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

论文引用：

He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(9): 1904-1916.

ABSTRACT

Existing deep convolutional neural networks (CNNs) require a fixed-size (e.g., 224 × 224) input image. This requirement is "artificial" and may reduce the recognition accuracy for the images or sub-images of an arbitrary size/scale. In this work, we equip the networks with another pooling strategy, "spatial pyramid pooling", to eliminate the above requirement. The new network structure, called SPP-net, can generate a fixed-length representation regardless of image size/scale. Pyramid pooling is also robust to object deformations. With these advantages, SPP-net should in general improve all CNN-based image classification methods. On the ImageNet 2012 dataset, we demonstrate that SPP-net boosts the accuracy of a variety of CNN architectures despite their different designs. On the Pascal VOC 2007 and Caltech101 datasets, SPP-net achieves state-of-the-art classification results using a single full-image representation and no fine-tuning. The power of SPP-net is also significant in object detection. Using SPP-net, we compute the feature maps from the entire image only once, and then pool features in arbitrary regions (sub-images) to generate fixed-length representations for training the detectors. This method avoids repeatedly computing the convolutional features. In processing test images, our method is 24-102 × faster than the R-CNN method, while achieving better or comparable accuracy on Pascal VOC 2007. In ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014, our methods rank #2 in object detection and #3 in image classification among all 38 teams. This manuscript also introduces the improvement made for this competition.

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. Introduction

We are witnessing a rapid, revolutionary change in our vision community, mainly caused by deep convolutional neural networks (CNNs) [1] and the availability of large scale training data [2]. Deep-networks-based approaches have recently been substantially improving upon the state of the art in image classification [3], [4], [5], [6], object detection [5], [7], [8], many other recognition tasks [9], [10], [11], [12], and even non-recognition tasks.

1. 引文

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。

However, there is a technical issue in the training and testing of the CNNs: the prevalent CNNs require a fixed input image size (e.g., 224 × 224), which limits both the aspect ratio and the scale of the input image. When applied to images of arbitrary sizes, current methods mostly fit the input image to the fixed size, either via cropping [3], [4] or via warping [7], [13], as shown in Fig. 1 (top). But the cropped region may not contain the entire object, while the warped content may result in unwanted geometric distortion. Recognition accuracy can be compromised due to the content loss or distortion. Besides, a pre-defined scale may not be suitable when object scales vary. Fixing input sizes overlooks the issues involving scales.

然而，CNNs 在训练和测试时都有一个技术问题，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变形，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。

So why do CNNs require a fixed input size? A CNN mainly consists of two parts: convolutional layers, and fully-connected layers that follow. The convolutional layers operate in a sliding-window manner and output feature maps which represent the spatial arrangement of the activations (Fig. 2). In fact, convolutional layers do not require a fixed image size and can generate feature maps of any sizes. On the other hand, the fully-connected layers need to have fixed-size/length input by their definition. Hence, the fixed-size constraint comes only from the fully-connected layers, which exist at a deeper stage of the network.

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列的特征图（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。

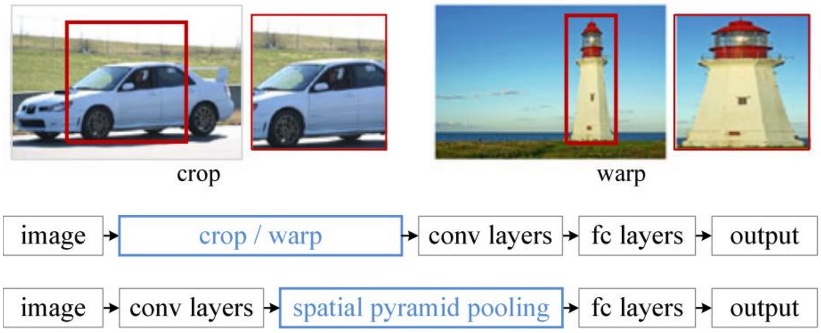


Figure 1. Top: Cropping or warping to fit a fixed size. Middle: A conventional CNN. Bottom: our spatial pyramid pooling network structure.

