

# 视觉目标检测和识别之过去，现在及可能

[IMG_257](https://www.zhihu.com/people/li-xi-hua-93)[李习华](https://www.zhihu.com/people/li-xi-hua-93" \t "https://zhuanlan.zhihu.com/p/_blank)

9 个月前

计算机视觉中目标检测、跟踪、识别是最基本的几个task，尤其又以检测最为重要和基础。同时基本上所有的检测任务都需要在给出物体的bounding box之后，给出物体的类别（也就是给识别物体），所以文章中不区分检测和识别任务。笔者从事计算机视觉中目标检测相关的工作还比较多，将自己看到的，学到的，以及相关的一些理解总结如下。帮助自己进行梳理，也希望对后来者有帮助。大致将目前学术及和工业界出现的目标检测算法分成3类：

1. 传统的目标检测算法：Cascade + Harr / SVM + HOG / DPM 以及上述的诸多改进、优化；

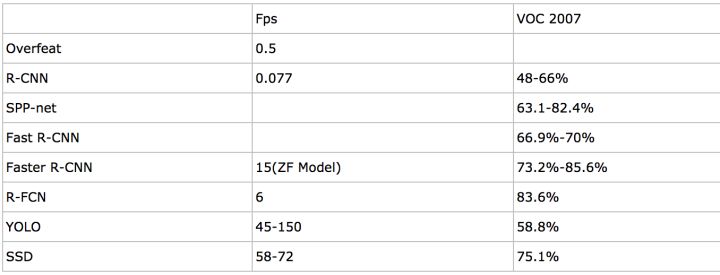
2. 候选窗+深度学习分类：通过提取候选区域，并对相应区域进行以深度学习方法为主的分类的方案，如：RCNN / Fast-RCNN / Faster-RCNN / SPP-net / R-FCN 等系列方法；

3. 基于深度学习的回归方法：YOLO / SSD / DenseBox 等方法；以及最近出现的结合RNN算法的RRC detection；结合DPM的Deformable CNN等

对于检测的目标，早期工业界关注的主要是人脸，人，车这些对监控、交通等领域非常重要的目标，到现在为止，计算机需要更全面的理解场景，检测的类别扩展到了生活的方方面面。

文章会针对上述的方法进行简单的介绍，以及表达一些笔者认为这种演进趋势背后的原因。总结起来，实际上所有的方法都可以概括成：**候选窗口 + 分类or回归**；逻辑上滑窗也是提取候选窗口的一种方式。

基于深度学习的方案的计算效率和进度大致对比如下：



*注：以上数据摘自论文*

下面对上面的3类目标检测的算法进行说明：

**传统的目标检测算法：**

传统的做目标检测的算法基本流程如下：

1. 使用不同尺度的滑动窗口选定图像的某一区域为候选区域；

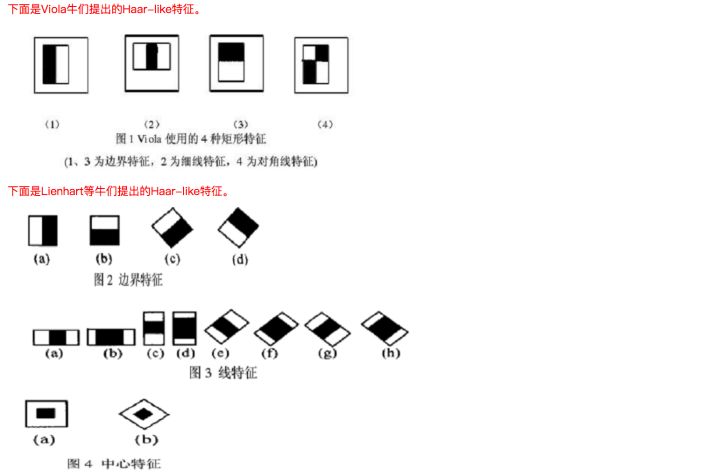
2. 从对应的候选区域提取如Harr HOG LBP LTP等一类或者多类特征；

