下所示：

$$\Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} * \Pr(\text{Class}_1 | \text{Object}) = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

公式2-2 类别置信度

公式2-2表明，框中特定类对象的概率以及预测框与对象之间的适配性都被考虑在内。

图2-9为YOLO网络架构图，对于$S \times S$的网格，每个单元格需要预测B个边框，每个边框包含四个坐标值和一个置信度，同时还要预测C个分类，因此输出大小为$S \times S \times (B \times 5 + 10)$，原文中，$B$取2，$C$取20，S取7，从而得到图中$7 \times 7 \times 30$的输出。

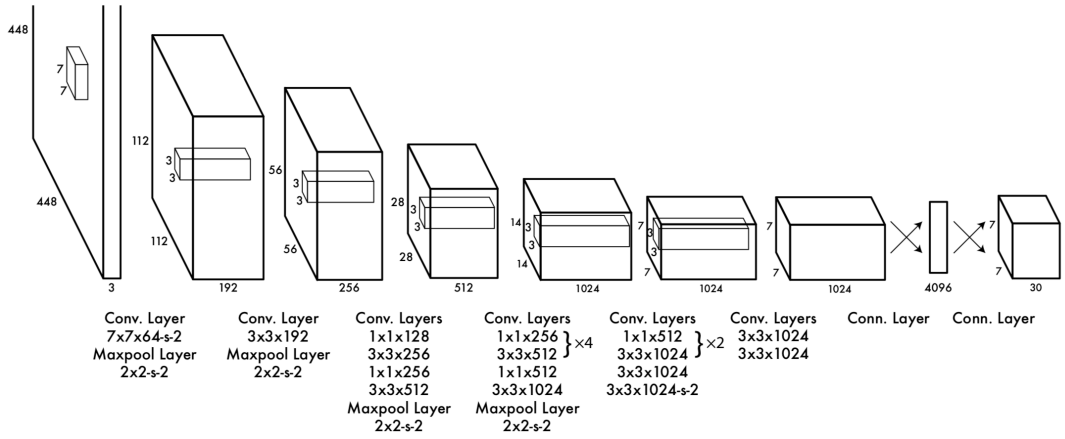


图2-9 YOLO架构图

YOLO网络可以以45 fps实时处理图像，简化版Fast YOLO可以达到155 fps，结果比其他实时检测器更好。此外，YOLO在背景上产生较少的误报，这使得与Fater R-CNN的结合变得可能。作为提升版本，YOLOv2采用了多种策略，如批标准化、锚点框和多尺度训练等，取得了不错的效果。

2.2.2.SSD

YOLO难以处理群体中的小物体，这是由于对边框预测强加的空间限制造成的。SSD的作者受到多重框（MultiBox）和多尺度表征的启发，利用一系列不同纵横比和尺度的先验框来离散化边框的输出空间。为了处理具有各种大小的对象，网络融合了具有不同分辨率的多个特征图的预测。

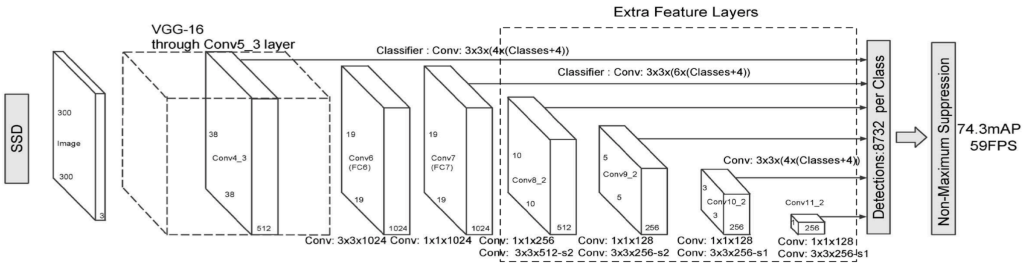


图2-10 SSD架构图

如图2-10所示，与YOLO只有单个尺度的输出不同，SSD有多个不同尺度的特征图作为输出。特征图的每个像素都是一个锚点，锚点对应了一系列的先验框（default box）。

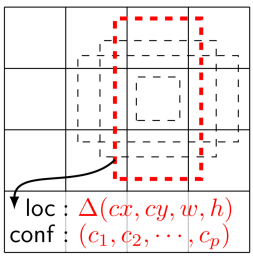


图2-11 MultiBox

如图2-11所示，一个像素对应多重的先验框，每个先验框都与ground truth产生一个位置偏移，从而进行边框回归，同时会输出对每个类别的置信度，用于计算分类损失。这里的先验框由不同的纵横比和尺度组合而成，纵横比和尺度是按经验选取的。某种意义上，先验框与R-CNN中的候选区域是类似的，R-CNN中的候选区域通过算法产生，个数较多，而先验框凭经验产生，数目较少。现在假设特征图大小为$m \times m$，每个像素点的候选框个数为k，总的类别数目为C，因此对于每个框需要$C+1$（+1为背景类）个数值进行分类预测，4个数值用于预测，则总的输出数为$m \times m \times (k \times (C+1+4))$，其中$k \times (C+1+4)$为特征图的通道数。

