

# 【目标检测】SPPNet算法详解

转载 于 2019-07-26 11:29:28 发布 · 485 阅读 · 2 · 2

**Yolo-v5**

Yolo

YOLO (You Only Look Once) 是一种流行的物体检测和图像分割模型，由华盛顿大学的Joseph Redmon 和Ali Farhadi 开发。YOLO 于2015 年推出，因其高速和...

[一键部署](#)

## 1 基础框架

CNN网络需要固定尺寸的图像输入，**SPPNet**将任意大小的图像池化生成固定长度的图像表示，提升R-CNN检测的速度**24-100**倍。

固定图像尺寸输入的问题，截取的区域未涵盖整个目标或者缩放带来图像的扭曲。

事实上，CNN的卷积层不需要固定尺寸的图像，全连接层是需要固定大小输入的，因此提出了SPP层放到卷积层的后面，改的网络如下图所示：

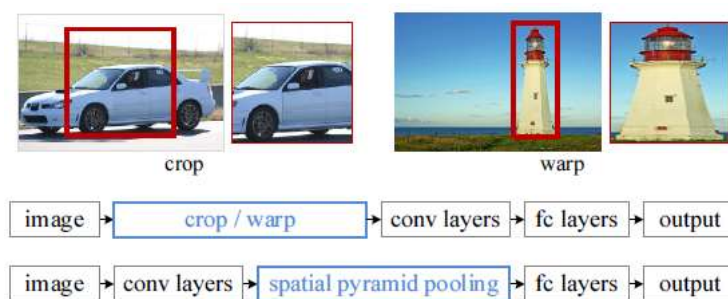


Figure 1: Top: cropping or warping to fit a fixed size. Middle: a conventional CNN. Bottom: our spatial pyramid pooling network structure. <https://arxiv.org/pdf/1704.04501v1.pdf>

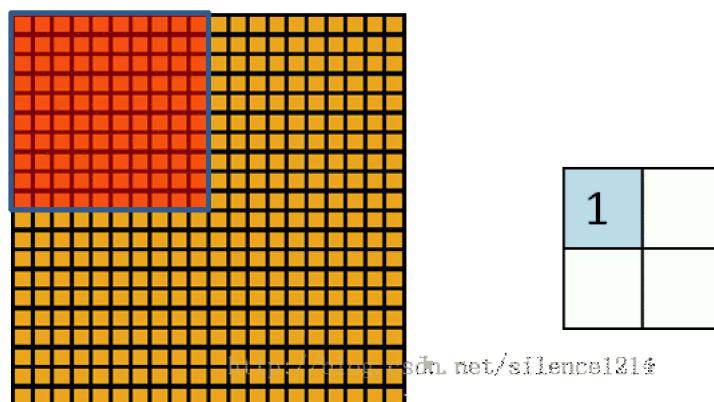
SPP是BOW的扩展，将图像从精细空间划分到粗糙空间，之后将局部特征聚集。在CNN成为主流之前，SPP在检测和分类的应用比较广泛。

**SPP的优点：**

- 1) 任意尺寸输入，固定大小输出
- 2) 层多
- 3) 可对任意尺度提取的特征进行池化。

## 2 理解池化

说下池化，其实池化很容易理解，先看图：



Convolved  
feature

Pooled  
feature

[http://blog.csdn.net/bryant\\_wang](http://blog.csdn.net/bryant_wang)

R-CNN提取特征比较耗时，需要对每个warp的区域进行学习，而SPPNet只对图像进行一次卷积，之后使用SPPNet在特征图取特征。结合EdgeBoxes提取的proposal，系统处理一幅图像需要0.5s。

SPP层的结构如下，将紧跟最后一个卷积层的池化层使用SPP代替，输出向量大小为 $kM$ ， $k=\text{filters}$ ,  $M=\text{bins}$ ，作为全连接层输入。至此，网络不仅可对任意长宽比的图像进行处理，而且可对任意尺度的图像进行处理。尺度在深层网络学习中也很重要。

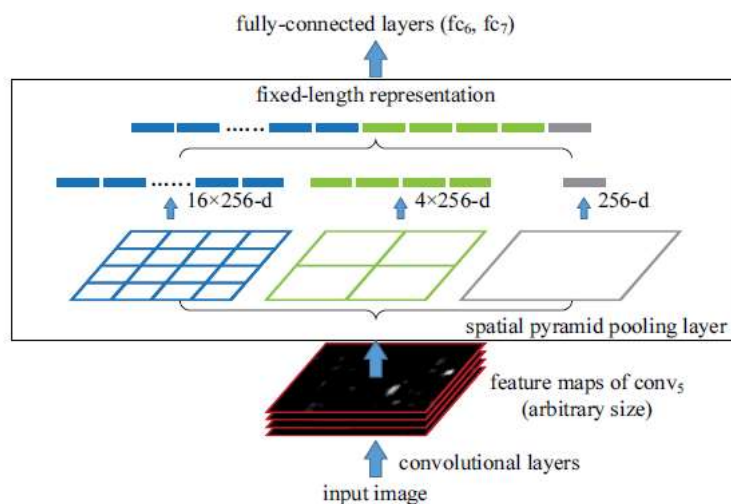


Figure 3: A network structure with a **spatial pyramid pooling layer**. Here 256 is the filter number of the  $\text{conv}_5$  layer, and  $\text{conv}_5$  is the last convolutional layer.