3. 使用Adaboost SVM 等分类算法对对应的候选区域进行分类，判断是否属于待检测的目标。

2001年viola的cascade+harr进行人脸检测的方案取得了很好的正面人脸的检测效果；其实这之前通过普通的颜色特征，形状特征也可以处理一些基本的目标检测问题，并且这类方法至今在工业检测的场景仍然在使用；下面我们对viola的cascade+harr人脸检测的方案进行详细的介绍，并为后续的方法埋下伏笔。

假设我们现在已经获取了一个图像区域，我们要判断这个区域是不是人脸，应该怎么做呢？那我们首先从这个区域提取特征，然后将特征送给分类器进行分类。特征呢？在viola的方案中主要使用的是harr特征，描述两个图像block之间的明暗对比；而分类器呢，最好是又快又好，viola提出的方法是Adaboost，通过弱分类器的组合和级联完成分类任务；假如一个640\*480大小的图片，滑窗时每次移动3个像素的话，总体上也有10w量级的滑窗，再加上由于目标尺寸的变化，需要对图片进行缩放或者扩大，总体滑窗的数量又会上升，因此对分类器的效率要求非常高。所以级联的cascade方法很好的解决了这个问题，在初期使用很少的计算量过滤掉大部分的候选窗口，对于越难进行分类的样本就计算更多的特征来进行判断，直到所有的候选窗口的分类完毕。后续会针对检测的结果进行重叠窗口以及不同尺度图像上窗口的融合。

viola最早提出来的harr特征主要是水平方向和竖直方向，后续又出现了非常多的变化；adaboost方法在后续也出现了非常多的变种，这里将harr特征的变化以及adaboost的变种进行一个罗列：

基本上可以看到，以上harr的变种都是改变明暗对比的计算方法。还有一些改进：如将harr特征进行组合的joint harr feature，如利用pixel明暗对比进行组合的joint pixel harr feature；基本逻辑就是获得更为全面的特征描述能力，以及提高计算速度。

这对adaboost也有非常多的改进，以及training方法的改进也很多，这里对adaboost的改进进行罗列，感兴趣的朋友可以查看：

vector boosting

output code to boost multiclass learning

multiple instance boosting（MIL等）

multi-class boosting

floatboost learning

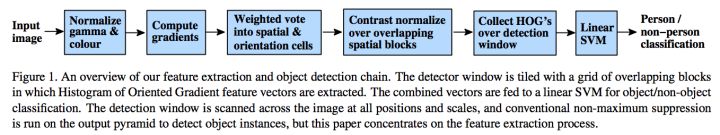
multi-class adaboost

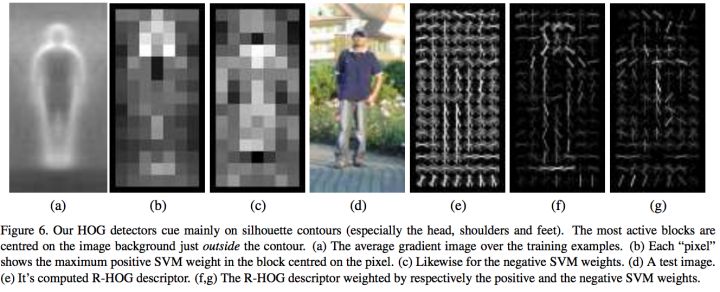
textonboost

等等

改进非常多，不过基本逻辑依旧是提升模型的表达能力；traning的改进在此不做罗列，总结起来基本上是：难分负样本的处理，trainning效率等。

使用SVM+HOG的方法进行行人检测方法和上述方案基本一致，不过针对行人检测的场景，harr通过区域的明暗对比计算的特征显然不能够太好的描述；而HOG特征本身含有一定的位置特性，正好可以描述人体如头部，四肢等不同的部位。方法的流程如下：

