SPPNet 论文翻译-空间金字塔池化 Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

用于视觉识别的深度卷积网络空间金字塔池化方法

Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun

摘要

当前深度卷积神经网络（CNNs）都需要输入的图像尺寸固定（比如 224×224）。这种人为的需要导致面对任意尺寸和比例的图像或子图像时降低识别的精度。本文中，我们给网络配上一个叫做“空间金字塔池化”(spatial pyramid pooling,)的池化策略以消除上述限制。这个我们称之为 SPP-net 的网络结构能够产生固定大小的表示（representation）而不关心输入图像的尺寸或比例。金字塔池化对物体的形变十分鲁棒。由于诸多优点，SPP-net 可以普遍帮助改进各类基于 CNN 的图像分类方法。在 ImageNet2012 数据集上，SPP-net 将各种 CNN 架构的精度都大幅提升，尽管这些架构有着各自不同的设计。在 PASCAL VOC 2007 和 Caltech101 数据集上，SPP-net 使用单一全图像表示在没有调优的情况下都达到了最好成绩。SPP-net 在物体检测上也表现突出。使用 SPP-net，只需要从整张图片计算一次特征图（feature map），然后对任意尺寸的区域（子图像）进行特征池化以产生一个固定尺寸的表示用于训练检测器。这个方法避免了反复计算卷积特征。在处理测试图像时，我们的方法在 VOC2007 数据集上，达到相同或更好的性能情况下，比 R-CNN 方法快 24-102 倍。在 ImageNet 大规模视觉识别任务挑战（ILSVRC）2014 上，我们的方法在物体检测上排名第 2，在物体分类上排名第 3，参赛的总共有 38 个组。本文也介绍了为了这个比赛所作的一些改进。

1. 简介

我们看到计算机视觉领域正在经历飞速的变化，这一切得益于深度卷积神经网络（CNNs）[1]和大规模的训练数据的出现[2]。近来深度网络对图像分类 [3][4][5][6]，物体检测 [7][8][5] 和其他识别任务 [9][10][11][12]，甚至很多非识别类任务上都表现出了明显的性能提升。

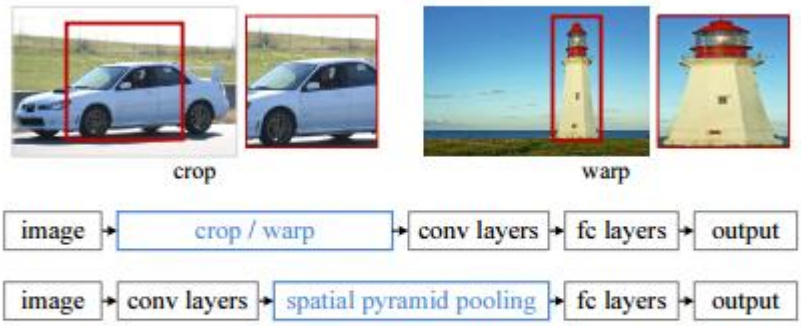

Figure 1: Top: cropping or warping to fit a fixed size. Middle: a conventional CNN. Bottom: our spatial pyramid pooling network structure.

然而，这些技术再训练和测试时都有一个问题，这些流行的 CNNs 都需要输入的图像尺寸是固定的（比如 224×224），这限制了输入图像的长宽比和缩放尺度。当遇到任意尺寸的图像是，都是先将图像适应成固定尺寸，方法包括裁剪[3][4]和变形[13][7]，如图 1（上）所示。但裁剪会导致信息的丢失，变形会导致位置信息的扭曲，就会影响识别的精度。另外，一个预先定义好的尺寸在物体是缩放可变的时候就不适用了。

那么为什么 CNNs 需要一个固定的输入尺寸呢？CNN 主要由两部分组成，卷积部分和其后的全连接部分。卷积部分通过滑窗进行计算，并输出代表激活的空间排布的特征图（feature map）（图 2）。事实上，卷积并不需要固定的图像尺寸，他可以产生任意尺寸的特征图。而另一方面，根据定义，全连接层则需要固定的尺寸输入。因此固定尺寸的问题来源于全连接层，也是网络的最后阶段。本文引入一种空间金字塔池化(spatial pyramid pooling，SPP)层以移除对网络固定尺寸的限制。尤其是，将 SPP 层放在最后一个卷积层之后。SPP 层对特征进行池化，并产生固定长度的输出，这个输出再喂给全连接层（或其他分类器）。换句话说，在网络层次的较后阶段（也就是卷积层和全连接层之间）进行某种信息“汇总”，可以避免在最开始的时候就进行裁剪或变形。图 1（下）展示了引入 SPP 层之后的网络结构变化。我们称这种新型的网络结构为 SPP-net。

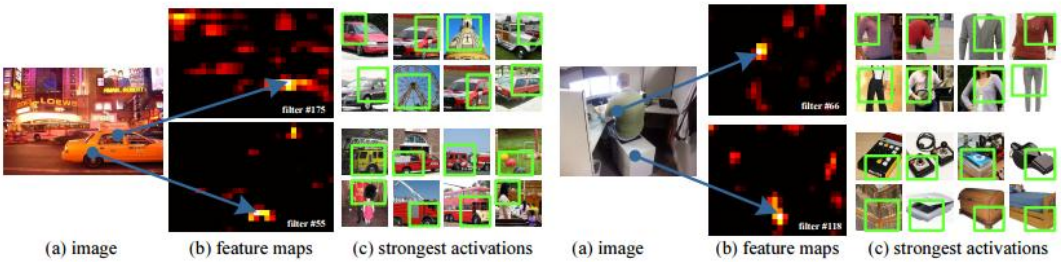

Figure 2: Visualization of the feature maps. (a) Two images in Pascal VOC 2007. (b) The feature maps of some $conv_5$ filters. The arrows indicate the strongest responses and their corresponding positions in the images. (c) The ImageNet images that have the strongest responses of the corresponding filters. The green rectangles mark the receptive fields of the strongest responses.