### 3 网络 训练

1. **multi-size训练**，输入尺寸在 $[180, 224]$ 之间，假设最后一个卷积层的输出大小为 $a \times a$ ，若给定金字塔层有 $n \times n$ 个bins，进行窗池化，窗口尺寸为 $\text{win} = \lceil a/n \rceil$ ，步长为 $\text{str} = \lceil a/n \rceil$ ，使用一个网络完成一个完整epoch的训练，之后切换到另外一个网络。只是在的时候用到多尺寸，测试时直接将SPPNet应用于任意尺寸的图像。

2.如果原图输入是224x224，对于conv5出来后的输出，是13x13x256的，可以理解成有256个这样的filter，每个filter对应一张13x13的reponse map。

**重点内容**如果像上图那样将reponse map分成4x4 2x2 1x1三张子图，做max pooling后，出来的特征就是固定长度的(16+4+1)那么多的维度了。如果原图的输入不是224x224，出来的特征依然是(16+4+1)x256

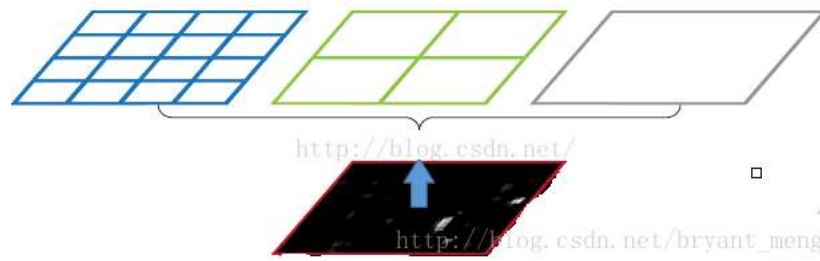
**直觉地说**，可以理解成将原来固定大小为(3x3)窗口的pool5改成了自适应窗口大小，窗口的大小和reponse map成比例，保证过pooling后出来的feature的长度是一致的。

#### 4 结论:

输入层：一张任意大小的图片,假设其大小为(w,h)。

输出层：21个神经元。

也就是我们输入一张任意大小的特征图的时候，我们希望提取出21个特征。空间金字塔特征提取的过程如下：



图片尺度划分

如上图所示，当我们输入一张图片的时候，我们利用不同大小的刻度，对一张图片进行了划分。上面示意图中，利用了三种不同的刻度，对一张输入的图片进行了划分，最后总共可以得到16+4+1=21个块，我们即将从这21个块中，每个块提取出一个特征，这样刚好就是我们要提取的21维特征向量。

1) **第一张图片**,我们把一张完整的图片，分成了16个块，也就是每个块的大小就是(w/4,h/4);

2) **第二张图片**，划分了4个块，每个块的大小就是(w/2,h/2);

3) **第三张图片**，把一整张图片作为了一个块，也就是块的大小为(w,h)

空间金字塔最大池化的过程，其实就是从这21个图片块中，分别计算每个块的最大值，从而得到一个输出神经元。最后把任意大小的图片转换成了一个固定大小的21维特征（当然你可以设计其它维数的输出，增加金字塔的层数，或者改变划分网格的大小）。

上面的三种不同刻度的划分，每一种刻度我们称之为：金字塔的一层，每一个图片块大小我们称之为：**windows size**了。我们希望，金字塔的某一层输出n\*n个特征，那么你就需要用windows size大小为：**(w/n,h/n)**进行池化了。

当我们有很多层网络的时候，当网络输入的是一张任意大小的图片，这个时候我们可以一直进行卷积、池化，直到网络的倒置的时候，也就是我们即将与全连接层连接的时候，就要使用金字塔池化，使得任意大小的特征图都能够转换成固定大小的特征向量，这就是空间金字塔池化的奥义（多尺度特征提取出固定大小的特征向量）

#### 5 拓展知识

BoW -> SPM

（SPP的思想来源于SPM，然后SPM的思想来源自BoW。关于BoW和SPM，找到了两篇相关的博文，就不在这里展开了。）

第九章三续：SIFT算法的应用—目标识别之Bag-of-words模型Spatial Pyramid 小结)

Spatial Pyramid Matching 小结

#### 6 SPPNet对ImageNet2012分类结果

1) 对已有网络增加SPP层提升性能, 包括ZF-5, Convnet-5, Overfeat-5/7, 单尺度和多尺度输入图像的实验结果top-1及top-5, 如下表所示, 第一个提出多尺度输入图像训练网络

		top-1 error (%)			
		ZF-5	Convnet*-5	Overfeat-5	Overfeat-7
(a)	no SPP	35.99	34.93	34.13	32.01
(b)	SPP single-size trained	34.98 (1.01)	34.38 (0.55)	32.87 (1.26)	30.36 (1.65)
(c)	SPP multi-size trained	34.60 (1.39)	33.94 (0.99)	32.26 (1.87)	29.68 (2.33)

		top-5 error (%)			
		ZF-5	Convnet*-5	Overfeat-5	Overfeat-7
(a)	no SPP	14.76	13.92	13.52	11.97
(b)	SPP single-size trained	14.14 (0.62)	13.54 (0.38)	12.80 (0.72)	11.12 (0.85)
(c)	SPP multi-size trained	13.64 (1.12)	13.33 (0.59)	12.33 (1.19)	10.95 (1.02)