HOG特征能够很好的描述人体的轮廓特性，大致可以有个直观的感受参考这个图：



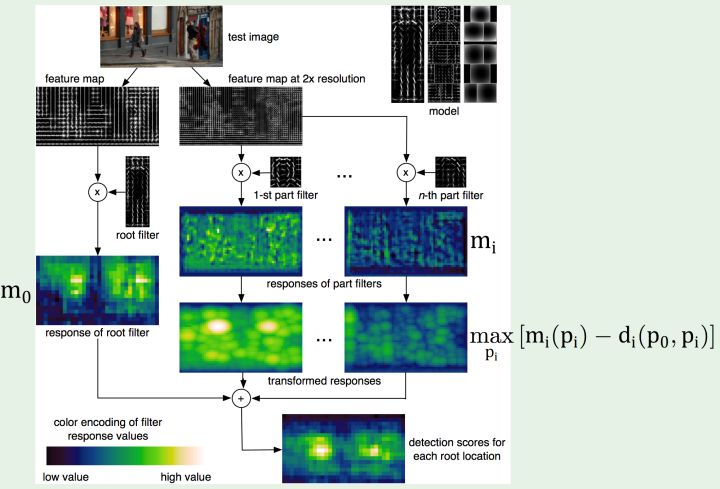
但上面的方案有个问题，实际上人的正面和侧面从视觉轮廓上来看，差异非常大；比如车辆也一样，从正面看，主要是挡风玻璃和前脸部分，从侧面看则是车门和轮廓部分；所以出现了后来非常出名的dpm算法。dpm算法的思想如下：

1. 物体都是由不同的part组成的，由于各种variations导致物体整体的视觉效果不一样，但是有些part的变化实际上不大，因此可以训练针对不同part的检测模块；

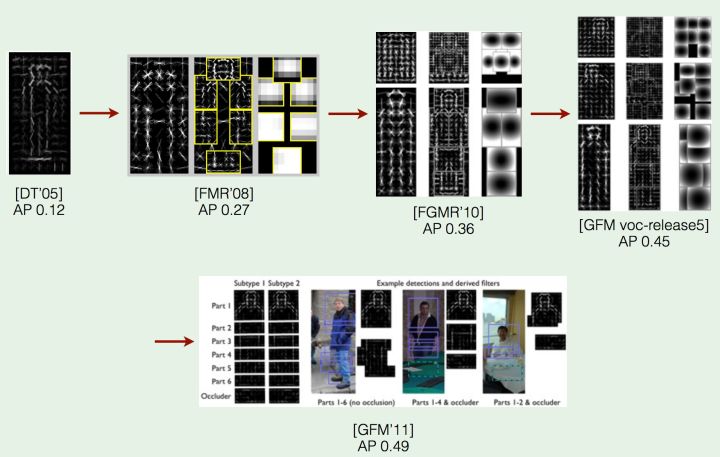
2. 一个物体，不同的part之间实际上是存在天然的拓扑结构的关系，比如人体头大部分情况都位于躯干的上面，车大部分情况轮子的部分都在地面上等等。也就是说这些不同的part之间的距离和位置关系符合一个标准的分布，某种位置关系非常多，而有些位置关系存在但很少，有些则根本不存在；

3. 有了各个部件，有了位置关系，就可以将不同的位置关系对目标物体的贡献看待成一个权重，最后由权重求和得到是否是需要检测的目标。

整体流程参考：

root filter 和 n个part filter进行最终的加权，得到检测目标。

dpm算法本身的进步roadmap可以参考这个：

传统的目标检测方法介绍到这里，而实际上后续的基于深度学习的目标检测方案多多少少参考了前面的工作，以及在深度学习目标检测的框架中也在结合以上方法中的思想，后面一一介绍。

**候选窗+深度学习分类的目标检测方案：**

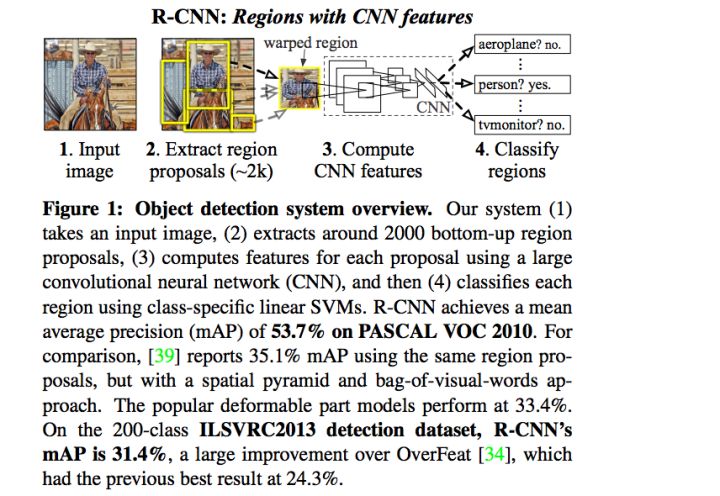
滑窗有一个问题，就是产生的候选窗口数量巨大，所以有各种region proposal的方案出来，最根本的需求就是在减少可能窗口的miss的情况下找到更少的候选窗口；从这个角度讲，金字塔+滑动窗口的方案实际上是一个100%召回的方案。