空间金字塔池化[14][15]（普遍称谓：空间金字塔匹配 spatial pyramid matching, SPM[15]），是一种词袋(Bag-of-Words, BoW)模型的扩展。池袋模型是计算机视觉领域最成功的方法之一。它将图像切分成粗糙到精细各种级别，然后整合其中的局部特征。在 CNN 之前，SPP 一直是各大分类比赛[17][18][19]和检测比赛（比如[20]）的冠军系统中的核心组件。对深度 CNNs 而言，SPP 有几个突出的优点：1）SPP 能在输入尺寸任意的情况下产生固定大小的输出，而以前的深度网络[3]中的滑窗池化(sliding window pooling)则不能；2）SPP 使用了多级别的空间箱(bin)，而滑窗池化则只用了一个窗口尺寸。多级池化对于物体的变形十分鲁

棒[15]；3）由于其对输入的灵活性，SPP 可以池化从各种尺度抽取出来的特征。通过实验，我们将展示影响深度网络最终识别精度的所有这些因素。

SPP-net 不仅仅让测试阶段允许任意尺寸的输入能够产生表示(representations)，也允许训练阶段的图像可以有各种尺寸和缩放尺度。使用各种尺寸的图像进行训练可以提高缩放不变性，以及减少过拟合。我们开发了一个简单的多尺度训练方法。为了实现一个单一的能够接受各种输入尺寸的网络，我们先使用分别训练固定输入尺寸的多个网络，这些网络之间共享权重（Parameters），然后再一起来代表这个单一网络（译者注：具体代表方式没有说清楚，看后面怎么说吧）。每个 epoch，我们针对一个给定的输入尺寸进行网络训练，然后在下一个 epoch 再切换到另一个尺寸。实验表明，这种多尺度训练和传统的单一尺度训练一样可以瘦脸，并且能达到更好的测试精度。

SPP 的优点是与各类 CNN 设计是正交的。通过在 ImageNet2012 数据集上进行一系列可控的实验，我们发现 SPP 对[3][4][5]这些不同的 CNN 架构都有提升。这些架构有不同的特征数量、尺寸、滑动距离（strides）、深度或其他的设计。所以我们有理由推测 SPP 可以帮助提升更多复杂的（更大、更深）的卷积架构。SPP-net 也做到了 Caltech101 [21]和 Pascal VOC 2007 [22]上的最好结果，而只使用了一个全图像表示，且没有调优。

在图像检测方面，SPP-net 也表现优异。目前领先的方法是 R-CNN[7]，候选窗口的特征是借助深度神经网络进行抽取的。此方法在 VOC 和 ImageNet 数据集上都表现出了出色的检测精度。但 R-CNN 的特征计算十分耗时，因为他对每张图片中的上千个变形后的区域的像素反复调用 CNN。本文中，我们展示了我们只需要在整张图片上运行一次卷积网络层（不关心窗口的数量），然后再使用 SPP-net 在特征图上抽取特征。这个方法缩减了上百倍的耗时。在特征图（而不是图像区域）上训练和运行检测器是一个很受欢迎的想法[23][24][20][5]。但 SPP-net 延续了深度 CNN 特征图的优势，也结合了 SPP 兼容任意窗口大小的灵活性，所以做到了出色的精度和效率。我们的实验中，基于 SPP-net 的系统（建立在 R-CNN 流水线上）比 R-CNN 计算特征要快 24-120 倍，而精度却更高。结合最新的推荐方法 EdgeBoxes[25]，我们的系统达到了每张图片处理 0.5s 的速度（全部步骤）。这使得我们的方法变得更加实用。

本论文的一个早先版本发布在 ECCV2014 上。基于这个工作，我们参加了 ILSVRC 2014 [26]，在 38 个团队中，取得了物体检测第 2 名和图像分类第 3 名的成绩。针对 ILSVRC 2014 我们也做了很多修改。我们将展示 SPP-nets 可以将更深、更大的网络的性能显著提升。进一步，受检测框架驱动，我们发现借助灵活尺寸窗口对特征图进行多视角测试可以显著提高分类精度。本文对这些改动做了更加详细的说明。另外，我们将代码放在了以方便大家研究（http://research.microsoft.com/en-us/um/people/kahe/，译者注：已失效）