图 1. 顶部：裁剪或变形以适合固定尺寸。中：传统的 CNN。下图：我们的空间金字塔池化网络结构。

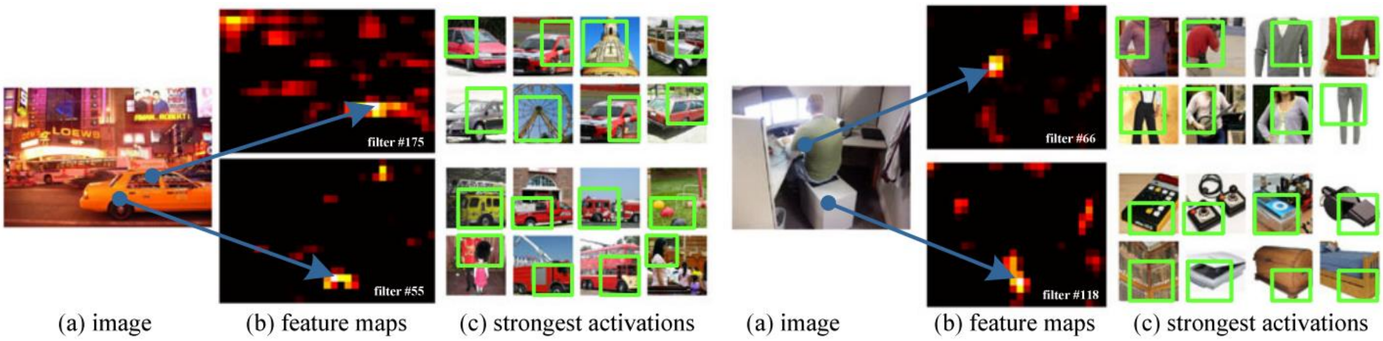


Figure 2. Visualization of the feature maps. (a) Two images in Pascal VOC 2007. (b) The feature maps of some conv5 filters. The arrows indicate the strongest responses

图 2. 可视化特征图。(a)Pascal VOC 2007 中的两张图像。(b)一些 conv5 过滤器的特征图。箭头表示图像中最强的响应及其对应的位置。(c)具有相应滤波器响应

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.

In this paper, we introduce a spatial pyramid pooling (SPP) [14] , [15] layer to remove the fixed-size constraint of the network. Specifically, we add an SPP layer on top of the last convolutional layer. The SPP layer pools the features and generates fixed-length outputs, which are then fed into the fully-connected layers (or other classifiers). In other words, we perform some information "aggregation" at a deeper stage of the network hierarchy (between convolutional layers and fully-connected layers) to avoid the need for cropping or warping at the beginning. Fig. 1 (bottom) shows the change of the network architecture by introducing the SPP layer. We call the new network structure SPP-net.

Spatial pyramid pooling [14], [15] (popularly known as spatial pyramid matching or SPM [15]), as an extension of the Bag-of-Words (BoW) model [16], is one of the most successful methods in computer vision. It partitions the image into divisions from finer to coarser levels, and aggregates local features in them. SPP has long been a key component in the leading and competition-winning systems for classification (e.g., [17], [18], [19]) and detection (e.g., [20]) before the recent prevalence of CNNs. Nevertheless, SPP has not been considered in the context of CNNs. We note that SPP has several remarkable properties for deep CNNs: 1) SPP is able to generate a fixed-length output regardless of the input size, while the sliding window pooling used in the previous deep networks [3] cannot; 2) SPP uses multi-level spatial bins, while the sliding window pooling uses only a single window size. Multi-level pooling has been shown to be robust to object deformations [15] ; 3) SPP can pool features extracted at variable scales thanks to the flexibility of input scales. Through experiments we show that all these factors elevate the recognition accuracy of deep networks.

SPP-net not only makes it possible to generate representations from arbitrarily sized images/windows for testing, but also allows us to feed images with varying sizes or scales during training. Training with variable-size images increases scale-invariance and reduces over-fitting. We develop a simple multi-size training method. For a single network to accept variable input sizes, we approximate it by multiple networks that share all parameters, while each of these networks is trained using a fixed input size. In each epoch we train the network with a given input size, and

最强的 ImageNet 图像。绿色矩形标记最强响应的感受域。

本文引入一种空间金字塔池化(spatial pyramid pooling，SPP) [14]，[15]层以移除对网络固定尺寸的限制。特别地，将 SPP 层放在最后一个卷积层之后。SPP 层对特征图进行池化，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。

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]），是词袋模型(Bag-of-Words, BoW)的扩展，SPP 模型是计算机视觉领域最成功的方法之一。它将图像划分为从更细到更粗的级别，并聚合他们的局部特征。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 使用了多层空间箱(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有十分强的鲁棒性[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，我们将展示所有提升深度网络最终识别精度的因素。

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尺度不变性，以及减少过拟合。我们开发了一个简单的多尺度训练方法。为了实现一个单一网络能够接受各种输入尺寸，我们使用多个共享所有权重（Parameters）的网络来近似得到这种效果，不过，这里的每个网络分别使用固定输入尺寸进行训练。每个 epoch 使用固定的尺寸训练这个网络，下一轮使用另一个尺寸来训练。实验表明，这种多尺度的训练与传统的

switch to another input size for the next epoch. Experiments show that this multi-size training converges just as the traditional single-size training, and leads to better testing accuracy.

