

原 (RegionProposal Network)RPN网络结构及详解

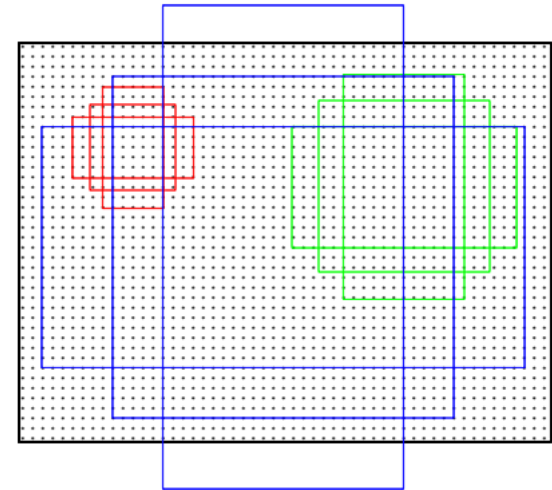
2018年05月23日 17:40:38 凹酱deep 阅读数：1990 更多

RPN(RegionProposal Network)区域生成网络

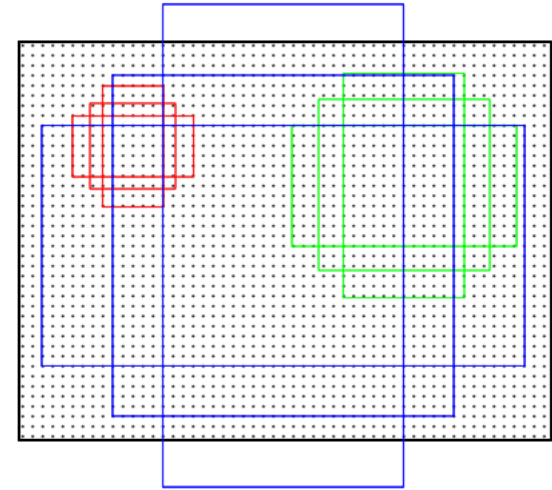
Faster-RCNN的核心。在这里整理。

1.anchors。

特征可以看做一个尺度51*39的256通道图像，对于该图像的每一个位置，考虑9个可能的候选窗口：三种面积三种比例，候选窗口称为anchors。51*39个anchor中心，以及9种anchor示例。



三种比例。这些候选窗口称为anchors。下图示出51*39个anchor中心，以及9种anchor示例。



2.SOFTMAX的两支



计算每个像素256-d的9个尺度下的值，得到9个anchor，我们给每个anchor分配一个二进制的标签（前景背景）。我们分配正标签前景给两类anchor：ground truth（GT）包围盒有最高的IoU重叠的anchor（也许不到0.7），2）与任意GT包围盒有大于0.7的IoU交叠的anchor。注意到一个GT包围盒可能给多个anchor。我们分配负标签（背景）给与所有GT包围盒的IoU比率都低于0.3的anchor。非正非负的anchor对训练目标没有任何作用，由此输出维18-d，anchor（label和概率）一共18维。

假设在conv5 feature map中每个点上有k个anchor（默认k=9），而每个anhcor要分foreground和background，所以每个点由256d feature转化为cls=21 每个anchor都有x y w h对应4个偏移量 所以req=4k coordinates

-----分割线-----

以上是传统的RPN，下面是Faster R-CNN 的RPN部分。

3.bounding box regression

前2.) 中已经计算出foreground anchors，使用bounding box regression回归得到预设anchor-box到ground-truth-box之间的变换参数，即平移 (d_x 和 d_y) (d_w 和 d_h)，由此得到初步确定proposal。

如图9所示绿色框为飞机的Ground Truth(GT)，红色为提取的foreground anchors，那么即便红色的框被分类器识别为飞机，但是由于红色的框定位不准，这张图相当于没有机。所以我们希望采用一种方法对红色的框进行微调，使得foreground anchors和GT更加接近。