使用深度学习方法进行分类能够更好的提取特征，以及模型能够具备更强大的描述能力；所以基本上对深度学习的使用也慢慢的从当成特征提取进步到特征提取与分类、回归等问题结合起来的end-to-end的方式。

先介绍rgbd大神的系列工作rcnn -> fast rcnn -> faster rcnn；以上工作非常有连续性，建议读者读原论文研读。这个链接有一些说明：

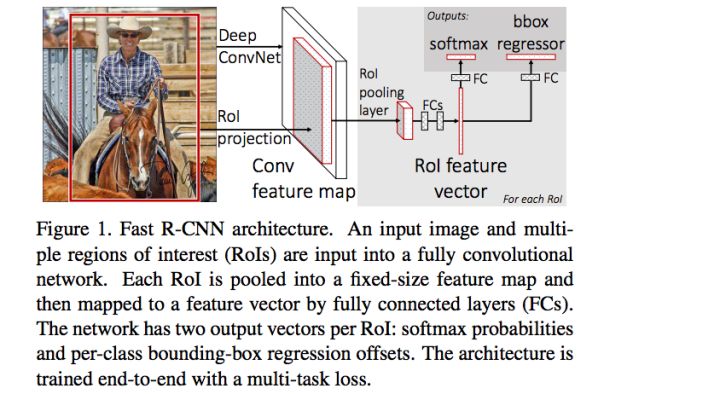
[RCNN, Fast-RCNN, Faster-RCNN的一些事](https://link.zhihu.com/?target=http://closure11.com/rcnn-fast-rcnn-faster-rcnn%E7%9A%84%E4%B8%80%E4%BA%9B%E4%BA%8B/" \t "https://zhuanlan.zhihu.com/p/_blank)

rcnn目标检测的基本流程如下：

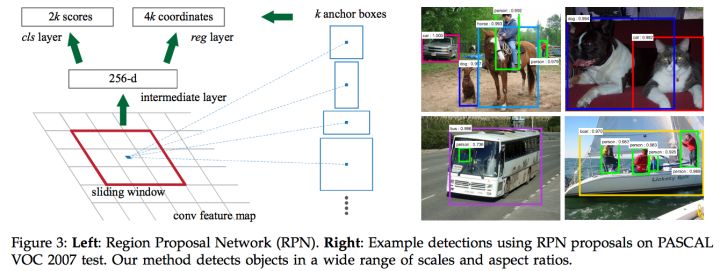
通过selective search的方式从图片中提取可能的目标的候选窗口，将窗口warp到同一个尺寸，通过卷积网络从warp之后的候选窗口提取特征，将特征送给分类器进行分类，最后在加上bounding box regression等操作得到更准确的目标位置。

以上方案有几个问题：1. selective search方式提取候选窗口非常耗时，同时会miss掉一些窗口；2. 从warp之后的候选窗口提取dnn特征的时候，重叠部分的卷积操作重复计算了；3. 提取特征之后再丢给SVM进行分类，逻辑上每一步都是分开训练，不利于全局最优。

