cv_family_z的博客

个人资料

cv_family_z

访问： 370026次

积分： 5398

等级： BLOG 6

排名： 第5219名

原创： 175篇 转载： 3篇

译文： 2篇 评论： 197条

文章搜索

Faster R-CNN 之再阅读

2016-07-12 16:54 1779人阅读 评论

分类： ZJ（77） 目标检测（41） 深度学习（90）

Faster R-CNN: Towards Real-Time Object
Detection with Region Proposal Networks

开源代码

https://github.com/ShaoqingRen/faster_rcnn MATLAB

https://github.com/rbgirshick/py-faster-rcnn Python

Faster R-CNN 是在 Fast R-CNN的基础改进的。Fast R-CNN 主要存在的问题就是 region proposal step 是在CPU跑的，比较耗时。CNN通过GPU加速，那么region proposal step能否也可以通过GPU来加速。

Faster R-CNN 通过提出 Region Proposal Networks (RPNs) 实现了快速提取region proposal 。 RPNs 被设计成有效预测候选区域，在一个比较大的尺度和宽高比范围内。

关闭

针对尺度问题，我们采用多个尺度的reference boxes

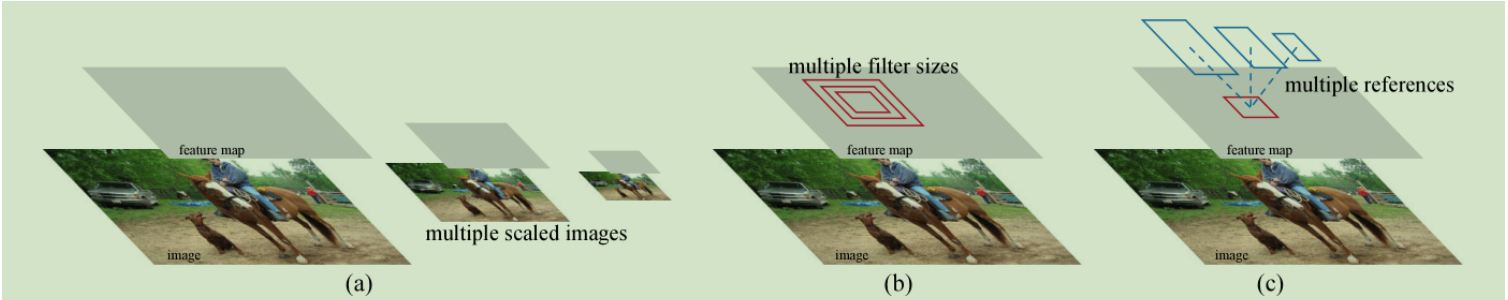

Figure 1: Different schemes for addressing multiple scales and sizes. (a) Pyramids of images and feature maps are built, and the classifier is run at all scales. (b) Pyramids of filters with multiple scales/sizes the feature map. (c) We use pyramids of reference boxes in the regression functions.

3 Faster R-CNN 包括两个部分：1）RPNs，深度全卷积网络用于提取候选区域，2）检测器用于物体检测。

文章分类

目标检测 (42)
模式识别 (3)
深度学习 (91)
目标识别 (7)
车型识别 (6)
行人检测 (13)
人脸识别 (20)
模型测试 (2)
3D-识别 (3)
行人检索 (12)
开源代码 (1)
行人属性 (5)
ZJ (78)
人脸检测 (3)
CVPR 2016 (37)
代码调试 (2)
目标跟踪 (4)
ICML2016 (2)
图像分割 (14)
ECCV2016 (3)
CNN网络压缩 (5)
杂项 (2)
车辆检索 (1)
车辆计数 (2)

文章存档

2017年09月 (6)

关闭

2017年08月 (6)
2017年07月 (3)
2017年06月 (3)
2017年05月 (4)
展开

阅读排行

开源代码文献 (15492)
SSD: Single Shot MultiB (13547)
SSD: Single Shot MultiB (13214)
论文提要“You Only Look (12565)
SPPNet (10875)
Deep Residual Learning (8490)
Inception-v3:"Rethinking (7385)
目标检测--PVANET: Dee (6629)
论文提要“Fast Feature P (6153)
Faster R-CNN (5678)