2. 基于空间金字塔池化的深度网络

2.1 卷积层和特征图

在颇受欢迎的七层架构中[3][4]中，前五层是卷积层，其中一些后面跟着吃常委曾。从他们也使用滑窗的角度来看，这些池化层也可以认为是“卷积的”。最后两层是全连接的，跟着一个 N 路 softmax 输出，其中 N 是类别的数量。上述的深度网络需要一个固定大小的图像尺寸。然后，我们注意到，固定尺寸的要求仅仅是因为全连接层的存在导致的。另一方面，卷积层使用滑动的特征过滤器，它们的输出基本保持了原始输入的比例关系。它们的输出就是特征图[1]-它们不仅涉及响应的强度，还包括空间位置。图 2 中，我们可视化了一些特征图。这些特征图来自于 conv5 层的一些过滤器。图 2（c）显示了 ImageNet 数据集中激活最强的若干图像。可以看到一个过滤器能够被一些语义内容激活。例如，第 55 个过滤器（图 2，

左下）对圆形十分敏感；第 66 层（图 2，右上）对 a^形状特别敏感；第 118 个过滤器（图 2，右下）对 a_形状非常敏感。这些输入图像中的形状会激活相应位置的特征图（图 2 中的箭头）。 值得注意的是，图 2 中生成的特征图并没有固定输入尺寸。深度卷积曾生成的特征图和传统方法[27][28]中的特征图很相似。这些传统方法中，SIFT 向量[29]或图像碎片[28]被密集地抽取出来，在通过矢量量化[16][15][30]，稀疏化[17][18]或 Fisher 核函数[19]进行编码。这些编码后的特征构成了特征图，然后通过词袋（BoW）[16]或空间金字塔[14][15]进行池化。类似的深度卷积的特征也可以这样做。

2.2 空间金字塔池化层

卷积层接受任意大小的输入，所以他们的输出也是各种大小。而分类器（SVM/softmax）或者全连接层 UI 需要固定的输入大小的向量。这种向量可以使用词袋方法[16]通过池化特征来生成。空间金字塔池化[14][15]对 BoW 进行了改进以便在池化过程中保留局部空间块（local spatial bins）中的空间保留。这些空间块的尺寸和图像的尺寸是成比例的，这样块的数量就是固定的了。而前述深度网络的滑窗池化则对依赖于输入图像的尺寸。

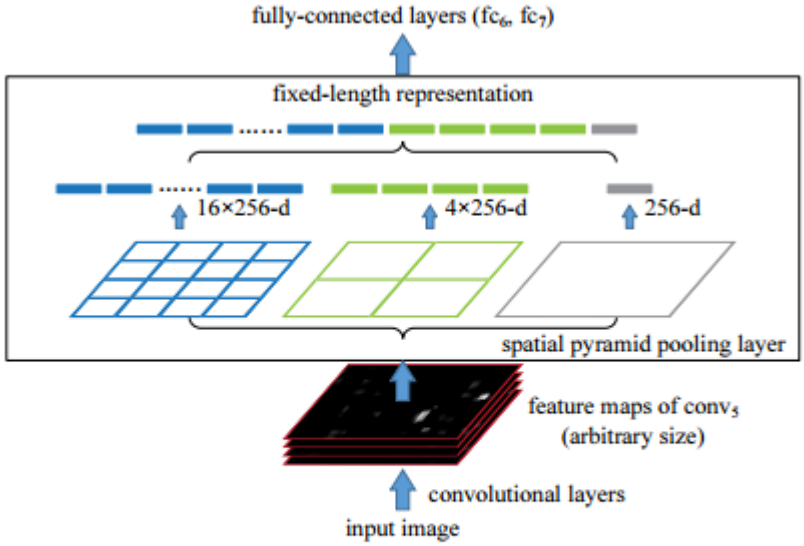

Figure 3: A network structure with a spatial pyramid pooling layer. Here 256 is the filter number of the $conv_5$ layer, and $conv_5$ is the last convolutional layer.

为了让我们的神经网络适应任意尺寸的图像输入，我们用一个空间金字塔池化层替换掉了最优一个池化层（最后一个卷积层之后的 pool5）。图 3 示例了这种方法。在每个空间块中，我们池化每个过滤器的响应（本文中采用了最大池化法）。空间金字塔的输出是一个 kM 维向量，M 代表块的数量，k 代表最后一层卷积层的过滤器的数量。这个固定维度的向量就是全连接层的输入。

有了空间金字塔池化，输入图像就可以是任意尺寸了。不但允许任意比例关系，而且支持任意缩放尺度。我们也可以将输入图像缩放到任意尺度(例如 min(w;h)=180,224,…)并且使用同一个深度网络。当输入图像处于不同的空间尺度时，带有相同大小卷积核的网络就可以在不同的尺度上抽取特征。跨多个尺度在传统方法中十分重要，比如 SIFT 向量就经常在多个尺度上进行抽取[29][27]（受碎片和高斯过滤器的大小所决定）。我们接下来会说明多尺度在深度网络精度方面的重要作用。

有趣的是，粗糙的金字塔级别只有一个块，覆盖了整张图像。这就是一个全局池化操作，当前有很多正在进行的工作正在研究它。[33]中，一个放在全连接层之后的全局平均池化被用

来提高测试阶段的精确度；[34]中，一个全局最大池化用于弱监督物体识别。全局池化操作相当于传统的词袋方法。

2.3 网络的训练

理论上将，上述网络结构可以用标准的反向传播进行训练[1]，与图像的大小无关。但实践中，GPU 的实现（如 cuda-convnet[3]和 Caffe[35]）更适合运行在固定输入图像上。接下来，我们描述我们的训练方法能够在保持空间金字塔池化行为的同时还能充分利用 GPU 的优势。