图9

缩进对于窗口一般使用四维向量 (x, y, w, h) 表示，分别表示窗口的中心点坐标和宽高。对于图 10，红色的框A代表原始的Foreground Anchors，绿色的GT，我们的目标是寻找一种关系，使得输入原始的anchor A经过映射得到一个跟真实窗口G更接近的回归窗口G'，即：给定 $A=(A_x, A_y, A_w, A_h)$ ，寻找使得 $f(A_x, A_y, A_w, A_h)=(G'_x, G'_y, G'_w, G'_h)$ ，其中 $(G'_x, G'_y, G'_w, G'_h) \approx (G_x, G_y, G_w, G_h)$ 。

图10

那么经过何种变换才能从图6中的A变为G'呢？比较简单的思路就是：

缩进 1. 先做平移

缩进 2. 再做缩放

缩进观察上面4个公式发现，需要学习的是 $d_x(A)$ ， $d_y(A)$ ， $d_w(A)$ ， $d_h(A)$ 这四个变换。当输入的anchor与GT相差较小时，可以认为这种变换是一种线性变换，可以用线性回归来建模对窗口进行微调（注意，只有当anchors和GT比较接近时，才能使用线性回归模型，否则就是复杂的非线性问题了）。对应于FRCNN，平移量 (t_x, t_y) 与尺度因子 (t_w, t_h) 如下：

缩进接下来的问题就是如何通过线性回归获得 $d_x(A)$ ， $d_y(A)$ ， $d_w(A)$ ， $d_h(A)$ 了。线性回归就是给定输入的特征向量X，学习一组参数W，使得经过线性回归值Y（即GT）非常接近，即 $Y=WX$ 。对于该问题，输入X是一张经过卷积获得的feature map，定义为 Φ ；同时还有训练传入的GT，即 (t_x, t_y, t_w, t_h) 。 $d_x(A)$ ， $d_y(A)$ ， $d_w(A)$ ， $d_h(A)$ 四个变换。那么目标函数可以表示为：

其中 $\Phi(A)$ 是对应anchor的feature map组成的特征向量，w是需要学习的参数，d(A)是得到的预测值（*表示x，y，w，h，也就是每一个变换对应一维）。为了让预测值 (t_x, t_y, t_w, t_h) 与真实值最小，得到损失函数：

函数优化目标为：

Boundingbox regression原理http://blog.csdn.net/elaine_bao/article/details/60469036

4.将预proposal利用feat_stride和im_info将anchors映射回原图，判断预proposal是否大范围超过边界，剔除严重超出边界的。

按照softmax score进行从大到小排序，提取前2000个预proposal，对这个2000个进行NMS(非极大值抑制)，将得到的再次进行排序，输出300个proposal。