The advantages of SPP are orthogonal to the specific CNN designs. In a series of controlled experiments on the ImageNet 2012 dataset, we demonstrate that SPP improves four different CNN architectures in existing publications [3], [4], [5] (or their modifications), over the no-SPP counterparts. These architectures have various filter numbers/sizes, strides, depths, or other designs. It is thus reasonable for us to conjecture that SPP should improve more sophisticated (deeper and larger) convolutional architectures. SPP-net also shows state-of-the-art classification results on Caltech101 [21] and Pascal VOC 2007 [22] using only a single full-image representation and no fine-tuning.

SPP-net also shows great strength in object detection. In the leading object detection method R-CNN [7], the features from candidate windows are extracted via deep convolutional networks. This method shows remarkable detection accuracy on both the VOC and ImageNet datasets. But the feature computation in R-CNN is time-consuming, because it repeatedly applies the deep convolutional networks to the raw pixels of thousands of warped regions per image. In this paper, we show that we can run the convolutional layers only once on the entire image (regardless of the number of windows), and then extract features by SPP-net on the feature maps. This method yields a speedup of over one hundred times over R-CNN. Note that training/running a detector on the feature maps (rather than image regions) is actually a more popular idea [5], [20], [23], [24]. But SPP-net inherits the power of the deep CNN feature maps and also the flexibility of SPP on arbitrary window sizes, which leads to outstanding accuracy and efficiency. In our experiment, the SPP-net-based system (built upon the R-CNN pipeline) computes features 24-102× faster than R-CNN, while has better or comparable accuracy. With the recent fast proposal method of EdgeBoxes [25] , our system takes 0.5 seconds processing an image (including all steps). This makes our method practical for real-world applications.

A preliminary version of this manuscript has been published in ECCV 2014. Based on this work, we attended the competition of ILSVRC 2014 [26], and ranked #2 in object detection and #3 in image classification (both are provided-

单尺度训练收敛速度是一样的，但是带来更好的测试精度。

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。

在目标检测方面，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（不管窗口的数量多少），然后再使用 SPP-net 在特征图上提取特征。这个方法相对于 R-CNN 缩减了上百倍的耗时。在特征图（而不是 region proposal）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 计算卷积特征要快 24-120 倍，而精度却更高。利用 EdgeBoxes [25]最近的快速提案方法，我们的系统处理图像需要 0.5 秒（包括所有步骤）。这使我们的方法适用于实际应用。

我们参加了 ILSVRC 2014 [26]的竞赛，在所有 38 个团队中，在目标检测中排名第二，在图像分类中排名第三（两者都提供了数据专用曲目）。 对于 ILSVRC 2014，[25]做了一些修改。我们表明，SPPnets 可以通过 no-SPP data-only tracks) among all 38 teams. There are a few modifications made for ILSVRC 2014. We show that the SPP-nets can boost various networks that are deeper and larger (Section 3.1.2-3.1.4) over the no-SPP counterparts. Further, driven by our detection framework, we find that multi-view testing on feature maps with flexibly located/sized windows (Section 3.1.5) can increase the classification accuracy. This manuscript also provides the details of these modifications.

We have released the code to facilitate future research (http://research.microsoft.com/en-us/um/people/kahe/).

2. Deep Networks with Spatial Pyramid Pooling

2.1 Convolutional Layers and Feature Maps

Consider the popular seven-layer architectures [3], [4]. The first five layers are convolutional, some of which are followed by pooling layers. These pooling layers can also be considered as “convolutional”, in the sense that they are using sliding windows. The last two layers are fully connected, with an N-way softmax as the output, where N is the number of categories.

The deep network described above needs a fixed image size. However, we notice that the requirement of fixed sizes is only due to the fully-connected layers that demand fixed-length vectors as inputs. On the other hand, the convolutional layers accept inputs of arbitrary sizes. The convolutional layers use sliding filters, and their outputs have roughly the same aspect ratio as the inputs. These outputs are known as feature maps [1]—they involve not only the strength of the responses, but also their spatial positions.