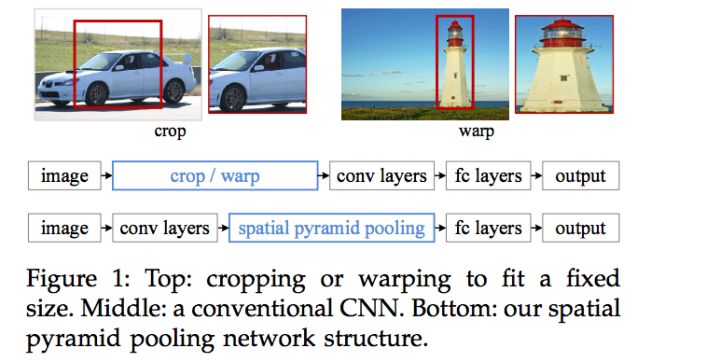
rgbd大神在fast rcnn中提出了ROI pooling层来解决重复的卷积计算的问题，框架如下：

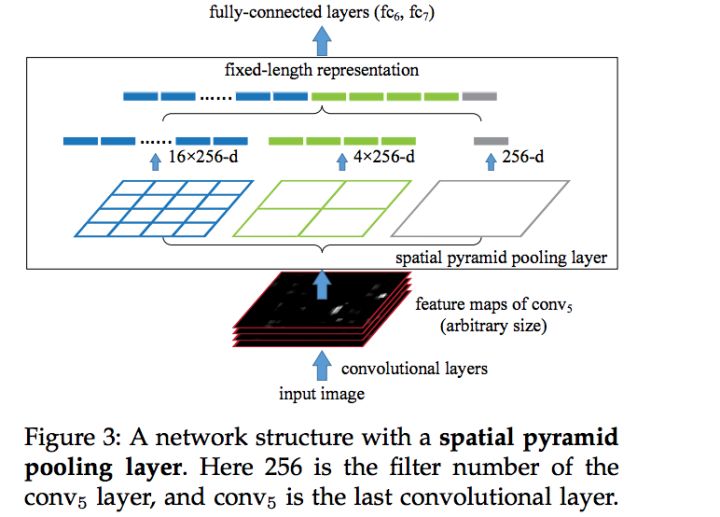
ROI pooling层逻辑上可以理解成，对于一个任意尺寸的box，都可以通过将这个box分成4\*4，8\*8等同样数量的网格，然后每个网格通过max, mean等操作计算相应的特征，就可以将不同尺寸的box转化成相同维度的特征。

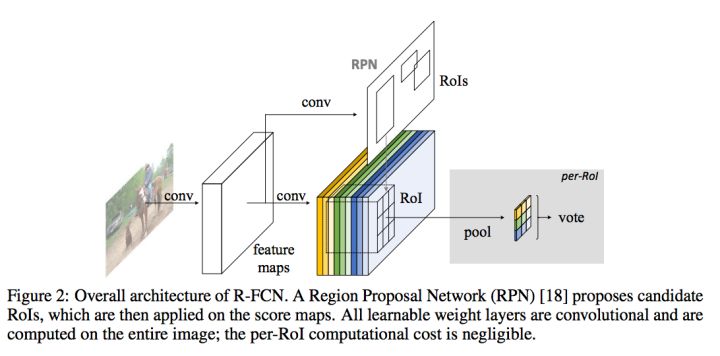
还不够，是否可以简化掉最费时的selective search 提取候选窗口呢？显然可以。rgbd大神在faster rcnn中引入region proposal network替代selective search，同时引入anchor box应对目标形状的变化问题。