SSD在训练过程中使用了hard negative mining算法。在SSD算法中，先验框的个数多达8000以上，而其中绝大部分与ground truth的交集很小，IoU（Intersection-over-Union：两个矩形交集占总面积的比例）是衡量匹配程度的指标，理想情况是1，即完全重合。对于那些IoU大于一定阈值的先验框定义为正样本，而小于该阈值的为负样本（即背景类）。训练时，在保留所有正样本的情况下，以正样本3倍数目保留负样本，其中负样本按照其背景类的置信度降序排序，选择置信度前N（正样本数目乘以3）的作为训练样本。

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

公式2-3 SSD损失函数

公式2-3展示了SSD的损失函数，该公式与Fast R-CNN的损失函数类似，划分为分类损失和定位损失。其中N是正样本数目，$y \in \{0,1,\cdots,C\}$是先验框所匹配的ground truth的真实类别，用0代表背景类（无匹配），l为预测的边框，g为ground truth，α用来平衡分类和定位损失。

$$L_{loc}(y,l,g) = \sum_{m\in\{cx,cy,w,h\}} \mathbb{I}\left[y>0\right] \mathrm{smooth}_{\mathrm{L1}}(l^m - \hat{g}^m)$$

$$\hat{g}^{cx} = \left(g^{cx} - d^{cx}\right)/d^w \quad \hat{g}^{cy} = \left(g^{cy} - d^{cy}\right)/d^h$$

$$\hat{g}^w = \log\left(\frac{g^w}{d^w}\right) \quad \hat{g}^h = \log\left(\frac{g^h}{d^h}\right)$$

公式2-4 SSD定位损失函数

公式2-4展示了定位损失函数，其中$\mathbb{I}\left[y>0\right]$指示只保留类别大于0，即只保留正样本用于定位损失。d代表先验框，$\hat{g}$是ground truth相对先验框的偏移，cx和cy为边框的中心，w和h是宽高，中心点偏移使用中心点距离偏移的百分比衡量，宽高的偏移使用对数缩放比来衡量。最后使用L1 smooth loss来衡量预测与ground truth的差距。

公式2-5展示了分类损失函数，即负对数似然函数。其中$\hat{c}$为预测概率，即softmax层的输出，$\hat{c}^y$代表预测为类别y的概率。

$$L_{conf}(y,c) = -\sum_{i\in Pos} \log\left(\hat{c}_i^y\right) - \sum_{i\in Neg} \log\left(\hat{c}_i^0\right)$$

公式2-5 SSD分类损失函数

SSD的作者发现，越多的特征图以及越多的先验框能获得更高的准确率。因此SSD也开拓了一个新的分支，之后一些学者开始在特征图上进行做文章。一般而言，越深的特征图语义性强，但分辨率低，适合检测大物体；越浅的特征图分辨率高，但其语义性制约了检测小物体。SSD算法也因此对小物体不够敏感。一些学者尝试使用一些超分辨率手段产生尺度更多样的特征图，即能保持较高的分辨率，又能保留语义性，从而增强对小物体的敏感性。如DSSD使用反卷积手段进行超分辨率，获得了比SSD更好的检测效果，但其性能大幅度下降，无法达到实时性。此外，STDN算法尝试使用尺度变换模块（scale-transferable module）产生不同尺度的特征图，用于检测不同大小的物体，其超分辨率的方式使用了无参数的pixel shuffle算法。相比DSSD和SSD，STDN兼顾了精度和速度。

3. 总结

由于其在处理遮挡，尺度变换和背景切换方面的强大学习能力和优势，基于深度学习的目标检测近年来一直是研究的热点。其中R-FCN摒弃了全连接层，在全卷积层上做分类，为后续的对象检测提供了重要的借鉴。SSD算法利用不同尺度的特征图进行逐像素地进行检测，同时生成固定的先验框，大大提升了检测性能，在精确度和实时性上达到了平衡。在未来，如何简化网络来提升性能，同时保证精确度，以及小物体检测的提升仍是相关学者研究的重点。