评论排行

论文提要“You Only Look (25)
人脸检测“A Fast and Acc (20)
CompCars模型测试 (19)
车辆检测“DAVE: A Uniec (9)
论文提要“Fast Feature P (9)
车型识别“A Large-Scale (9)
论文提要“Hypercolumns (7)

the RPNs module tells the Fast R-CNN module where to look

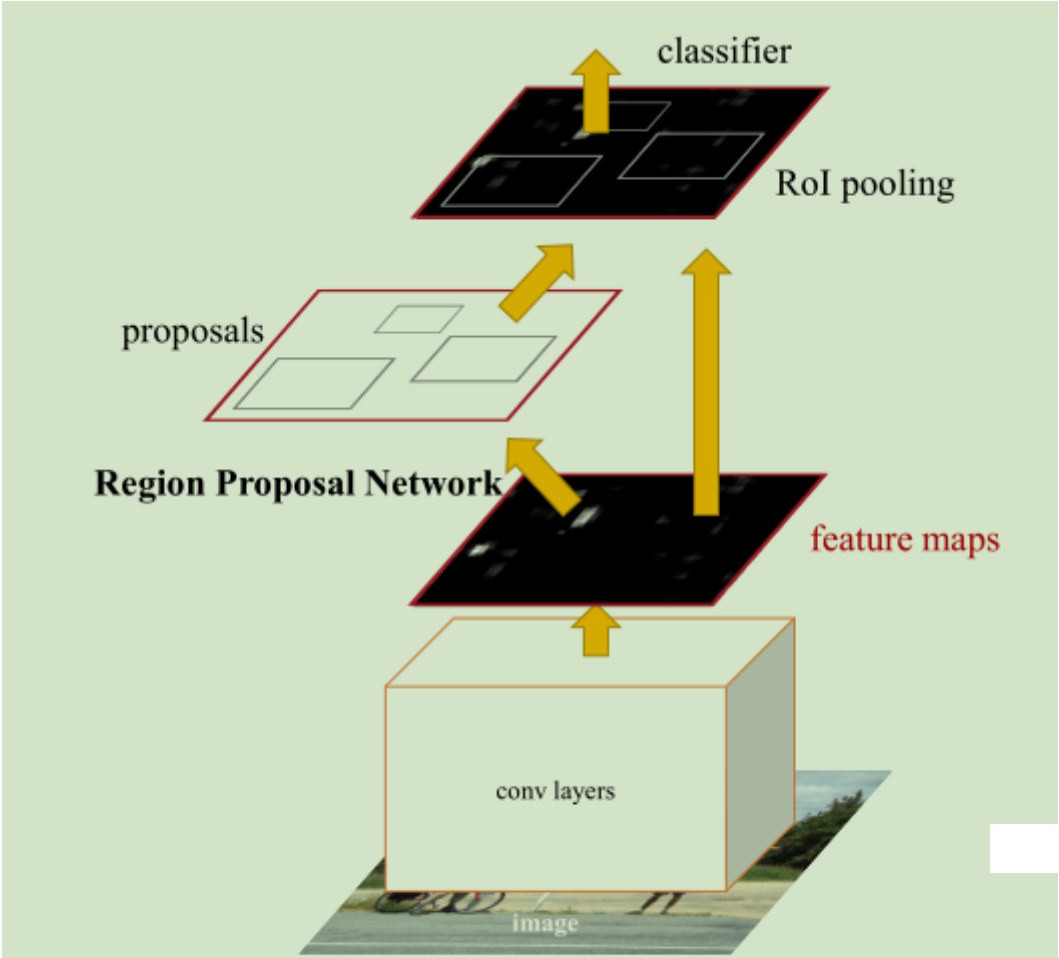

Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

3.1 Region Proposal Networks

RPN 将一幅任意尺寸图像作为输入，输出若干矩形框，每个矩形框有一个 objectness score。为了与Fast R-CNN检测网络共享计算，我们对RPN使用卷积网络， we assume that both nets share a

关闭

人脸识别 - A Discriminat (6)
Beyond Local Search: Tr (5)
A Lightened CNN for De (4)

common set of convolutional layers。在我们的实验中， Zeiler and Fergus model（ZF）有5个 shareable convolutional layers，VGG-16 有13个 shareable convolutional layers。

为了生成候选区域，我们在卷积特征层（最后一个共享卷积层的输出）上滑动一个小的网络。这个小的网络将卷积特征层的 n*n领域窗口作为输入，将其映射到一个低维的特征（256-d for ZF and 512-d for VGG, with ReLU [33]following），然后将这个特征输入给两个 sibling 全链接层：a box-regression layer (reg) and a box-classification layer (cls)。这里我们使用 n=3，对应输入图像的尺寸是 171 and 228 pixels for ZF and VGG, respectively。因为对于所有的位置，全链接层是共享的，所以架构可以如下实现：n*n卷积层，接着是两个 sibling 1*1卷积层。