在结合end-to-end的training，faster rcnn基本做到的实时。

SPP-net的主要思想是去掉原始图像上的crop/warp操作，通过在卷积特征层上的空间金字塔池化来达成；二者的差异可以参考这个图：

Spatial pyramid pooling的示意图如下，是一个非常重要的将尺寸不一致的图像pooling到相同的特征纬度的方法：

R-FCN方案也使用region proposal network提取候选窗口，同时使用和RPN同享的特征进行分类和回归，流程如下，请读者参考论文详情：



以上方案可以看到明显的进步，并且每一个进步都有重要的背后的逻辑。几个值得关注的点重申如下：

ROI pooling 操作

Anchor box机制

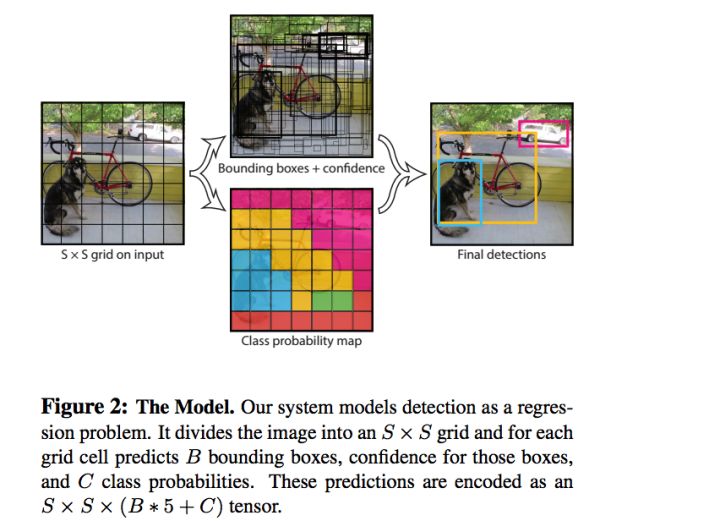
RPN特征共享

卷积层的空间金字塔

等等

**基于深度学习end-to-end的回归方法用于目标检测：**

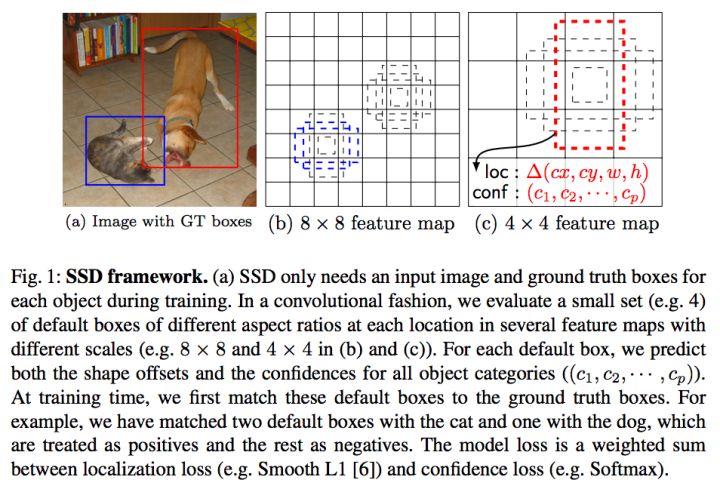
YOLO可以认为是这类方法的开篇之作，速度很快，详细介绍如下。有人说yolo的方案去除了候选窗口或者滑窗的思想，但实际上并没有，只是yolo使用对输出的图像进行网格划分来提取候选窗口而已。

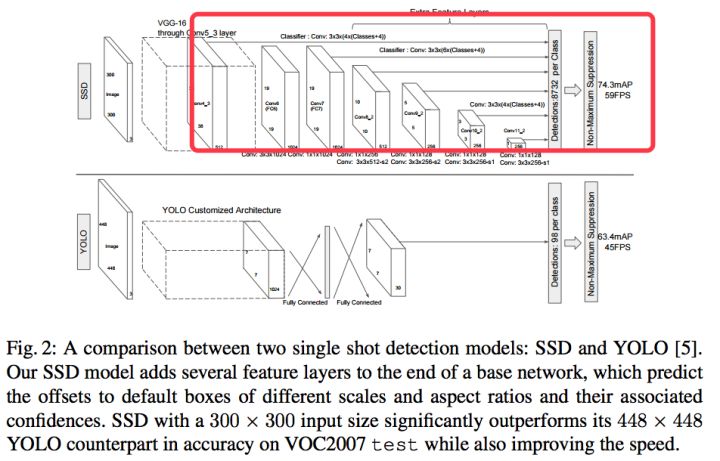
YOLO方案的输入是比如448\*448的图片，输入是一个7\*7的网格，每个网格的信息包含：1. 存在某个类别的物体的概率，比如有20类目标需要检测，那就是20类+1类（backgroud）；2. 对应物体的bounding box参数，使用4个数字描述；3. 对应目标的概率。