In Fig. 2, we visualize some feature maps. They are generated by some filters of the conv5 layer. Fig. 2c shows the strongest activated images of these filters in the ImageNet dataset. We see a filter can be activated by some semantic content. For example, the 55th filter (Fig. 2, bottom left) is most activated by a circle shape; the 66th filter (Fig. 2, top right) is most activated by a∧-shape; and the 118th filter (Fig. 2, bottom right) is most activated by a ∨-shape. These shapes in the input images (Fig. 2 a) activate the feature maps at the corresponding positions (the arrows in Fig. 2).

It is worth noticing that we generate the feature maps in Fig.

counterparts 来推动更深和更大的各种网络（第 3.1.2-3.1.4 节）。 此外，在我们的检测框架的驱动下，我们发现在具有灵活可变定位/大小的窗口的特征图上进行多视图测试（第 3.1.5 节）可以提高分类精度。

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，其中一些后面跟着池化层。从他们也使用滑窗的角度来看，池化层也可以认为是“卷积层”。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，卷积层接受任意尺寸的输入。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 中生成的特征图并不需要固定的输

2 without fixing the input size. These feature maps generated by deep convolutional layers are analogous to the feature maps in traditional methods [27], [28]. In those methods, SIFT vectors [29] or image patches [28] are densely extracted and then encoded, e.g., by vector quantization [15], [16], [30], sparse coding [17], [18], or Fisher kernels [19]. These encoded features consist of the feature maps, and are then pooled by Bag-of-Words [16] or spatial pyramids [14], [15] . Analogously, the deep convolutional features can be pooled in a similar way.

2.2 The Spatial Pyramid Pooling Layer

The convolutional layers accept arbitrary input sizes, but they produce outputs of variable sizes. The classifiers (SVM/softmax) or fully-connected layers require fixed-length vectors. Such vectors can be generated by the Bag-of-Words approach [16] that pools the features together. Spatial pyramid pooling [14], [15] improves BoW in that it can maintain spatial information by pooling in local spatial bins. These spatial bins have sizes proportional to the image size, so the number of bins is fixed regardless of the image size. This is in contrast to the sliding window pooling of the previous deep networks [3], where the number of sliding windows depends on the input size.

To adopt the deep network for images of arbitrary sizes, we replace the last pooling layer (e.g., pool 5, after the last convolutional layer) with a spatial pyramid pooling layer. Fig. 3 illustrates our method. In each spatial bin, we pool the responses of each filter (throughout this paper we use max pooling). The outputs of the spatial pyramid pooling are kM-dimensional vectors with the number of bins denoted as M (k is the number of filters in the last convolutional layer). The fixed-dimensional vectors are the input to the fully-connected layer.

With spatial pyramid pooling, the input image can be of any sizes. This not only allows arbitrary aspect ratios, but also allows arbitrary scales. We can resize the input image to any scale (e.g., min(w,h) =180, 224, …) and apply the same deep network. When the input image is at different scales, the network (with the same filter sizes) will extract features at different scales. The scales play important roles in traditional methods, e.g., the SIFT vectors are often extracted at multiple scales [27], [29] (determined by the sizes of the patches and Gaussian filters). We will show that the scales are also important for the accuracy of deep networks.

入尺寸。这些由深度卷积层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，使得通过池化可以保留局部空间块（local spatial bins）的信息。这些空间块的尺寸和图像的尺寸是成比例的，所以这样块的数量是固定的了。而前述深度网络的滑窗池化中的滑窗的数量则依赖于输入图像的尺寸。

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后一个池化层（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。