2) 使用全图作为SPPNet的输入及224\*224图像中心的crop对比, 网络使用ZF-5, Overfeat-7, 结果如下表

SPP on	test view	top-1 val
ZF-5, single-size trained	1 crop	38.01
ZF-5, single-size trained	1 full	37.55
ZF-5, multi-size trained	1 crop	37.57
ZF-5, multi-size trained	1 full	37.07
Overfeat-7, single-size trained	1 crop	33.18
Overfeat-7, single-size trained	1 full	32.72
Overfeat-7, multi-size trained	1 crop	32.57
Overfeat-7, multi-size trained	1 full	31.25

Table 3: Error rates in the validation set of ImageNet 2012 using a single view. The images are resized so  $\min(w, h) = 256$ . The crop view is the central  $224 \times 224$  of the image.

3) 与其他方法在ImageNet分类的对比

rank	team	top-5 test
1	GoogLeNet [32]	6.66
2	VGG [33]	7.32
3	<u>ours</u>	<u>8.06</u>
4	Howard	8.11
5	DeeperVision	9.50
6	NUS-BST	9.79
7	TTIC_ECP	10.22

Table 5: The competition results of ILSVRC 2014 classification [26]. The best entry of each team is listed.

SPPNet for Object Detection, R-CNN重复使用深层卷积网络在~2k个窗口上提取特征, 特征提取非常耗时。SPPNet比较耗内存, 卷积计算对整幅图像只进行一次, 之后使用spp将窗口特征图池化为一个固定长度的特征表示。

## 7 检测算法

使用ss生成~2k个候选框，缩放图像 $\min(w,h)=s$ 之后提取特征，每个候选框使用一个4层的空间金字塔池化特征，网络使用的5的SPPNet形式。之后将12800d的特征输入全连接层，SVM的输入为全连接层的输出。SVM如下使用，正样本为groundtruth，负样本与正样本的IoU<0.3，训练20类样本只需1h。

使用ZF-5与R-CNN对比实验结果如下：

	SPP (1-sc) (ZF-5)	SPP (5-sc) (ZF-5)	R-CNN (ZF-5)
ftfc <sub>7</sub>	54.5	<u>55.2</u>	55.1
ftfc <sub>7</sub> bb	58.0	<u>59.2</u>	59.2
conv time (GPU)	0.053s	0.293s	14.37s
fc time (GPU)	0.089s	0.089s	0.089s
total time (GPU)	0.142s	0.382s	14.46s
speedup (vs. RCNN)	102×	38×	-

Table 10: Detection results (mAP) on Pascal VOC 2007, using the same pre-trained model of SPP (ZF-5).

## 8 如何训练网络

Reference Link <http://zhangliliang.com/2014/09/13/paper-note-sppnet/>

理论上说，SPP-net支持直接以多尺度的原始图片作为输入后直接BP即可。实际上，caffe等实现中，为了计算的方便，输入定了尺度的了。

所以为了使得在固定输出尺度的情况下也能够做到SPP-net的效果，就需要定义一个新的SSP-layer

作者以输入224x224举例，这时候conv5出来的reponse map为13x13，计算出来的步长如下图所示。

[pool3x3] type=pool pool=max inputs=conv5 sizeX=5 stride=4	[pool2x2] type=pool pool=max inputs=conv5 sizeX=7 stride=6	[pool1x1] type=pool pool=max inputs=conv5 sizeX=13 stride=13
[fc6] type=fc outputs=4096 inputs=pool3x3,pool2x2,pool1x1		

具体的计算方法，看一眼2.3的Single-size training部分就明白了。如果输入改成180x180，这时候conv5出来的reponse map: 10x10，类似的方法，能够得到新的pooling参数。

两种尺度下，在SSP后，输出的特征维度都是(9+4+1)x256，之后接全连接层即可。

训练的时候，224x224的图片通过随机crop得到，180x180的图片通过缩放224x224的图片得到。之后，迭代训练，即用224的图训练一个epoch，之后180的图片训练一个epoch，交替地进行。