NMS（非极大值抑制）

由于锚点经常重叠，因此建议最终也会在同一个目标上重叠。为了解决重复建议的问题，我们使用一个简单的算法，称为非极大抑制（NMS）

- 将所有候选框的得分进行排序，选中最高分及其所对应的BB；

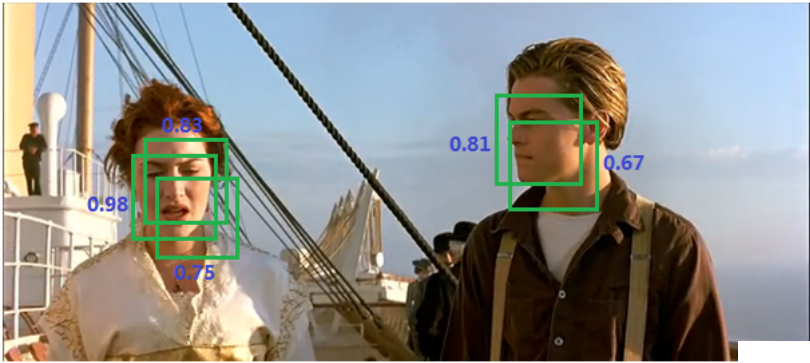


图10

- 遍历其余的框，如果它和当前最高得分框的重叠面积大于一定的阈值，我们将其删除。

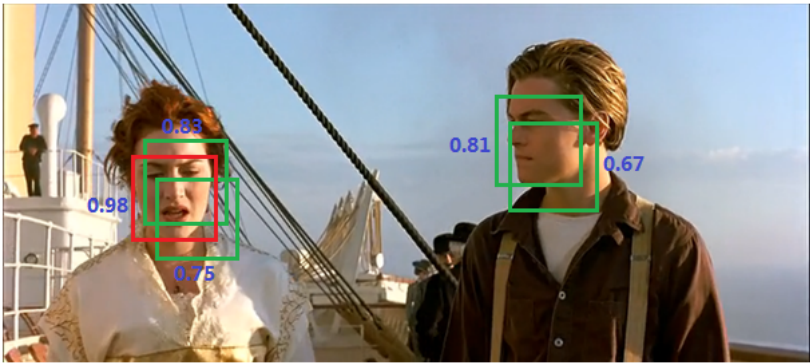
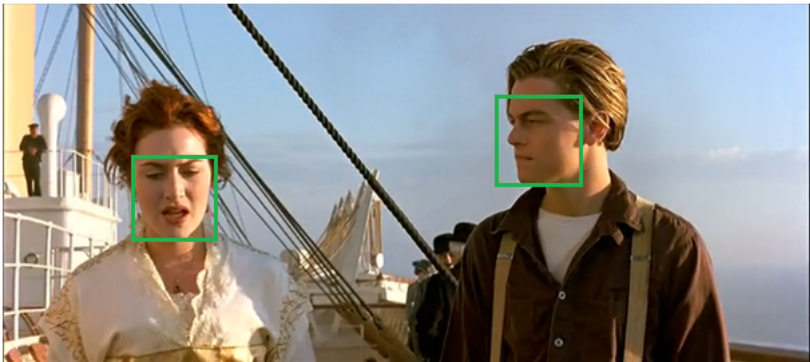


图11

- 从没有处理的框中继续选择一个得分最高的，重复上述过程。



框架流程：from:<https://blog.csdn.net/zjucor/article/details/78232024>

我的这个可以看到具体数据流动情况。

```
1 layer{
2     name:"data"
3     top:"data"
4     type:"Input"
5     input_param{
6         shape{ dim: 1 dim: 3 dim: 224 dim: 224}
7     }
8 }
9 #===== conv1-conv5 =====
10
11 layer {
12     name: "conv1"
13     type: "Convolution"
```

```

14 |         bottom: "data" 15 |         top: "conv1"
16 |         param { lr_mult: 1.0 }
17 |         param { lr_mult: 2.0 }
18 |         convolution_param {
19 |             num_output: 96
20 |             kernel_size: 7
21 |             pad: 3
22 |             stride: 2
23 |         }
24 |     }
25 |     layer {
26 |         name: "relu1"
27 |         type: "ReLU"
28 |         bottom: "conv1"
29 |         top: "conv1"
30 |     }
31 |     layer {
32 |         name: "norm1"
33 |         type: "LRN"
34 |         bottom: "conv1"
35 |         top: "norm1"
36 |         lrn_param {
37 |             local_size: 3
38 |             alpha: 0.00005
39 |             beta: 0.75
40 |             norm_region: WITHIN_CHANNEL
41 |         engine: CAFFE
42 |     }
43 | }
44 | layer {
45 |     name: "pool1"
46 |     type: "Pooling"
47 |     bottom: "norm1"
48 |     top: "pool1"
49 |     pooling_param {
50 |         kernel_size: 3
51 |         stride: 2
52 |         pad: 1
53 |         pool: MAX
54 |     }
55 | }
56 | layer {
57 |     name: "conv2"
58 |     type: "Convolution"
59 |     bottom: "pool1"
60 |     top: "conv2"
61 |     param { lr_mult: 1.0 }
62 |     param { lr_mult: 2.0 }
63 |     convolution_param {
64 |         num_output: 256
65 |         kernel_size: 5
66 |         pad: 2
67 |         stride: 2
68 |     }
69 | }
70 | layer {
71 |     name: "relu2"
72 |     type: "ReLU"
73 |     bottom: "conv2"
74 |     top: "conv2"
75 | }
76 | layer {
77 |     name: "norm2"
78 |     type: "LRN"
79 |     bottom: "conv2"
80 |     top: "norm2"
81 |     lrn_param {
82 |         local_size: 3
83 |         alpha: 0.00005
84 |         beta: 0.75