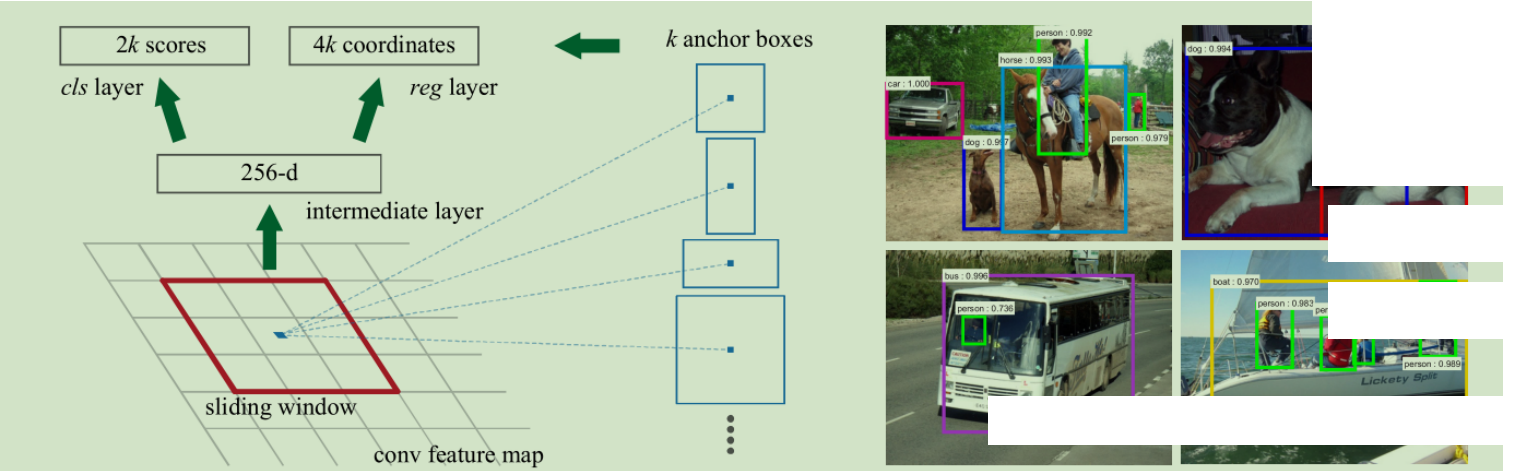

Figure 3: **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios.

3.1.1 Anchors

对于每个小网络的滑动窗口位置，我们同时预测k个矩形区域，对应不同尺度和宽高比。本文采用3个尺度，3个宽高比，所以每个位置我们检测3*3=9个矩形区域，每个区域计算其是否含有 object（ a two-class softmax layer，这是二分类，也可以进行K分类）

关闭

Translation-Invariant Anchors

我们这个方法一个很重要的特征就是平移不变性，就是对于同一个object，在图像不同位置出现，同样的 anchors and the functions 可以提出该 object。MultiBox 方法则没有这个特征。

平移不变性也降低了模型的大小。

Multi-Scale Anchors as Regression References

多个不同尺度的Anchors 可以很好（速度快）的解决 object 尺度问题。以前解决尺度问题的计算量比较大。

3.1.2 Loss Function

在训练 RPNs时，我们对每个 anchor 赋予一个二类标记（是不是 object）。对于以下两类 anchor,我们

1）最大IoU 的anchor，2）IoU 超过0.7的 anchor。对所有 ground-truth boxes 的IoU 都小于0.3的anchor

既不是正值，也不是负值的anchor 对训练函数没有贡献。

我们的损失函数定义如下：

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*).$$

3.1.3 Training RPNs

RPN的训练采用 back-propagation and stochastic gradient descent (SGD), 采用文献【2】中的 “image-centric”的采样策略。

3.2 Sharing Features for RPN and Fast R-CNN

怎么共享RPN and Fast R-CNN 的卷积层，文中讨论了三种方式：

关闭

1）Alternating training 交替训练，先训练RPN，然后用RPN得到的候选区域训练Fast R-CNN，得到的Fast R-CNN用于初始化RPN，这样交替训练，本文所有的实验都是采用这种方式。

2） Approximate joint training

这里我们将两个网络融合成一个网络训练，如图2所示。在每个SGD迭代步骤中，前向计算产生候选区域，然后假定这些候选区域固定，用于训练Fast R-CNN。后向传播计算像以前一样，对于共享网络，两个网络的损失被结合起来候选误差传播 。这种方式实现比较简单，但是它忽略了 the derivative w.r.t. the proposal boxes' coordinates that are also network responses，所以它是近似的，与交替训练相比，该方式的训练时间减少 25-50%。在Python 代码中我们给出了这种方法的实现。