## 9 如何测试网络

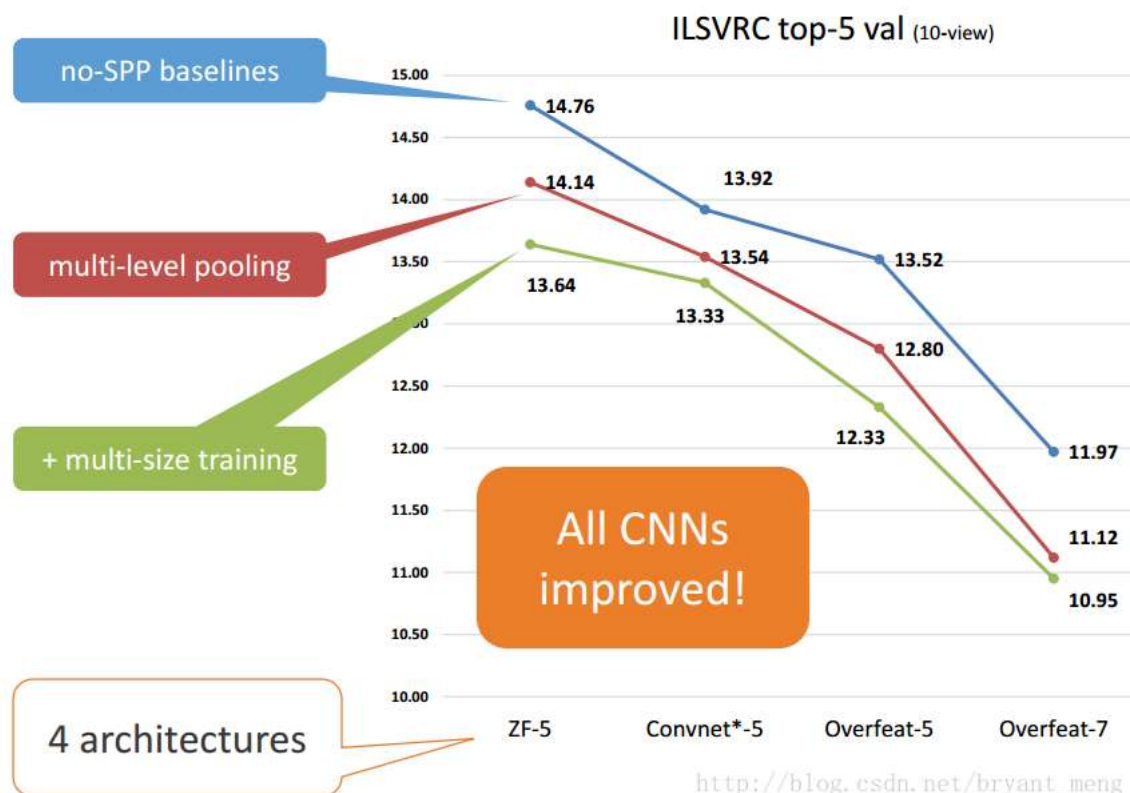
作者说了一句话：Note that the above single/multi-size solutions are for training only. At the testing stage, it is straightforward to apply SPP-net on images of any sizes.

笔者觉得没有那么简单吧，毕竟caffe对于test网络也是有固定尺度的要求的。



## 10 实验

之后是大量的实验。分类实验，如下图，一句话概括就是，都有提高。



## 11 一些细节

为了保证公平，test时候的做法是将图片缩放到短边为256，然后取10crop。这里的金字塔为{6x6 3x3 2x2 1x1}（笔者注意到里算是增加了特征，因为常规pool5后来说，只有6x6；这里另外多了9+4+1个特征）

作者将金字塔减少为{4x4 3x3 2x2 1x1}，这样子，每个filter的feature从原来的36减少为30，但依然有提高。（笔者认为这个保留意见比较好）

其实这部分的实验比较多，详见论文，不在这里写了。

在ILSVRC14上的cls track，作者是第三名

## 12 定位实验

这里详细说说笔者较为关心的voc07上面的定位实验,用来对比的对象是RCNN。

### 方法简述：

提取region proposal部分依然用的是**selective search**，CNN部分，结构用的是**ZF-5**（单尺度训练），金字塔用了{6x6 3x3 2x2 1x1}，共50个bin，分类器也是用了**SVM**，后处理也是用了**cls-specific regression**，所以**主要差别是在第二步**，做出的主要改变是SPP-net能够一次得到整个feature map，大大减少了计算proposal的特征时候的运算开销。

### 具体做法：

将图片缩放到 $s \in \{480, 576, 688, 864, 1200\}$ 的大小，于是得到了6个feature map。尽量让region在s集合中对应的尺度接近224x224，然后选择对应的feature map进行提取。（具体如何提取？后面的附录会说）最后效果如图：

	SPP (1-sc)	SPP (5-sc)	R-CNN
pool <sub>5</sub>	43.0	<u>44.9</u>	44.2
fc <sub>6</sub>	42.5	44.8	<u>46.2</u>
ftfc <sub>6</sub>	52.3	<u>53.7</u>	53.1
ftfc <sub>7</sub>	54.5	<u>55.2</u>	54.2
ftfc <sub>7</sub> bb	58.0	<b>59.2</b>	58.5
conv time (GPU)	0.053s	0.293s	8.96s
fc time (GPU)	0.089s	0.089s	0.07s
total time (GPU)	0.142s	0.382s	9.03s
speedup (vs. RCNN)	64×	24×	-

Table 9: Detection results (mAP) on Pascal VOC 2007. “ft” and “bb” denote fine-tuning and bounding box regression. [http://blog.csdn.net/bryant\\_meng](http://blog.csdn.net/bryant_meng)

准确率从58.5提高到了59.2，而且速度快了24x，如果用两个模型综合，又提高了一点，到60.9。

method	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
SPP-net (1)	59.2	68.6	69.7	57.1	41.2	40.5	66.3	71.3	72.5	34.4	67.3	61.7	63.1	71.0	69.8	57.6	29.7	59.0	50.2	65.2	68.0
SPP-net (2)	59.1	65.7	71.4	57.4	42.4	39.9	67.0	71.4	70.6	32.4	66.7	61.7	64.8	71.7	70.4	56.5	30.8	59.9	53.2	63.9	64.6
combination	60.9	68.5	71.7	58.7	41.9	42.5	67.7	72.1	73.8	34.7	67.0	63.4	66.0	72.5	71.3	58.9	32.8	60.9	56.1	67.9	68.8

Table 11: Detection results on VOC 2007 using model combination. The results of both models use “ftfc<sub>7</sub> bb”. [http://blog.csdn.net/bryant\\_meng](http://blog.csdn.net/bryant_meng)

### 13 附录

如何将图像的ROI映射到feature map?