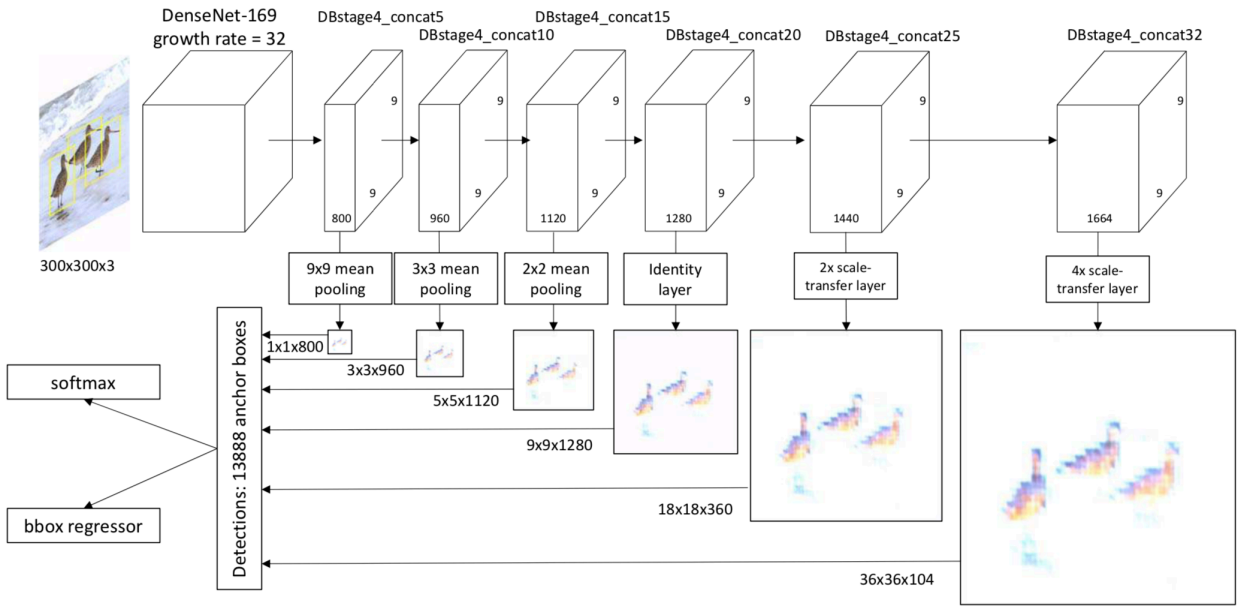
参考文献

- R. Girshick *et al.*, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. CVPR*, 2014, pp. 580–587.
- Ross Girshick. 2015. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV) (ICCV '15). IEEE Computer Society, Washington, DC, USA, 1440-1448. DOI=http://dx.doi.org/10.1109/ICCV.2015.169
- Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'15), C. Cortes, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 1. MIT Press, Cambridge, MA, USA, 91-99.
- J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016

- K.He, X.Zhang, S.Ren and J.Sun.Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014 .
- J. Redmon *et al.*, “You only look once: Unified, real-time object detection,” in *Proc. CVPR*, 2016, pp. 779–788.
- W. Liu *et al.*, “SSD: Single shot multibox detector,” in *Proc. ECCV*, 2016, pp. 21–37.
- C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659*, 2017.
- *Peng Zhou, Bingbing Ni, Cong Geng, Jianguo Hu, Yi Xu*; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 528-537

附录（论文《Scale-Transferrable Object Detection 》复现）

该论文是SSD框架的延伸，其整体框架保持不变，使用了hard negtive mining的训练方式，以及损失函数沿用公式2-3，公式2-4以及公式2-5。主要的不同在于网络结构



Scale-transferable net 架构图

Scale-transferable网络（STDN）的架构图如上图，其骨干网络使用Dense-Net，并抽取最后一个dense block中的特征图，进一步进行缩放。其中缩小使用卷积加池化操作，放大使用transfer模块。最后用缩放后的特征图进行回归和分类。

$$I^{SR}_{x,y,c} = I^{LR}_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, r \cdot mod(y,r)+mod(x,r)+c \cdot r^2}$$

超分辨率公式

如上公式所示为超分辨率公式，I^{SR}为超分辨率图像，I^{LR}为低分辨率图像，r是放大倍数，x和y分别为图像横竖两个方向的坐标，c为通道数。

SSD直接利用深层的几个卷积层进行检测，在小物体上效果并不好。原因是较浅的特征图语义性不好，难以检测物体；而较深的卷积层语义性强，但由于其低分辨率难以检测小物体。DSSD在SSD基础上加入反卷积生成高分辨率，同时具有强语义性的特征图，从而得到巨大的准确率提升，但过深的网络导致其无法进行实时性检测。STDN在两者上进行了折中。它对于语义性较低，分辨率较高的浅层特征图进行卷积，生成语义性强，分辨率低的特征图用于检测。对于语义性高，分辨率低的深层特征图，它使用transfer模块进行超分辨率，从而在保留语义性的同时，增大了分辨率。这样STDN兼顾了大小物体的检测，同时保证实时的检测速度。

具体实现使用PyTorch，训练框架沿用了 https://github.com/amdegroot/ssd.pytorch 的实现，本人在此基础上实现了网络架构。项目链接为 https://github.com/IntoxicatedDING/stdn 。