```

```

85 |         norm_region: WITHIN_CHANNEL 86 |         engine: CAFFE
87 |     }
88 | }
89 | layer {
90 |     name: "pool2"
91 |     type: "Pooling"
92 |     bottom: "norm2"
93 |     top: "pool2"
94 |     pooling_param {
95 |         kernel_size: 3
96 |         stride: 2
97 |         pad: 1
98 |         pool: MAX
99 |     }
100 | }
101 | layer {
102 |     name: "conv3"
103 |     type: "Convolution"
104 |     bottom: "pool2"
105 |     top: "conv3"
106 |     param { lr_mult: 1.0 }
107 |     param { lr_mult: 2.0 }
108 |     convolution_param {
109 |         num_output: 384
110 |         kernel_size: 3
111 |         pad: 1
112 |         stride: 1
113 |     }
114 | }
115 | layer {
116 |     name: "relu3"
117 |     type: "ReLU"
118 |     bottom: "conv3"
119 |     top: "conv3"
120 | }
121 | layer {
122 |     name: "conv4"
123 |     type: "Convolution"
124 |     bottom: "conv3"
125 |     top: "conv4"
126 |     param { lr_mult: 1.0 }
127 |     param { lr_mult: 2.0 }
128 |     convolution_param {
129 |         num_output: 384
130 |         kernel_size: 3
131 |         pad: 1
132 |         stride: 1
133 |     }
134 | }
135 | layer {
136 |     name: "relu4"
137 |     type: "ReLU"
138 |     bottom: "conv4"
139 |     top: "conv4"
140 | }
141 | layer {
142 |     name: "conv5"
143 |     type: "Convolution"
144 |     bottom: "conv4"
145 |     top: "conv5"
146 |     param { lr_mult: 1.0 }
147 |     param { lr_mult: 2.0 }
148 |     convolution_param {
149 |         num_output: 256
150 |         kernel_size: 3
151 |         pad: 1
152 |         stride: 1
153 |     }
154 | }
155 | layer {