说实话，笔者还是没有完全看懂这里的操作。先记录目前能够理解的部分。总体的映射思路为：In our implementation, we project the corner point of a window onto a pixel in the feature maps, such that this corner point (in the image domain) is closest to the center of the receptive field of that pixel.

略绕，我的理解是：

映射的是ROI的两个角点，左上角和右下角，这两个角点就可以唯一确定ROI的位置了。

将feature map的pixel映射回来图片空间

从映射回来的pixel中选择一个距离角点最近的pixel，作为映射。

如果以ZF-5为例子，具体的计算公式为：

image domain, and the effective stride of conv<sub>5</sub> in the image domain is 16. Denote  $x$  and  $x_{conv5}$  as the coordinates in the image domain and conv<sub>5</sub> (we use the MATLAB convention, i.e.,  $x$  starts from 1). We project the top-left corner by:  $x_{conv5} = \lfloor (x - 139/2 + 63)/16 \rfloor + 1$ . Here 139/2 is the radius of the receptive field, 16 is the effective stride, and 63 is an offset. The offset is computed by the top-left corner of the receptive field in the image domain. Similarly, we project the bottom-right corner by:  $x_{conv5} = \lceil (x + 139/2 - 75)/16 \rceil - 1$ . Here

这里有几个变量

139代表的是感受野的直径，计算这个也需要一点技巧了：如果一个filter的kernel size= $x$ , stride= $y$ ，而输出的response map的长 $n$ ，那么其对应的感受野的长度为： $n+(n-1)(stride-1)+2((kernel size-1)/2)$

Handwritten calculations for SPPNet layers:

- conv5 (3,1)  $1 + 0 \times 0 + 2 \times 1 = 3 + 2 = 5$
- conv4 (3,1)  $3 + 2 \times 0 + 2 \times 1 = 3 + 2 + 2 = 7$
- conv3 (3,1)  $5 + 4 \times 0 + 2 \times 1 = 5 + 2 = 7$
- pool2 (3,2)  $7 + 6 \times 1 + 2 \times 1 = 7 + 6 + 2 = 15$
- conv2 (5,2)  $15 + 14 \times 1 + 2 \times 2 = 15 + 14 + 4 = 33$
- pool1 (3,2)  $33 + 32 \times 1 + 2 \times 1 = 33 + 32 + 2 = 67$
- conv1 (7,2)  $67 + 66 \times 1 + 2 \times 3 = 67 + 66 + 6 = 139$

http://blog.csdn.net/bryant\_meng

16是effective stride，这里笔者理解为，将conv5的pixel映射到图片空间后，两个pixel之间的stride。（计算方法，所有stride乘，对于ZF-5  $2 \times 2 \times 2 \times 2 = 16$ ）

63和75怎么计算，还没有推出来

## 14 再次理解SPP

参考:<http://blog.csdn.net/xyy19920105/article/details/50817725>

SPP网络，我不得不要先说，这个方法的思想在Fast RCNN，Faster RCNN上都起了举足轻重的作用。SPP网络主要是解决网络固定输入层尺寸的这个限制，也从各个方面说明了不限制输入尺寸带来的好处。

文章在一开始的时候就说明了目前深度网络存在的弊端：如果固定网络输入的话，要么选择**crop策略**，要么选择**warp策略**，就是从一个大图扣出网络输入大小的patch（比如 $227 \times 227$ ），而warp则是把一个bounding box的内容resize成 $227 \times 227$ 。无论是哪种策略，都能很明显看出有影响网络训练的不利因素，比如crop就有可能crop出object的一个部分，而无法准确训练出类别，而warp改变object的正常宽高比，使得训练效果变差。接着，分析了出深度网络需要固定输入尺寸的原因是因为有全链接层，不过在那时候，还没有FCN的思想，那如何去做才能使得网络不受输入尺寸的限制呢？Kaiming He大神就想出，用**不同尺度的pooling**来pooling出**固定尺度大小的feature map**，这样就可以不受全链接层约束任意更改输入尺度了。

### SPP网络的核心思想：

通过对feature map进行相应尺度的pooling，使得能pooling出 $4 \times 4$ ， $2 \times 2$ ， $1 \times 1$ 的feature map，再将这些feature map concat成与下一层全链接层相连。这样就消除了输入尺度不一致的影响。

训练的时候就用常规方法训练，不过由于不受尺度的影响，可以进行多尺度训练，即先resize成几个固定的尺度，然后用SPP进行训练，学习。这里讲了这么多，实际上我想讲的是下面的东西，SPP如何用在检测上面。