这个问题就被format成一个回归问题，448\*448\*3作为输入；7\*7\*（b\*5+c）作为回归的参数，然后进行end-to-end的学习。

显然，YOLO的方案有几个问题：1. 针对小目标的检测效果会不太好，因为7\*7的网格的划分可能太粗糙了；2. 经过多层卷积和pooling的操作，图像的边缘特征可能丢失较多了，而回归bounding box参数的只使用了高层的卷积层特征，会导致bounding box不准确；3. 当两个目标同时落入7\*7的某一个网格中的时候，对应位置只能检测到一个物体。

SSD方案则是faster rcnn和yolo的结合，使用了yolo end-to-end训练的思想，同时有结合了anchor box，不同层的卷积特征的组合等方案。SSD的整体流程如下：

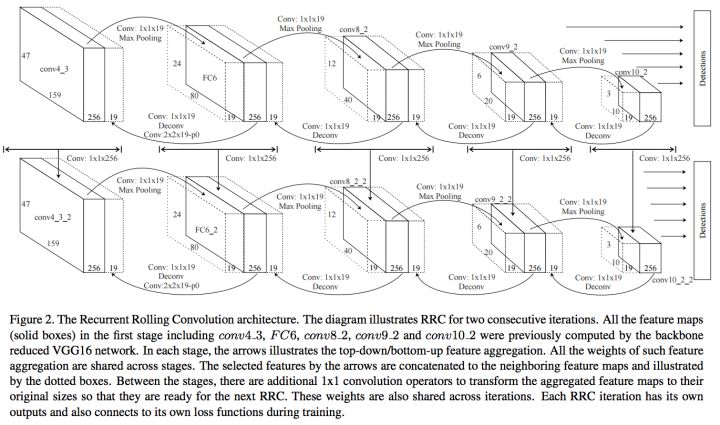
SSD网络和YOLO网络的差异如下：

anchor机制的引入，以及多个卷积层特征的组合能够帮助SSD获得更为准确的对目标bounding box的估计。

KITTI上top 20的目标检测方案及精度大致如下：

对排名第7的RRC 方案做一个描述，之所以介绍这个方法，是因为他在hard上的得分超过了其他方案非常多，并且在moderate easy上的结果也和最好的结果差异不大，而笔者也认为这个方案背后的思想值得学习。

RRC实际上是把RNN的一些思想融入到了目标检测当中：

网络结构比较复杂，做如下几个说明：

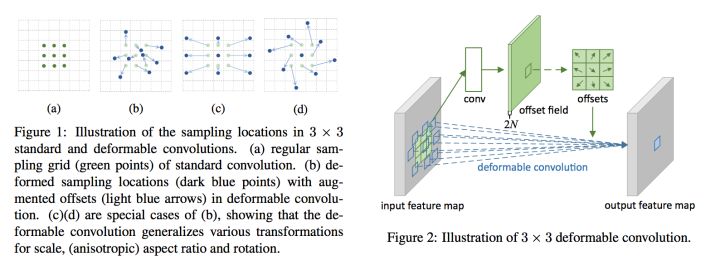
1. 我们在做bounding box回归的时候，一定是需要一些有用的底层的边缘信息的，但并不是所有的底层信息都有用，那这些信息可以通过一定的方式传递的后面的决策层；