```

```

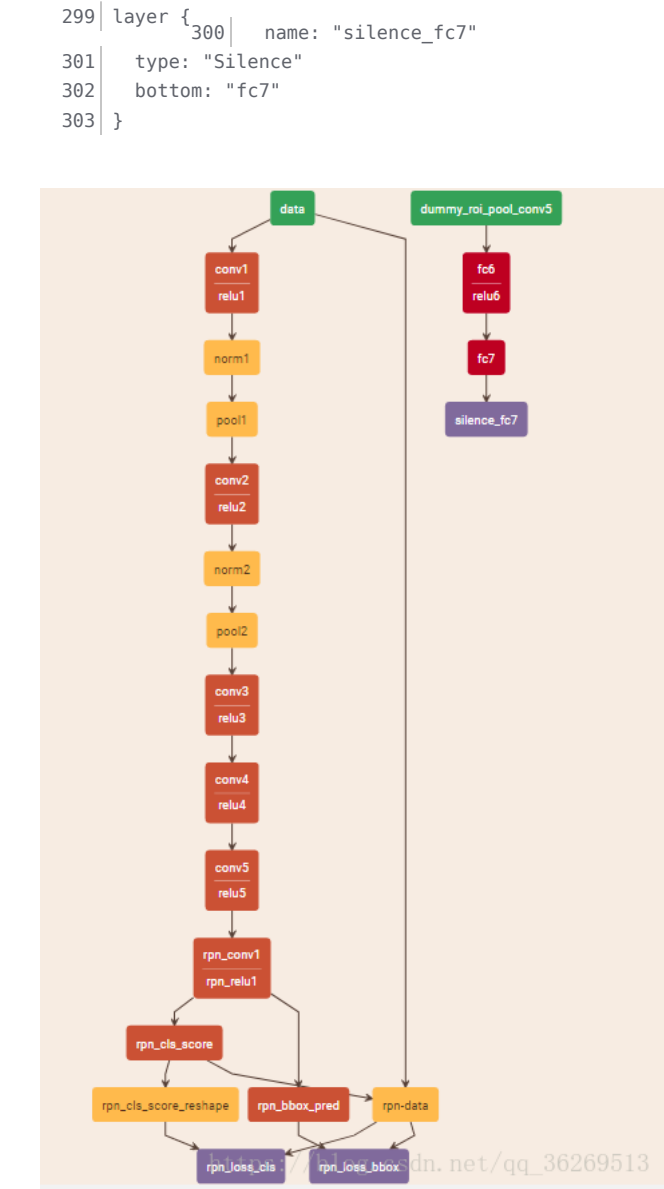
156     name: "relu5"157 |         type: "ReLU"
158     bottom: "conv5"
159     top: "conv5"
160 }
161
162
163 layer {
164     name: "rpn_conv1"
165     type: "Convolution"
166     bottom: "conv5"
167     top: "rpn_conv1"
168     param { lr_mult: 1.0 }
169     param { lr_mult: 2.0 }
170     convolution_param {
171         num_output: 256
172         kernel_size: 3 pad: 1 stride: 1
173         weight_filler { type: "gaussian" std: 0.01 }
174         bias_filler { type: "constant" value: 0 }
175     }
176 }
177 layer {
178     name: "rpn_relu1"
179     type: "ReLU"
180     bottom: "rpn_conv1"
181     top: "rpn_conv1"
182 }
183 layer {
184     name: "rpn_cls_score"
185     type: "Convolution"
186     bottom: "rpn_conv1"
187     top: "rpn_cls_score"
188     param { lr_mult: 1.0 }
189     param { lr_mult: 2.0 }
190     convolution_param {
191         num_output: 18
192         kernel_size: 1 pad: 0 stride: 1
193         weight_filler { type: "gaussian" std: 0.01 }
194         bias_filler { type: "constant" value: 0 }
195     }
196 }
197 layer {
198     name: "rpn_bbox_pred"
199     type: "Convolution"
200     bottom: "rpn_conv1"
201     top: "rpn_bbox_pred"
202     param { lr_mult: 1.0 }
203     param { lr_mult: 2.0 }
204     convolution_param {
205         num_output: 36
206         kernel_size: 1 pad: 0 stride: 1
207         weight_filler { type: "gaussian" std: 0.01 }
208         bias_filler { type: "constant" value: 0 }
209     }
210 }
211 layer {
212     bottom: "rpn_cls_score"
213     top: "rpn_cls_score_reshape"
214     name: "rpn_cls_score_reshape"
215     type: "Reshape"
216     reshape_param { shape { dim: 0 dim: 2 dim: -1 dim: 0 } }
217 }
218 layer {
219     name: 'rpn-data'
220     type: 'Python'
221     bottom: 'rpn_cls_score'
222     bottom: 'gt_boxes'
223     bottom: 'im_info'
224     bottom: 'data'
225     top: 'rpn_labels'
226     top: 'rpn_bbox_targets'
227     top: 'rpn_bbox_inside_weights'

```

```

228 | top: 'rpn_bbox_outside_weights'
      |                                     229 | python_param {
230 |     module: 'rpn.anchor_target_layer'
231 |     layer: 'AnchorTargetLayer'
232 |     param_str: "'feat_stride': 16"
233 | }
234 | }
235 | layer {
236 |     name: "rpn_loss_cls"
237 |     type: "SoftmaxWithLoss"
238 |     bottom: "rpn_cls_score_reshape"
239 |     bottom: "rpn_labels"
240 |     propagate_down: 1
241 |     propagate_down: 0
242 |     top: "rpn_cls_loss"
243 |     loss_weight: 1
244 |     loss_param {
245 |         ignore_label: -1
246 |         normalize: true
247 |     }
248 | }
249 | layer {
250 |     name: "rpn_loss_bbox"
251 |     type: "SmoothL1Loss"
252 |     bottom: "rpn_bbox_pred"
253 |     bottom: "rpn_bbox_targets"
254 |     bottom: "rpn_bbox_inside_weights"
255 |     bottom: "rpn_bbox_outside_weights"
256 |     top: "rpn_loss_bbox"
257 |     loss_weight: 1
258 |     smooth_l1_loss_param { sigma: 3.0 }
259 | }
260 |
261 |
262 | layer {
263 |     name: "dummy_roi_pool_conv5"
264 |     type: "DummyData"
265 |     top: "dummy_roi_pool_conv5"
266 |     dummy_data_param {
267 |         shape { dim: 1 dim: 9216 }
268 |         data_filler { type: "gaussian" std: 0.01 }
269 |     }
270 | }
271 | layer {
272 |     name: "fc6"
273 |     type: "InnerProduct"
274 |     bottom: "dummy_roi_pool_conv5"
275 |     top: "fc6"
276 |     param { lr_mult: 0 decay_mult: 0 }
277 |     param { lr_mult: 0 decay_mult: 0 }
278 |     inner_product_param {
279 |         num_output: 4096
280 |     }
281 | }
282 | layer {
283 |     name: "relu6"
284 |     type: "ReLU"
285 |     bottom: "fc6"
286 |     top: "fc6"
287 | }
288 | layer {
289 |     name: "fc7"
290 |     type: "InnerProduct"
291 |     bottom: "fc6"
292 |     top: "fc7"
293 |     param { lr_mult: 0 decay_mult: 0 }
294 |     param { lr_mult: 0 decay_mult: 0 }
295 |     inner_product_param {
296 |         num_output: 4096
297 |     }
298 | }