单一尺寸训练

如前人的工作一样，我们首先考虑接收裁剪成 224×224 图像的网络。裁剪的目的是数据增强。对于一个给定尺寸的图像，我们先计算空间金字塔池化所需要的块（bins）的大小。试想一个尺寸是 axa（也就是 13×13）的 conv5 之后特征图。对于 nxn 块的金字塔级，我们实现一个滑窗池化过程，窗口大小为 win = 上取整[a/n]，步幅 str = 下取整[a/n]. 对于 l 层金字塔，我们实现 l 个这样的层。然后将 l 个层的输出进行连接输出给全连接层。图 4 展示了一个 cuda 卷积网络风格的 3 层金字塔的样例。(3×3, 2×2, 1×1)。

单一尺寸训练的主要目的是开启多级别池化行为。实验表明这是获取精度的一个原因。

```
[pool3x3]          [pool2x2]          [pool1x1]
type=pool          type=pool          type=pool
pool=max           pool=max           pool=max
inputs=conv5       inputs=conv5       inputs=conv5
sizeX=5            sizeX=7            sizeX=13
stride=4           stride=6           stride=13

[fc6]
type=fc
outputs=4096
inputs=pool3x3,pool2x2,pool1x1
```

Figure 4: An example 3-level pyramid pooling in the cuda-convnet style [3]. Here sizeX is the size of the pooling window. This configuration is for a network whose feature map size of $conv_5$ is 13×13, so the $pool_{3\times3}$, $pool_{2\times2}$, and $pool_{1\times1}$ layers will have 3×3, 2×2, and 1×1 bins respectively.

多尺寸训练

携带 SPP 的网络可以应用于任意尺寸，为了解决不同图像尺寸的训练问题，我们考虑一些预设好的尺寸。现在考虑这两个尺寸：180×180,224×224。我们使用缩放而不是裁剪，将前述的 224

的区域图像变成 180 大小。这样，不同尺度的区域仅仅是分辨率上的不同，而不是内容和布局上的不同。对于接受 180 输入的网络，我们实现另一个固定尺寸的网络。本例中，conv5 输出的特征图尺寸是 axa=10×10。我们仍然使用 win = 上取整[a/n]，str = 下取整[a/n]，实现每个金字塔池化层。这个 180 网络的空间金字塔层的输出的大小就和 224 网络的一样了。这样，这个 180 网络就和 224 网络拥有一样的参数了。换句话说，训练过程中，我们通过

使用共享参数的两个固定尺寸的网络实现了不同输入尺寸的 SPP-net。
为了降低从一个网络（比如 224）向另一个网络（比如 180）切换的开销，我们在每个网络上训练一个完整的 epoch，然后在下一个完成的 epoch 再切换到另一个网络（权重保留）。依此往复。实验中我们发现多尺寸训练的收敛速度和单尺寸差不多。
多尺寸训练的主要目的是在保证已经充分利用现在被较好优化的固定尺寸网络实现的同时，模拟不同的输入尺寸。除了上述两个尺度的实现，我们也在每个 epoch 中测试了不同的 sxs 输入，s 是从 180 到 224 之间均匀选取的。后面将在实验部分报告这些测试的结果。
注意，上面的单尺寸或多尺寸解析度只用于训练。在测试阶段，是直接对各种尺寸的图像应用 SPP-net 的。

3 用于图像分类的 SPP-NET

3.1 ImageNet 2012 分类实验

我们在 1000 类别的 Image2012 训练集上训练了网络。我们的训练算法参照了前人的实践工作[3][4][36]。图像会被缩放，以便较小的维度是 256，再从中间获得四个角裁出 224×224。图像会通过水平翻转和颜色变换[3]进行数据增强。
最后两层全连接层会使用 Dropout[3]。learning rate 起始值是 0.01，当错误率停滞后就除以 10。我们的实现基于公开的 cuda-convnet 源代码[3]和 Caffe[35]。所有网络都是在单一 GeForceGTX TitanGPU（6G 内存）耗时二到四周训练的。

3.1.1 基准网络架构

SPP 的优势是和使用的卷积神经网络无关。我研究了四种不同的网络架构[3][4][5]（或他们的修改版），对所有这些架构，SPP 都提升了准确度。基准架构如表 1，简单介绍如下：
– ZF-5：基于 Zeiler 和 Fergus 的“快速”模式[4]的网络架构。数字 5 代表 5 层卷积网络。

model	conv$_1$	conv$_2$	conv$_3$	conv$_4$	conv$_5$	conv$_6$	conv$_7$
ZF-5	96×7^2, str 2 LRN, pool 3^2, str 2 map size 55×55	256×5^2, str 2 LRN, pool 3^2, str 2 27×27	384×3^2 13×13	384×3^2 13×13	256×3^2 13×13	-	-
Convnet*-5	96×11^2, str 4 LRN, map size 55×55	256×5^2 LRN, pool 3^2, str 2 27×27	384×3^2 pool 3^2, 2 13×13	384×3^2 13×13	256×3^2 13×13	-	-
Overfeat-5/7	96×7^2, str 2 pool 3^2, str 3, LRN map size 36×36	256×5^2 pool 2^2, str 2 18×18	512×3^2 18×18	512×3^2 18×18	512×3^2 18×18	512×3^2 18×18	512×3^2 18×18

Table 1: Network architectures: filter number×filter size (*e.g.*, 96×7^2), filter stride (*e.g.*, str 2), pooling window size (*e.g.*, pool 3^2), and the output feature map size (*e.g.*, map size 55×55). LRN represents Local Response Normalization. The padding is adjusted to produce the expected output feature map size.