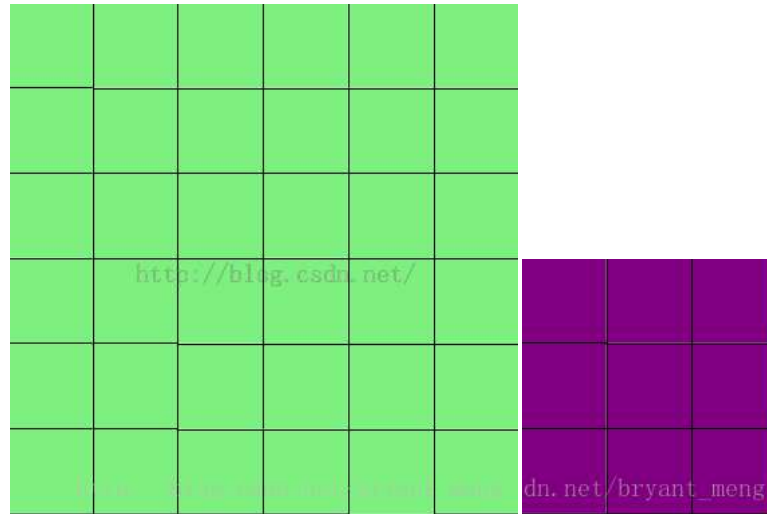


论文中实际上我觉得最关键的地方是提出了一个如何将原图的某个region映射到conv5的一种机制，虽然，我并不是太认可这射机制，等下我也会说出我认为合理的映射方法。论文中是如何映射的，实际上我也是花了好久才明白。

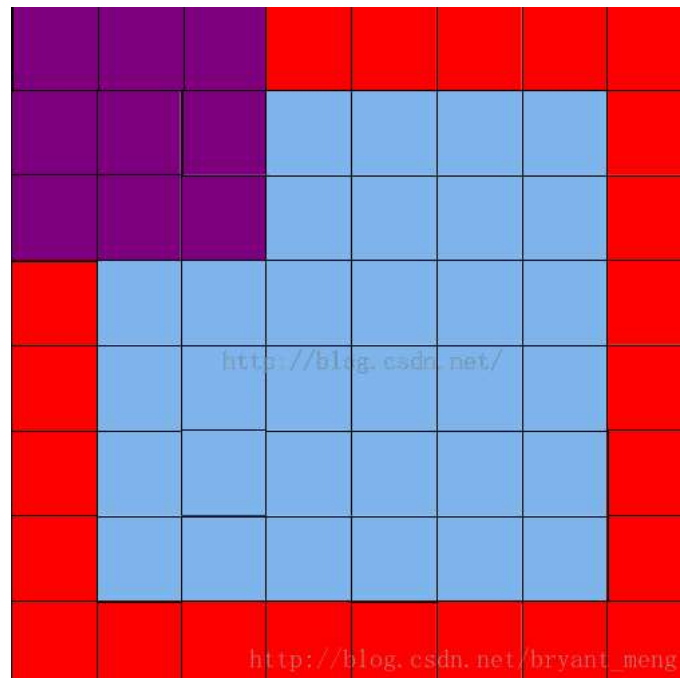
首先，我想先说明函数这个东东，当然我不是通过严谨的定义来说明。

什么是 $y=f(x)$ ，我认为只要输入 $x$ ，有一组固定的操作 $f$ ，然后产生一个对应的 $y$ ，这样子就算是函数。根据输入有一个——对应出，这就是函数。这样理解的话，卷积也是函数，pooling也是函数。当然我并不想说明函数是什么，什么是函数，实际上我想强是——对应这样的关系。大家都知道，现在默许的无论是卷积还是pooling（无stride），都会加相应的pad，来使得卷积后的尺寸积前相同，当然这种做法还有个好处就是使得边缘不会只被卷积一次就消失了~

这样子的话，实际上原图与卷积后的图就是一一对应的关系。原图的每一个点（包括边缘）都可以卷积得到一个新的点，这弱一对应了。如下图所示（自己画得太丑）：



绿色部分是图片，紫色部分是卷积核。

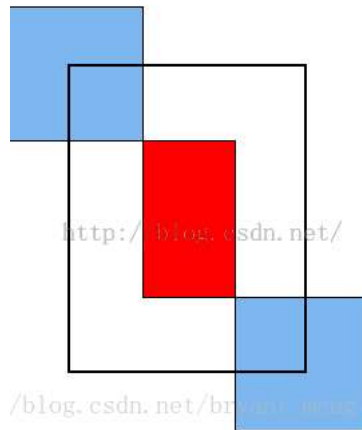


如上图可以看出，蓝色的区域是原图区域，而红色的区域是padding区域，紫色的是卷积核。卷积后得到的区域与原区域是一应的。

而卷积或pooling增加stride的话就相当于与原图先进行卷积或池化，再进行sampling，这还是能一一对应的，就这样原图的某个就可以通过除以网络的所有stride来映射到conv5后去区域。

终于把这里讲出来了，大家如果直接按照函数的一一对应关系去理解，很容易理解为什么原图的区域除以所有的stride就是conv5的区域。这样子就可以在原图上的一些操作放在conv5上进行，这样可以减小任务复杂度。

不过，我并不是太认可这种映射机制，这种映射只能是点到点的关系，不过我觉得从原图的某个区域R映射到conv5的区域r，对R敏感，换句话说，应该r感受野应该与R有交集。这样子的话，示意图如下：



