①传统的目标检测中，多尺度形变部件模型DPM[1]（Deformable Part Model）是出类拔萃的，连续获得VOC（Visual Object Class）2007到2009的检测冠军，2010年其作者Felzenszwalb Pedro被VOC授予”终身成就奖”。

②Ross Girshick 在2013年开山之作R-CNN《Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation》[2]，是第一个真正可以工业级应用的解决方案奠定了这个子领域的基础，这篇论文后续版本发表在CVPR 2014[3]，期刊版本发表在PAMI 2015[4]

③其实在R-CNN之前已经有很多研究者尝试用Deep Learning的方法来做目标检测了，包括文献[5]和OverFeat[6]

④R-CNN这个领域目前研究非常活跃，先后出现了R-CNN[2,3,4]、SPP-net[7]、Fast R-CNN[8]、Faster R-CNN[9]、R-FCN[10]、YOLO[11]、SSD[12]等研究。Ross Girshick作为这个领域的开山鼻祖总是神一样的存在，R-CNN、Fast R-CNN、Faster R-CNN、YOLO都和他有关。

⑤深度学习相关的目标检测方法也可以大致分为两派：

基于区域提名的：如R-CNN、SPP-net、Fast R-CNN、Faster R-CNN、R-FCN

端到端（End-to-End），无需区域提名的：如YOLO、SSD。

目前来说，基于区域提名的方法依然占据上风，但端到端的方法速度上优势明显，后续的发展拭目以待。

⑥ 选择性搜索

目标检测的第一步是要做区域提名（Region Proposal），也就是找出可能的感兴趣区域（Region Of Interest, ROI）。区域提名类似于光学字符识别（OCR）领域的切分，OCR切分常用过切分方法，简单说就是尽量切碎到小的连通域（比如小的笔画之类），然后再根据相邻块的一些形态学特征进行合并。但目标检测的对象相比OCR领域千差万别，而且图形不规则，大小不一，所以一定程度上可以说区域提名是比OCR切分更难的一个问题。

区域提名可能的方法有：

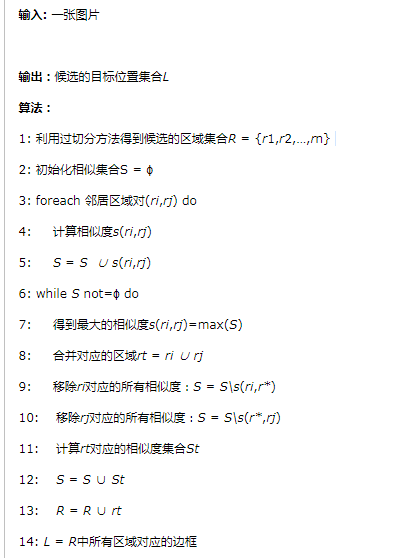
一、滑动窗口。滑动窗口本质上就是穷举法，利用不同的尺度和长宽比把所有可能的大大小小的块都穷举出来，然后送去识别，识别出来概率大的就留下来。很明显，这样的方法复杂度太高，产生了很多的冗余候选区域，在现实当中不可行。

二、规则块。在穷举法的基础上进行了一些剪枝，只选用固定的大小和长宽比。这在一些特定的应用场景是很有效的，比如拍照搜题APP小猿搜题中的汉字检测，因为汉字方方正正，长宽比大多比较一致，因此用规则块做区域提名是一种比较合适的选择。但是对于普通的目标检测来说，规则块依然需要访问很多的位置，复杂度高。

三、选择性搜索。从机器学习的角度来说，前面的方法召回是不错了，但是精度差强人意，所以问题的核心在于如何有效地去除冗余候选区域。其实冗余候选区域大多是发生了重叠，选择性搜索利用这一点，自底向上合并相邻的重叠区域，从而减少冗余。

区域提名并不只有以上所说的三种方法，实际上这块是非常灵活的，因此变种也很多，有兴趣的读者不妨参考一下文献[13]。

选择性搜索的具体算法细节[14]如算法1所示。总体上选择性搜索是自底向上不断合并候选区域的迭代过程。

算法1

[1] Felzenszwalb P F, Girshick R B, McAllester D, et al. Object detection with discriminatively trained part-based models[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2010, 32(9): 1627-1645.

[2] R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. ImageNet Large-Scale Visual Recognition Challenge workshop, ICCV, 2013.

[3] R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[4] R. Girshick, J. Donahue, T. Darrell, J. Malik. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, May. 2015.

[5]Erhan D , Szegedy C , Toshev A , et al. Scalable Object Detection Using Deep Neural Networks[C]// 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, 2014.

[6] P. Sermanet, D. Eigen, X.Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. In ICLR, 2014.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV. 2014.

[8] Girshick, R. Fast R-CNN. ICCV 2015.

[9] S. Ren, K. He, R. Girshick, J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems 28 (NIPS), 2015.

[10] R-FCN: Object Detection via Region-based Fully Convolutional Networks. Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. Conference on Neural Information Processing Systems (NIPS), 2016.

[11] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016)

[12] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[J]. arXiv preprint arXiv:1512.02325, 2015.

[13] J. Hosang, R. Benenson, P. Dolla ́r, and B. Schiele. What makes for effective detection proposals? TPAMI, 2015.

[14] J.R. Uijlings, K.E. vandeSande, T. Gevers, and A.W. Smeulders. Selective search for object recognition. IJCV, 2013.

⑦算法源码

OverFeat source code:http://cilvr.nyu.edu/doku.php?id=software:OverFeat:start

R-CNN: Region-based Convolutional Neural Networks: https://github.com/rbgirshick/rcnn

SPP-net: https://github.com/ShaoqingRen/SPP\_net

Fast R-CNN: https://github.com/rbgirshick/fast-rcnn

Faster R-CNN: https://github.com/rbgirshick/py-faster-rcnn

YOLO: http://pjreddie.com/darknet/yolo/

SSD: https://github.com/weiliu89/caffe/tree/ssd

R-FCN: https://github.com/daijifeng001/r-fcn