– Convnet*-5：基于 Krizhevsky 等人工作[3]的修改。我们在 conv2 和 conv3（而不是 conv1 和 conv2）之后加入了两个池化层。这样，每一层之后的特征图就和 ZF-5 的尺寸一样了。
– Overfeat-5/7：基于 Overfeat 论文[5]，使用了[6]的修改。对比 ZF-5/Convnet*-5，这个架构在最后一个池化层产生了更大的特征图（18×18 而不是 13×13）。还在 conv3 和后续的卷基层使用了更多的过滤器（512）。我们也研究了 7 层卷积网络，其中 conv3 和 conv7 结构一样。
基准模型中，最后卷积层之后的池化层会产生 6×6 的特征图，然后跟着两个 4096 维度的全连接层，和一个 1000 路的 softmax 层。这些基准网络的表现参见表 2(a)，我们针对 ZF-5 进行了 70 个 epoch，而其他的用了 90 个 epoch。ZF-5 的表现比[4]中报告的那个要好。增益主要来源于角落裁切来源于整张图片，[36]中也提到了这点。

		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)

Table 2: Error rates in the validation set of ImageNet 2012. All the results are obtained using standard 10-view testing. In the brackets are the gains over the "no SPP" baselines.

3.1.2 多层次池化提升准确度

表 2(b)中我们显示了使用单尺寸训练的结果。训练和测试尺寸都是 224×224.这些网络中，卷积网络都和他们的基准网络有相同的结构，只是最后卷积层之后的池化层，被替换成了 SPP 层。表 2 中的结果我们使用了 4 层金字塔，f6x6, 3×3, 2×2, 1x1g(总共 50 个块)。为了公平比较，我们仍然使用标准的 10-view 预测法，每个 view 都是一个 224×224 的裁切。表 2（b）中的结果显示了明显的性能提升。有趣的是，最大的提升（top-1 error，1.65%）来自于精度最高的网络架构。既然我们一直使用相同 10 个裁切 view。这些提升只能是来自于多层次池化。

值得注意的是多层次池化带来的提升不只是因为更多的参数；而是因为多层次池化对对象的变形和空间布局更加鲁棒[15]。为了说明这个，我们使用一个不同的 4 层金字塔（f4×4, 3×3, 2×2, 1×1g，供 30 个块）训练另一个 ZF-5 网络。这个网络有更少的参数，因为他的全连接层 fc6 有 30×256 维输入而不是 36×256 维。 网络的 top-1/top-5 错误率分别是 35.06/14.04 和 50 块的金字塔网络相近，明显好于非 SPP 基准网络（35.99/14.76）。

3.1.3 多尺寸训练提升准确度

表 2（c）展示了多尺寸训练的结果。训练尺寸是 224 和 180，测试尺寸是 224。我们还使用标准的 10-view 预测法。所有架构的 top-1/top-5 错误率进一步下降。SPP-net(Overfeat-7)的 Top-1 错误率降到 29.68%，比非 SPP 网络低了 2.33%，比单尺寸训练降低了 0.68%。除了使用 180 和 224 两个尺寸，我们还随机选了[180;224]之间多个尺寸。SPP-net(Overfeat-7)的 top1/5 错误率是 30.06%/10.96%。Top-1 错误率比两尺寸版本有所下降，可能因为 224 这个尺寸（测试时用的尺寸）被更少的访问到。但结果仍然比但尺寸版本要好。

之前的 CNN 解决方案[5][36]也处理了不同尺寸问题，但他们主要是基于测试。在 Overfeat[5]和 Howard 的方法[36]中，单一网络在测试解决被应用于不同的尺度，然后将分支平均。Howard 进一步在低/高两个分辨率图像区域上训练了两个不同的网络，然后平均分支。据我们所知，我们是第一个对不同尺寸训练单一网络的方法。

3.1.4 全图像表示提升准确度

接下来我们研究全图像视角的准确度。我们将图像保持比例不变的情况下缩放到 min(w;h)=256。SPP-net 应用到一整张图像上。为了公平比较，我们也计算中央 224×224 裁切这单一视图（上述评估都用过）的准确度。单视图比较的准确度见表 3。验证了 ZF-5/Overfeat-7，top-1 错误率在全视图表示中全部下降。这说明保持完整内容的重要性。即使网络训练时只使用了正方形图像，却也可以很好地适应其他的比例。

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 224×224 of the image.