其中蓝色的为conv的神经元感受野，红色的是原图的某个感兴趣区域，而黑色框我才认为是要映射到conv5的区域。

使用SPP进行检测，先用提候选proposals方法（selective search）选出候选框，不过不像RCNN把每个候选区域给深度网络征，而是整张图提一次特征，再把候选框映射到conv5上，因为候选框的大小尺度不同，映射到conv5后仍不同，所以需要再通过层提取到相同维度的特征，再进行分类和回归，后面的思路和方法与RCNN一致。

实际上这样子做的话就比原先的快很多了，因为之前RCNN也提出了这个原因就是深度网络所需要的感受野是非常大的，这话需要每次将感兴趣区域放大到网络的尺度才能卷积到conv5层。这样计算量就会很大，而SPP只需要计算一次特征，剩下的只在conv5层上操作就可以了。

当然即使是这么完美的算法，也是有它的瑕疵的，可能Kaiming He大神太投入 SPP的功效了，使得整个流程框架并没有变得完美。首先在训练方面，SPP没有发挥出它的优势，依旧用了传统的训练方法，这使得计算量依旧很大，而且分类和bounding box回归问题也可以联合学习，使得整体框架更加完美。这些Kaiming He都给忽略了，这样也就有了第二篇神作 Fast RCNN。

您可能感兴趣的与本文相关的镜像



Yolo-v5 Yolo

YOLO (You Only Look Once) 是一种流行的物体检测和图像分割模型，由华盛顿大学的Joseph Redmon 和Ali Farhadi 开发。YOLO 于2015 年推出，因其高速和...

一键部署

觉得还不错? 一键收藏



monk1992

关注

2

2

0

分享

...

专栏目录

## 目标检测——SPPNet【含全网最全翻译】

PRIS-SCMonkey的博客

文章目录0. 摘要1. 简介2. 基于空间金字塔池化的深度网络2.1 卷积层和特征图2.2 空间金字塔池化层2.3 网络的训练2.3.1 单一尺寸训练2.3.2 多尺寸训练3. 用于图像分类的SPP-

SPP Net 目标检测网络学习笔记（附代码）

PLANTTHESON的博客

SPP Net 目标检测网络学习笔记（附代码）

## 深入浅出之SPPNET网络

深入浅出之SPPNET网络 一、定义 SPP-Net(Spatial Pyramid Pooling Network),即空间金字塔池化网络,是由何凯明等人在2015年提出的一种深度学习框架,旨在解决卷积神经网络

## 目标检测之SSPNet

SPPNet在R-CNN的基础上提出了改进,通过候选区域和feature map的映射,配合SPP层的使用从而达到了CNN层的共享计算,减少了运算时间,后面的FastR-CNN等也是受SPPNet