(iii) Non-approximate joint training

这种方法主要是将上个方法忽略的矩形框的梯度问题考虑进来。所以是非近似的。本文没有实现改方

4-Step Alternating Training

1）：使用 ImageNet-pre-trained 模型初始化网络，然后训练RPN

2）：使用1）产生的候选区域，训练一个单独的 Fast R-CNN检测网络，该网络也是使用 ImageNet-pre-trained模型初始化的。

3）：使用检测网络初始化RPN训练，固定共享卷积层，只微调与RPN特有的网络层，这样两个网络就共享了卷积层

4）：固定共享卷积层，微调Fast R-CNN特有网络层

可以做更多的交替训练，但是没有提升效果。

关闭

实验结果：

Table 4: Detection results on **PASCAL VOC 2012 test set**. The detector is Fast R-CNN and VGG-16. Training data: "07": VOC 2007 trainval, "07++12": union set of VOC 2007 trainval+test and VOC 2012 trainval. For RPN, the train-time proposals for Fast R-CNN are 2000. †: http://host.robots.ox.ac.uk:8080/anonymous/HZJTQA.html. ‡: http://host.robots.ox.ac.uk:8080/anonymous/YNPLXB.html. §: http://host.robots.ox.ac.uk:8080/anonymous/XEDH10.html.

method	# proposals	data	mAP (%)
SS	2000	12	65.7
SS	2000	07++12	68.4
RPN+VGG, shared†	300	12	67.0
RPN+VGG, shared‡	300	07++12	**70.4**
RPN+VGG, shared§	300	COCO+07++12	**75.9**

Table 5: **Timing** (ms) on a K40 GPU, except SS proposal is evaluated in a CPU. "Region-wise" inc pooling, fully-connected, and softmax layers. See our released code for the profiling of running t

model	system	conv	proposal	region-wise	total	rate
VGG	SS + Fast R-CNN	146	1510	174	1830	0.5 fps
VGG	RPN + Fast R-CNN	141	**10**	47	**198**	**5 fps**
ZF	RPN + Fast R-CNN	31	**3**	25	**59**	**17 fps**

Table 6: Results on PASCAL VOC 2007 test set with Fast R-CNN detectors and VGG-16. For RPN, th proposals for Fast R-CNN are 2000. RPN* denotes the unsharing feature version.

method	# box	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep
SS	2000	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4
SS	2000	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1
RPN*	300	07	68.5	74.1	77.2	67.7	53.9	51.0	75.1	79.2	78.9	50.7	78.0	61.1						
RPN	300	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3
RPN	300	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6
RPN	300	COCO+07+12	**78.8**	**84.3**	**82.0**	**77.7**	**68.9**	**65.7**	**88.1**	**88.4**	**88.9**	**63.6**	**86.3**	**70.8**	**85.9**	**87.6**	**80.1**	**82.3**	**53.6**	**80.4**

method	sofa	train	tv
SS	67.3	81.1	67.6
SS	73.9	83.0	72.6
RPN*			
RPN	73.6	73.9	83.0	72.6
RPN	**75.8**	**86.6**	**78.9**

Table 7: Results on PASCAL VOC 2012 test set with Fast R-CNN detectors and VGG-16. For RPN, the train-time proposals for Fast R-CNN are 2000.

method	# box	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SS	2000	12	65.7	80.3	74.7	66.9	46.9	37.7	73.9	68.6	87.7	41.7	71.1	51.1	86.0	77.8	79.8	69.8	32.1	65.5	63.8	76.4	61.7
SS	2000	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	**65.7**	80.4	64.2
RPN	300	12	67.0	82.3	76.4	71.0	48.4	45.2	72.1	72.3	87.3	42.2	73.7	50.0	86.8	78.7	78.4	77.4	34.5	70.1	57.1	77.1	58.9
RPN	300	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
RPN	300	COCO+07++12	**75.9**	**87.4**	**83.6**	**76.8**	**62.9**	**59.6**	**81.9**	**82.0**	**91.3**	**54.9**	**82.6**	**59.0**	**89.0**	**85.5**	**84.7**	**84.1**	**52.2**	**78.9**	65.5	**85.4**	**70.2**

关闭

上一篇 Practical Recommendations for Gradient-Based Training of Deep Architectures
下一篇 SSD: Single Shot MultiBox Detector 之再阅读

查看评论

2楼 cv_family_z 2016-07-14 08:47发表

在第三步 固定共享卷积层，这样两个网络使用了同样的卷积层，这里是人为强制共享的，所谓共享就是两个网络使用了同样的卷积层，这么做是为了提高速度，减少运算量

1楼 ShuLanguhS 2016-07-13 16:00发表

在4-Step Alternating Training 中作者提到This detection net-work is also initialized by the ImageNet-pre-trained model. At this point the two networks do not share convolutional layers. In the third step, we use the detector network to initialize RPN training, but we fix the shared convolutional layers and only fine-tunethe layers unique to RPN.在第二步中两个网络没有共享的卷积层，在第三步中固定的共享卷积层是哪里来的呢？本人新手刚接触该领域

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

关闭