对比表 2 和表 3 我们发现，结合多种视图大体上要好于全图像视图。然而全视图图像的表示仍然有价值。首先，经验上看，我们发现（下节会讨论）即使结合几十个视图，额外增加两个全图像视角（带翻转）仍然可以提高准确度大约 0.2%。其次，全图像视图从方法论上讲与传统方法[15][17][19]保持了一致，这些方法中对整张图像进行编码的 SIFT 向量被池化在一起。第三，在其他一些应用中，比如图像恢复[37]，相似度评分需要图像表示而不是分类得分。一个全图像的表示就会成为首选。

3.1.5 特征图上的多视图测试

【略】

3.2 Experiments on VOC 2007 Classification

【略】

3.3 Experiments on Caltech101

【略】

4 SPP-NET 用于物体检测

深度网络已经被用于物体检测。我们简要回顾一下最先进的 R-CNN[7]。R-CNN 首先使用选择性搜索[20]从每个图像中选出 2000 个候选窗口。然后将每个窗口中的图像区域变形到固定大小 227×227。一个事先训练好的深度网络被用于抽取每个窗口的特征。然后用二分类的 SVM 分类器在这些特征上针对检测进行训练。R-CNN 产生的引人注目的成果。但 R-CNN 在一张图像的 2000 个窗口上反复应用深度卷积网络，十分耗时。在测试阶段的特征抽取式主要的耗时瓶颈。

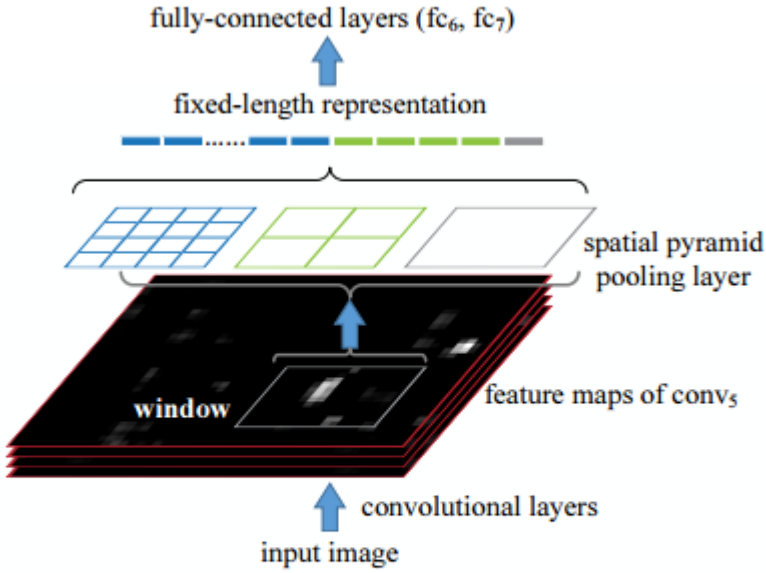

Figure 5: Pooling features from arbitrary windows on feature maps. The feature maps are computed from the entire image. The pooling is performed in candidate windows.

我们将 SPP-net 应用于物体检测。只在整张图像上抽取一次特征。然后在每个特征图的候选窗口上应用空间金字塔池化，形成这个窗口的一个固定长度表示（见图 5）。因为只应用一次卷积网络，我们的方法快得多。我们的方法是从特征图中直接抽取特征，而 R-CNN 则要从图像区域抽取。之前的一些工作中，可变性部件模型(Deformable Part Model, DPM)从 HOG[24]特征图的窗口中抽取图像，选择性搜索方法[20]从 SIFT 编码后的特征图的窗口中抽取特征。Overfeat 也是从卷积特征图中抽取特征，但需要预定义的窗口尺寸。作为对比，我们的特征抽取可以在任意尺寸的深度卷积特征图窗口上。

4.1 检测算法

我们使用选择性搜索[20]的“fast”模式对每张图片产生 2000 个候选窗口。然后缩放图像以满足 min(w;h) = s，并且从整张图像中抽取特征图。我们暂时使用 ZF-5 的 SPP-net 模型（单一尺寸训练）。在每个候选窗口，我们使用一个 4 级空间金字塔（1×1, 2×2, 3×3, 6×6, 总共 50 块）。每个窗口将产生一个 12800（256×50）维的表示。这些表示传递给网络的全连接层。然后我们针对每个分类训练一个二分线性 SVM 分类器。我们的 SVN 实现追随了[20][7]。我们使用真实标注的窗口去生成正例。负例是那些与正例窗口重叠不超过 30%的窗口（使用 IoU 比例）。

如果一个负例与另一个负例重叠超过 70%就会被移除。我们使用标准的难负例挖掘算法（standard hard negative mining [23]）训练 SVM。这个步骤只迭代一次。对于全部 20 个分类训练 SVM 小于 1 个小时。测试阶段，训练器用来对候选窗口打分。然后在打分窗口上使用最大值抑制[23]算法（30%的阈值）。

通过多尺度特征提取，我们的方法可以得到改进。将图像缩放成 min(w;h) = s \belongs S = {480; 576; 688; 864; 1200 }，然后针对每个尺度计算 conv5 的特征图。一个结合这些这些不同尺度特征的策略是逐个 channel 的池化。但我们从经验上发现另一个策略有更好的效果。对于每个候选窗口，我们选择一个单一尺度 s \belongs S，令缩放后的候选窗口的像素数量

接近与 224×224。然后我们从这个尺度抽取的特征图去计算窗口的特征。如果这个预定义的尺度足够密集，窗口近似于正方形。我们的方法粗略地等效于将窗口缩放到 224×224，然后再从中抽取特征。但我们的方法在每个尺度只计算一次特征图，不管有多少个候选窗口。我们参照[7]对预训练的网络进行了调优。由于对于任意尺寸的窗口，我们都是从 conv5 的特征图中吃画出特征来，为了简单起见，我们只调优全连接层。