```



参考文献：

<https://blog.csdn.net/ying86615791/article/details/72788414>

http://www.360doc.com/content/17/0809/10/10408243_677742029.shtml

<https://blog.csdn.net/zjucor/article/details/78232024>

<https://blog.csdn.net/WZZ18191171661/article/details/79439212>

http://www.360doc.com/content/18/0523/16/55931805_756422650.shtml

劲爆！光谷25岁美女用微信做这个，1个月存款吓呆父母！！

1 · 爆炒

想对作者说点什么？ 我来说一句

moyemoji：博主，256-d的-d是什么意思啊？还请指教。（1周前 #1楼）

[caffe笔记005]：通过代码理解faster-RCNN中的RPN

注意：整个RPN完全是笔者自己的理解，可能会有一些理解错误的地方。1. RPN简介RPN是regional proposal netw...

Region Proposal Network

RPN

2.5万

来自：Kevincc的专栏

8173

来自：Daniel2333的博客

从0到1学好区块链开发，一年编程经验学完月薪可达40K+

立即申请试学，成为时代颠覆者

Faster-Rcnn中RPN (Region Proposal Network) 的理解

👁 1.4万

卷积后的pool层特征既可以用于类别判别，也可以用于回归BoundingBox，可以这样想，object的外围存在着一个看...

来自： [MLLearnerTJ的博客](#)

RPN网络通俗理解

👁 642

【提特征】RPN网络前面是一个提特征的网络，比如VGG,Res等，传给RPN网络的是一个特征图，其实也就是一个...

来自： [Emiedon的博客](#)

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks(更快的RCNN...

👁 1.2万

本文翻译了Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks，希望学习深度学...

来自： [u011326478的专栏](#)

基于Region Proposal的目标跟踪算法-A New Target-specific Object Proposal Generation Method f

👁 555

最近的CVPR2018上出现了一篇使用Region Proposal Network来做目标跟踪的文章，题目是"High Performance ...

来自： [HUSTbest_的博客](#)

劲爆！光谷25岁美女用微信做这个，1个月存款吓呆父母！！

1 · 熿熿

Faster-Rcnn中RPN(Region Proposal Network)的理解 - CSDN博客

Faster-Rcnn中RPN(Region Proposal Network)的理解 欧西特 阅读量:14096 2016-...在实际使用中,pool层后增加了一个调整层(convolution,见train.prototxt的rpn_conv...

Region Proposal Network - Daniel2333的博客 - CSDN博客

rpn_conv/3x3的bottom layer是conv5_3, 因为stride = 1, pad =1, num_output...RPN(RegionProposal Network)区域生成网络 时Faster-RCNN的核心。在此整理。...

目标检测中region proposal的作用

👁 1908

作者：YJHMITWEB链接：https://www.zhihu.com/question/265345106/answer/294410307来源：知乎著作权归作...