2. 利用当前层进行预测的时候，实际上和他的上一层和下一层的信息是有关的。

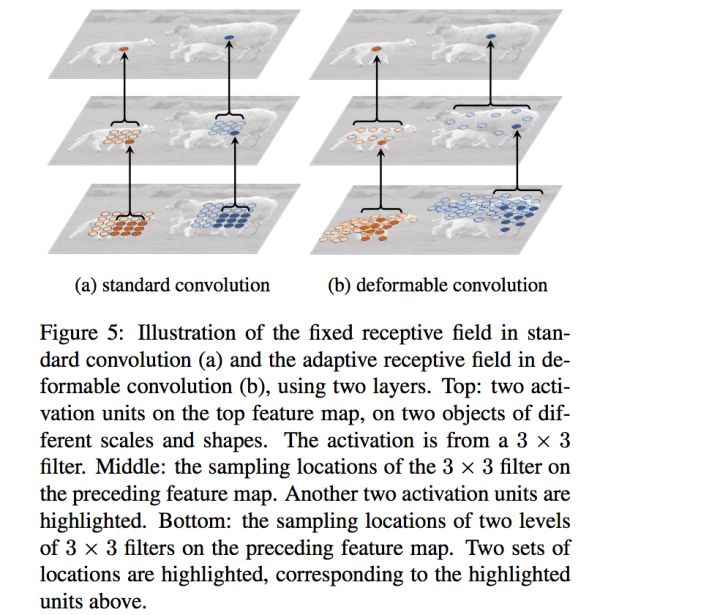
详细请参考RRC detection论文。

Deformable CNN方法：是一种把DPM和CNN结合起来的思想。我们一起思考一下，卷积核一般情况下都是一个固定的尺寸或者是性状，比如3\*3，5\*5；但实际上在我们进行目标检测的时候，我们的目标的性状是千差万别的。比如人的手掌，实际上就是一个长方形或者有一定弧形的区域。那么如果我们可以通过某种方式将卷积核变成或者说是训练成可能的目标的部件的性状，那么他计算出来的特征就越合理。

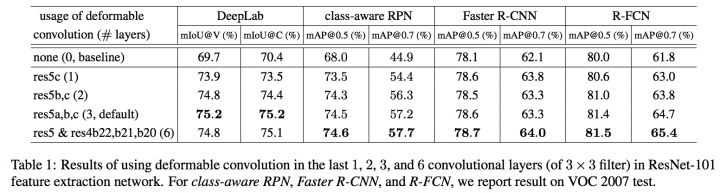
所以我们是否可以在学习卷积核的时候，也引入一些可学习的卷积核的形状参数呢？答案是肯定的，这个就是deformable cnn卷积：

通过对卷积核引入offsets来实现。

通过一个直观的感受来看看学习到的deformable cnn核是什么样子？

可以看到，学习到的deformable cnn核会遵循目标的轮廓进行分布；我们可以想见，或许学习到的对人脸特别敏感的核应该是一个椭圆形的等等。

感受一些deformable cnn的效果：

逻辑上：deformable cnn的效果可以通过多个标准的卷积核组合得到，就相当于椭圆可以通过很多个矩形来近似一样；这种组合似乎有一些inception v3 v4里面的组合的味道；不过我个人认为这种人造的组合可能上限会比通过参数学习的组合低，笔者认为deformable cnn的方案还是非常值得期待的。

**总结、可能：**

基本上对于过去和现在的方法总结得差不多了，似乎该展望未来。不过笔者对于未来的判断一般都没有准确过，所以不敢使用未来这个字眼。但是对于目标检测的一些思考和可能也有一些自己的判断。

基于深度学习方法的几个可能的方向：

1. 想办法从原始图像、低层的feature map层，以及高层的语义层获取更多的信息，从而得到对目标bounding box的更准确的估计；

2. 对bounding box的估计可以结合图片的一些coarse-to-fine的分割信息；

3. bounding box的估计可能需要更多的局部的content的信息，需要想办法引入；

4. 目标检测数据集的标注难度非常大，如何把其他如classfication领域学习到的知识用于检测当中，甚至是将classification的数据和检测数据集做co-training（如YOLO9000）的方式，至少可以从数据层面获得更多的信息；

5. 更好的启发式的学习方式，人在识别物体的时候，第一次可能只是知道这是一个单独的物体，也知道bounding box，但是不知道类别；当人类通过其他渠道学习到类别的时候，下一次就能够识别了；目标检测也是如此，我们不可能标注所有的物体的类别，但是如何将这种快速学习的机制引入，也是一个问题；

6. RRC，deformable cnn中卷积和其他的很好的图片的操作、机器学习的思想的结合未来也有很大的空间；

7. 语意信息和分割的结合，可能能够为目标检测提供更多的有用的信息；

8. 场景信息也会为目标检测提供更多信息；比如天空不会出现汽车等等。

笔者自己一直喜欢非常简单的框架，比如cascade+harr，比如yolo；这种简单的框架对工程上的实现以及后续的优化提供了最多的可能性；也信奉一个道理，在工业界应该是**simple is beautiful**

到这里基本上已经写完了，检测和识别这些基本的task的进步才能够让计算机更好的理解世界，共勉！