本例中，数据层接受 conv5 之后的固定长度的池化后的特征，后面跟着 fc_{6,7}和一个新的 21 路（有一个负例类别）fc8 层。fc8 的权重使用高斯分布进行初始化 σ=0.01。我们修正所有的 learning rate 为 1e-4，再将全部三层调整为 1e-5。调优过程中正例是与标注窗口重叠度达到[0.5, 1]的窗口，负例是重叠度为[0.1, 0.5)的。每个 mini-batch，25%是正例。我们使用学习率 1e-4 训练了 250k 个 minibatch，然后使用 1e-5 训练 50k 个 minibatch。

因为我们只调优 fc 层，所以训练非常的快，在 GPU 上只需要 2 个小时，不包括预缓存特征图所需要的 1 小时。另外，遵循[7]，我们使用了约束框回归来后处理预测窗口。用于回归的特征也是 conv5 之后的池化后的特征。用于回归训练的是那些与标注窗口至少重叠 50%的窗口。

4.2 检测结果

我们在 Pascal VOC 2007 数据集的检测任务上，评测了我们的方法。表 9 展示了我们的不同层的结果，使用了 1-scale（s=688）或 5-scale。R-CNN 的结果见[7]，他们使用了 5 个卷积层的 AlexNet[3]。使用 pool5 层我们的结果是 44.9%，R-CNN 的结果是 44.2%。但使用未调优的 fc6 层，我们的结果就不好。可能是我们的 fc 层针对图像区域进行了预训练，在检测案例中，他们用于特征图区域。而特征图区域在窗口框附近会有较强的激活，而图像的区域就不会这样。这种用法的不同是可以通过调优解决的。使用调优后的 fc 层，我们的结果就比 R-CNN 稍胜一筹。经过约束狂回归，我们的 5-scale 结果（59.2%）比 R-CNN（58.5%）高 0.7%。，而 1-scale 结果（58.0%）要差 0.5%。

	SPP (1-sc) (ZF-5)	SPP (5-sc) (ZF-5)	R-CNN (Alex-5)
$pool_5$	43.0	44.9	44.2
fc_6	42.5	44.8	46.2
$ftfc_6$	52.3	53.7	53.1
$ftfc_7$	54.5	55.2	54.2
$ftfc_7$ bb	58.0	**59.2**	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.

表 10 中，我们进一步使用相同预训练的 SPPnet 模型（ZF-5）和 R-CNN 进行比较。本例中，我们的方法和 R-CNN 有相当的平均成绩。R-CNN 的结果是通过预训练模型进行提升的。这是因为 ZF-5 比 AlexNet 有更好的架构，而且 SPPnet 是多层次池化（如果使用非 SPP 的 ZF-5，R-CNN 的结果就会下降）。表 11 表明了每个类别的结果。表也包含了其他方法。

选择性搜索（SS）[20]在 SIFT 特征图上应用空间金字塔匹配。DPM[23]和 Regionlet[39]都是基于 HOG 特征的[24]。Regionlet 方法通过结合包含 conv5 的同步特征可以提升到 46.1%。DetectorNet[40]训练一个深度网络，可以输出像素级的对象遮罩。这个方法仅仅需要对整张图片应用深度网络一次，和我们的方法一样。但他们的方法 mAP 比较低(30.5%)。

	SPP (1-sc) (ZF-5)	SPP (5-sc) (ZF-5)	R-CNN (ZF-5)
$ftfc_7$	54.5	55.2	55.1
$ftfc_7$ bb	58.0	**59.2**	**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).

method	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
DPM [23]	33.7	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5
SS [20]	33.8	43.5	46.5	10.4	12.0	9.3	49.4	53.7	39.4	12.5	36.9	42.2	26.4	47.0	52.4	23.5	12.1	29.9	36.3	42.2	48.8
Regionlet [39]	41.7	54.2	52.0	20.3	24.0	20.1	55.5	68.7	42.6	19.2	44.2	49.1	26.6	57.0	54.5	43.4	16.4	36.6	37.7	59.4	52.3
DetNet [40]	30.5	29.2	35.2	19.4	16.7	3.7	53.2	50.2	27.2	10.2	34.8	30.2	28.2	46.6	41.7	26.2	10.3	32.8	26.8	39.8	47.0
RCNN $ftfc_7$ (A5)	54.2	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7
RCNN $ftfc_7$ (ZF5)	55.1	64.8	68.4	47.0	39.5	30.9	59.8	70.5	65.3	33.5	62.5	50.3	59.5	61.6	67.9	54.1	33.4	57.3	52.9	60.2	62.9
SPP $ftfc_7$ (ZF5)	55.2	65.5	65.9	51.7	38.4	32.7	62.6	68.6	69.7	33.1	66.6	53.1	58.2	63.6	68.8	50.4	27.4	53.7	48.2	61.7	64.7
RCNN bb (A5)	58.5	68.1	72.8	56.8	**43.0**	36.8	**66.3**	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8
RCNN bb (ZF5)	**59.2**	68.4	**74.0**	54.0	40.9	35.2	64.1	**74.4**	69.8	**35.5**	66.9	53.8	**64.2**	69.9	69.6	**58.9**	**36.8**	**63.4**	**56.0**	62.8	64.9
SPP bb (ZF5)	**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**

Table 11: Comparisons of detection results on Pascal VOC 2007.

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 12: Detection results on VOC 2007 using model combination. The results of both models use "$ftfc_7$ bb".