<p>什么是 SPP-Net <b>最新发布</b></p> <p>SPP-Net (空间金字塔池化网络) 是2014年提出的计算机视觉重要创新, 通过引入SPP层使CNN能处理任意尺寸图像, 解决了传统模型需固定输入尺寸的局限. 该网络在ILSVF</p>	彬彬侠的博客
<p><b>目标检测网络之SPP-net详解</b></p> <p>SPP-net<b>详解</b>: 简要概述: SPP-net网络是基于R-CNN结构改进, 如详细了解SPP-net请先阅读<b>目标检测</b>开山之作R-CNN, 文章链接: <a href="https://blog.csdn.net/a24444054d/article/">https://blog.csdn.net/a24444054d/article/</a></p>	a24444054d的博客
<p>论文阅读    <b>目标检测系列 —— SPPNet详解_spp net</b></p> <p>1 <b>SPPNet</b>网络介绍 1.1 <b>SPPNet</b>网络处理流程 1.候选框提取:使用<b>算法</b>selective search获取输入图片的候选框,提取约2000个候选区域 2.CNN+SPP+FC(与RCNN不同之处): 把整</p>	
<p><b>目标检测系列之SPPNet——让深度卷积网络告别输入尺寸枷锁</b></p> <p><b>SPPNet</b>,全称Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,出自何恺明等大神2014年的同名论文。其核心创新是一个叫做空间金字塔池化的结</p>	
<p><b>【目标检测算法】SPP-Net学习笔记</b></p> <p>在现有的CNN中, 对于结构已经确定的网络, 需要输入一张固定大小的图片, 比如224*224, 32*32,96*96等。这样对于我们希望检测各种大小的图片的时候, 需要经过裁剪, 且</p>	littlemichelle
<p>两阶段<b>目标检测原理详解--SPPNet</b></p> <p>目录 一、RCNN的瓶颈 二、SPP的改进 (一) 共享卷积网络 SPP-Net是2014年有名的两阶段<b>目标检测算法</b>。论文: <a href="https://arxiv.org/pdf/1406.4729.pdf">https://arxiv.org/pdf/1406.4729.pdf</a> 一、RCNN的瓶颈 在RC</p>	AI学长
<p>空间金字塔池化网络<b>SPPNet详解_spp-net算法</b></p> <p>SPP-Net解决了深度神经网络需要固定输入尺寸的问题,通过空间金字塔池化层,允许网络接受不同尺寸的输入图像,保持输出尺寸固定,避免了传统CNN预处理时的crop和wrap操</p>	
<p><b>【目标检测】SPP-Net论文理解(超详细版本)</b></p> <p>SPP-Net是出自2015年发表在IEEE上的论文,全名为《Spatial Pyramid Pooling in Deep ConvolutionalNetworks for Visual Recognition》,作者为何恺明、张祥雨、任少卿、孙剑</p>	
<p><b>目标检测经典算法和API详解 (笔记)</b></p> <p>文章目录商品<b>目标检测</b>1. <b>目标检测</b>概述1.1.项目演示介绍学习目标1.1.1 项目演示1.1.2 项目结构1.1.3 项目安排1.2 图像识别背景学习目标1.2.1 图像识别三大任务1.2.2 图像识别</p>	源仔
<p>经典<b>目标检测算法详解</b> (R-CNN、SPP-Net、Fast R-CNN、Faster R-CNN、YOLO、SSD)</p> <p>1 <b>目标检测算法</b>分类 两步走的<b>目标检测</b>: 先进行区域推荐, 而后进行目标分类 代表: R-CNN、SPP-net、Fast R-CNN、Faster R-CNN 端到端的<b>目标检测</b>: 采用一个网络一步到</p>	tigerlib的博客
<p>[AI深度追问]空间金字塔池化网络(<b>SPPNet</b>):用于图像分类和<b>目标检测</b>的深...</p> <p>空间金字塔池化网络(Spatial Pyramid PoolingNetwork,<b>SPPNet</b>)是一种用于图像分类和<b>目标检测</b>的深度神经网络,可以处理任意大小的输入图像。<b>SPPNet</b>通过空间金字塔池化(S</p>	
<p><b>目标检测(R-CNN,SPPNET,FAST-RCNN,FASTER-RCNN)</b></p> <p>一文搞懂rcnn, <b>sppnet</b>, fastrcnn, fasterrcnn。</p>	2303_79620604的博客
<p>04<b>目标检测-Two-stage的目标检测算法</b></p> <p>前面我们对进行了对比分析, 同时上一篇我们介绍了传统<b>算法</b>在<b>目标检测</b>的原理和效果, 此篇我们记录了我学习<b>目标检测算法</b>的深度, 深入讲述一下深度学习<b>算法</b>在<b>目标检测</b>的</p>	qq_41946216的博客
<p><b>【目标检测算法系列】二、SPP-Net算法</b></p> <p>上次我们介绍了R-CNN<b>算法</b>【<b>目标检测算法系列</b>】一、R-CNN<b>算法</b> 这次介绍下SPP-Net<b>算法</b> SPP-Net<b>算法</b>是在R-CNN的基础上做了相关改进 我们知道, R-CNN<b>算法</b>有几个缺</p>	u010057965的专栏
<p><b>目标检测——SPPNet算法解读</b></p> <p><b>SPPNet算法</b>解读</p>	lishanlu136的博客
<p><b>【目标检测】SPPnet论文详解</b></p> <p>原文: <a href="https://www.jianshu.com/p/90f9d41c6436">https://www.jianshu.com/p/90f9d41c6436</a> <b>SPPnet</b>是<b>目标检测</b>领域不可忽略的一篇文章, 中国人何恺明大神的作品, 阅读起来感觉亲 (简) 切 (单) 多了。在我看来, <b>SF</b></p>	weixin_40500230的博客
<p>论文阅读    <b>目标检测系列 —— SPPNet详解</b></p> <p>1 <b>SPPNet</b>网络介绍 1.1 <b>SPPNet</b>网络处理流程 1. 候选框提取: 使用<b>算法</b>selective search获取输入图片的候选框, 提取约2000个候选区域 2. CNN+SPP+FC (与RCNN不同之处)</p>	magic_1l的博客
<p><b>SPPnet 详解</b></p> <p>RCNN系列: RCNN, <b>SPPNet</b>, Fast RCNN, Faster RCNN, R-FCN。 前不久刚刚看完rcnn和fast rcnn, 对<b>目标检测</b>的学习更进一步。<b>sppNet</b>简介: 作者: 何凯明 2016</p>	wait_ButterFly的博客
<p><b>SPPNet详解 (白话讲解——附图文) 热门推荐</b></p> <p><b>SPPNet</b>是何凯明大神提出的, 为了解决R-CNN中速度慢问题。在神经网络中输入图片的尺寸必须是固定的, 这是因为在设计的时候FC层中神经元的个数都是固定的, 导致输</p>	weixin_45074568的博客
<p>CV之DL之<b>SPPNet</b>: SPP-Net<b>算法</b>的简介(论文介绍)、架构<b>详解</b>、案例应用等配图集合之详细攻略</p> <p>CV之DL之<b>SPPNet</b>: SPP-Net<b>算法</b>的简介(论文介绍)、架构<b>详解</b>、案例应用等配图集合之详细攻略 目录 相关论文 SPP-Net<b>算法</b>的简介 SPP-Net关键步骤 相关文章 DL之SPP-Ne</p>	头部AI社区如有邀博主AI主题演讲请私信一心比天高...
<p>深度学习<b>目标检测算法与代码复现详解</b></p> <p>深度学习的<b>目标检测算法</b>主要分为两大类: Two stage和One stage方法。Two stage方法将检测过程分为两个阶段。首先是Region Proposal生成阶段, 该阶段使用卷积神经网络</p>	