来自： [baidu_26454813的博客](#)

【转】Faster RCNN原理分析（二）：Region Proposal Networks详解

👁 996

转自：http://lib.csdn.net/article/deeplearning/61641 Region Proposal Networks是Faster RCNN出新提出来的propo...

来自： [lanyuxuan100的博客](#)

Region proposal学习笔记

👁 371

Region Proposal (候选区域)，就是预先找出图中目标可能出现的位置，通过利用图像中的纹理、边缘、颜色...

来自： [infinita_LV的博客](#)

Region Proposal Networks

👁 8960

Region Proposal Networks RPNs 从任意尺寸的图片中得到一系列的带有 objectness score 的 object proposals。具...

来自： [CNV_2305](#)

博主推荐

 AI之路 关注 145篇文章	 hjimce 关注 197篇文章	 大饼博士X 关注 93篇文章
--	--	--

区域推荐网络RPN - CSDN博客

区域推荐网络RPN

...Detection with Region Proposal Networks(更快的RC..._CSDN博客

本文中,我们介绍一种区域建议网络(Region Proposal Network, RPN),它和检测网络共享全图的卷积特征,使得区域建议几乎不花时间。RPN是一个全卷积网络,在每个位置同

Faster RCNN原理分析 :Region Proposal Networks详解 -..._CSDN博客

缩进Region Proposal Networks是Faster RCNN出新提出来的proposal生成网络。其替代了之前RCNN和Fast RCNN中的selective search方法,将所有内容整合在一个网络中

目标检测中region proposal的作用 - baidu_26454813的..._CSDN博客

而2 stage多出来的这个stage就是Regional Proposal...RPN(RegionProposal Network)区域生成网络 时Faster...

目标检测方法简介:RPN(Region Proposal Network) 和 SS..._CSDN博客

最近几年深度学习在计算机视觉领域取得了巨大的成功,而在目标检测这一计算机视觉的经典问题上直到去年(2015)才有了完全使用深度学习的方法,也就是大名鼎鼎的Faster

Region proposal networks by faster rcnn - wejoncy的专栏 - ...

缩进Region Proposal Networks是Faster RCNN出新提出来的proposal生成网络。其替代...RPN(RegionProposal Network)区域生成网络 时Faster-RCNN的核心。在此整理。



凹酱deep

关注

原创15

粉丝6

喜欢7

评论3

等级： 博客 已

访问： 1万+

积分： 287

排名： 29万+



室内定位



最新文章

目标检测：SSD

python import引入不同路径下的模块

pytorch install安装

Multi-pseudo Regularized Label for Generated Samples in Person Re-Identification (多伪正则化标签(MpRL)、GAN)

配置opencv3.1+caffe

个人分类

计算机视觉1篇

深度学习1篇

opencv1篇

检测识别6篇

行人重识别3篇

展开

归档

2018年8月1篇

2018年6月2篇

2018年5月13篇

2017年8月1篇

2017年7月1篇

热门文章

全连接层的作用
阅读量：2605

FCN的学习及理解 (Fully Convolutional Networks for Semantic Segmentation)
阅读量：2500

(RegionProposal Network)RPN网络结构及详解
阅读量：1965

ubuntu下运行python提示: no module
named pip
阅读量：1726

GMS快速高鲁棒性特征学习
阅读量：1536

最新评论

GMS快速高鲁棒性特征学习

shi_shi08：请问博主可以发一份这个算法的程序给我吗？

(RegionProposal N...

baidu_26646129：博主，256-d的-d是什么意思啊？还请指教。

配置keras

weixin_41040035：必须要装tensorflow才能装keras 吗

诚聘 打字员

主要负责小说手稿题库录入、会议文件录入、快递物流单号录入等
兼职全职均可，时间地点自由，薪资待遇：2000字15元，4000字30元
以此类推，无排版要求，2个标点符号算一个字，多劳多得

正规网络兼职



联系我们



扫码联系客服



下载CSDN APP

 kefu@csdn.net

 400-660-0108

 QQ客服

 客服论坛

关于我们 招聘 广告服务 网站地图

 百度提供站内搜索 京ICP证09002463号

©2018 CSDN版权所有

经营性网站备案信息 网络110报警服务

北京互联网违法和不良信息举报中心

中国互联网举报中心