4.3 复杂度和运行时间

【略】

4.4 用于检测的多模型结合

模型结合对于提升 CNN 为基础的分类准确度有重要的提升作用[3]。我们提出一种简单的用于检测的结合方法。

首先在 ImageNet 上预训练另一个网络，使用的结构都相同，只是随机初始化不同。然后我们重复上述的检测算法。表 12（SPP-net（2））显示了这个网络的结果。他的 mAP 可以和第一名的网络相媲美（59.1%vs59.2%），并且在 11 个类别上要好于第一网络。

给定两个模型，我们首先使用每个模型对测试图像的候选框进行打分。然后对并联的两个候选框集合上应用最大化抑制。一个方法比较置信的窗口就会压制另一个方法不太置信的窗口。通过这样的结合，mAP 提升到了 60.9%（表 12）。结合方法在 20 类中的 17 个的表现要好于单个模型。这意味着双模型是互补的。

我们进一步发现这个互补性主要是因为卷积层。我们尝试结合卷积模型完全相同的两个模型，

则没有任何效果。

4.5 ILSVRC 2014 Detection

【略】

5 结论

SPP 对于处理不同的尺度、尺寸和长宽比是十分灵活的解决方案。这些问题在视觉识别时非常重要，但深度网络中大家却很少考虑这个问题。我们建议使用空间金字塔池化层来训练深度网络。这种 SPP-net 在分类和检测任务上都表现出了出色的精度并显著加速了 DNN 为基础的检测任务。我们的研究也表明很多 CV 领域成熟的技术再基于深度网络的识别中仍然可以发挥重要的作用。

文献

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural computation, 1989.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei, “Imagenet: A large-scale hierarchical image database,” in CVPR, 2009.

[3] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in NIPS, 2012.

[4] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional neural networks,” arXiv:1311.2901, 2013.

[5] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” arXiv:1312.6229, 2013.

[6] A. V. K. Chatfield, K. Simonyan and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in ArXiv:1405.3531, 2014.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in CVPR, 2014.

[8] W. Y. Zou, X. Wang, M. Sun, and Y. Lin, “Generic object detection with dense neural patterns and regionlets,” in ArXiv:1404.4316, 2014.

[9] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: An astounding baseline for recogniton,” in CVPR 2014, DeepVision Workshop, 2014.

[10] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in CVPR, 2014.

[11] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdevr, “Panda: Pose aligned networks for deep attribute modeling,” in CVPR, 2014.

[12] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, “Multi-scale orderless pooling of deep convolutional activation features,” in ArXiv:1403.1840, 2014.

[13] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” arXiv:1310.1531, 2013.

[14] K. Grauman and T. Darrell, “The pyramid match kernel: Discriminative classification with sets of image features,” in ICCV, 2005.

[15] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in CVPR, 2006.
[16] J. Sivic and A. Zisserman, “Video google: a text retrieval approach to object matching in videos,” in ICCV, 2003.
[17] J. Yang, K. Yu, Y. Gong, and T. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in CVPR, 2009.
[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Localityconstrained linear coding for image classification,” in CVPR, 2010.
[19] F. Perronnin, J. Sanchez, and T. Mensink, “Improving the fisher ´ kernel for large-scale image classification,” in ECCV, 2010.
[20] K. E. van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, “Segmentation as selective search for object recognition,” in ICCV, 2011.
[21] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories,” CVIU, 2007.
[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,” 2007.
[23] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained partbased models,” PAMI, 2010.
[24] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in CVPR, 2005.
[25] C. L. Zitnick and P. Dollar, “Edge boxes: Locating object ´ proposals from edges,” in ECCV, 2014.
[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., “Imagenet large scale visual recognition challenge,” arXiv:1409.0575, 2014.
[27] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, “The devil is in the details: an evaluation of recent feature encoding methods,” in BMVC, 2011.
[28] A. Coates and A. Ng, “The importance of encoding versus training with sparse coding and vector quantization,” in ICML, 2011.
[29] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” IJCV, 2004.
[30] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders, “Kernel codebooks for scene categorization,” in ECCV, 2008.
[31] M. Lin, Q. Chen, and S. Yan, “Network in network,” arXiv:1312.4400, 2013.
[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” arXiv:1409.4842, 2014.
[33] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv:1409.1556, 2014.
[34] M. Oquab, L. Bottou, I. Laptev, J. Sivic et al., “Learning and transferring mid-level image representations using convolutional neural networks,” in CVPR, 2014.
[35] Y. Jia, “Caffe: An open source convolutional architecture for fast feature embedding,” http://caffe.berkeleyvision.org/, 2013.
[36] A. G. Howard, “Some improvements on deep convolutional neural network based image classification,” ArXiv:1312.5402, 2013.

[37] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid, “Aggregating local image descriptors into compact codes,” TPAMI, vol. 34, no. 9, pp. 1704–1716, 2012.
[38] C.-C. Chang and C.-J. Lin, “Libsvm: a library for support vector machines,” ACM Transactions on Intelligent Systems and Technology (TIST), 2011.
[39] X. Wang, M. Yang, S. Zhu, and Y. Lin, “Regionlets for generic object detection,” in ICCV, 2013.
[40] C. Szegedy, A. Toshev, and D. Erhan, “Deep neural networks for object detection,” in NIPS, 2013.