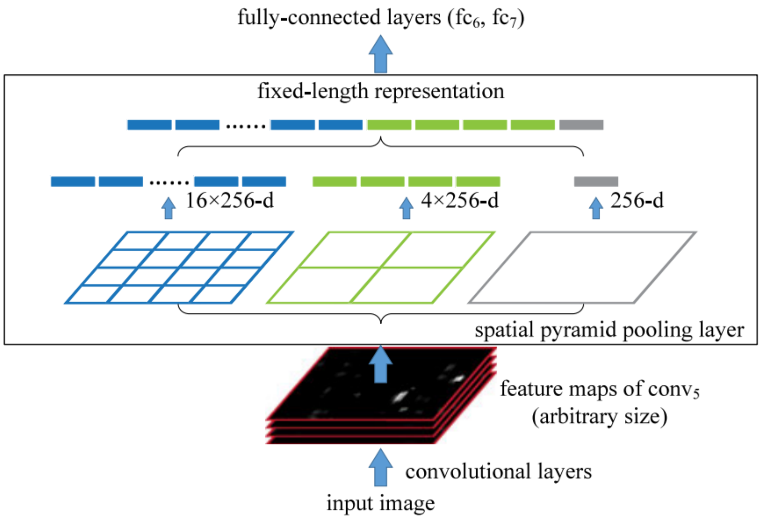

fully-connected layers (fc6, fc7)
fixed-length representation
16×256-d
4×256-d
256-d
spatial pyramid pooling layer
feature maps of conv5
(arbitrary size)
convolutional layers
input image

Figure 3. A network structure with a spatial pyramid pooling layer. Here 256 is the filter number of the conv 5 layer, and conv5 is the last convolutional layer.

Interestingly, the coarsest pyramid level has a single bin that covers the entire image. This is in fact a "global pooling" operation, which is also investigated in several concurrent works. In [31], [32] a global average pooling is used to reduce the model size and also reduce overfitting; in [33], a global average pooling is used on the testing stage after all fc layers to improve accuracy; in [34], a global max pooling is used for weakly supervised object recognition. The global pooling operation corresponds to the traditional Bag-of-Words method.

2.3 Training the Network

Theoretically, the above network structure can be trained with standard back-propagation [1], regardless of the input image size. But in practice the GPU implementations (such as cuda-convnet [3] and Caffe [35]) are preferably run on fixed input images. Next we describe our training solution that takes advantage of these GPU implementations while still preserving the spatial pyramid pooling behaviors.

2.3.1 Single-Size Training

As in previous works, we first consider a network taking a fixed-size input (224 × 224) cropped from images. The cropping is for the purpose of data augmentation. For an image with a given size, we can pre-compute the bin sizes needed for spatial pyramid pooling. Consider the feature maps after conv5 that have a size of a × a (e.g., 13 × 13). With a pyramid level of n × n bins, we implement this pooling level as a sliding window pooling, where the window size win=⌈a/n⌉ and stride str=⌊a/n⌋ with ⌈·⌉ and ⌊·⌋ denoting ceiling and floor operations. With an l-level pyramid, we implement l such layers. The next fully-

图 3. 具有空间金字塔池层的网络结构。这里 256 是 conv 5 层的滤波器通道数，conv5 是最后一个卷积层。

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。

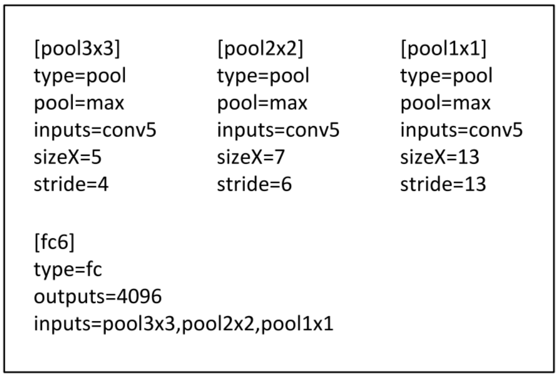
2.3 用空间金字塔池层训练网络

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。

2.3.1 单尺度训练

如前人的工作一样，我们首先考虑接收裁剪成 224×224 图像输入的网络。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。考虑一个尺寸是 a×a（也就是 13×13）的 conv5 之后特征图。对于 n×n 块的金字塔层的块 bins，我们将这个池 level 作为一个滑动窗口池来实现，窗口的大小的 win=⌈a/n⌉和步长 str=⌊a/n⌋，其中⌈·⌉和⌊·⌋分别表示向上和向下取整。用 l 级的金字塔，我们实现 l 个这样的层。下一个完全连接的层（fc6）将连接则这 l 个输出。如下图 4 显示了一个 3 级金字塔池（3×3, 2×2, 1×1）的示例配置。

connected layer (fc6) will concatenate the l outputs. Fig. 4 shows an example configuration of three-level pyramid pooling (3 × 3, 2 × 2, 1 × 1) in the cuda-convnet style [3] .



```
[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 three-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 pool3×3, pool2×2, and pool 1×1 layers will have 3 × 3, 2 × 2, and 1 × 1 bins respectively.

图4. cuda-convnet风格的三级金字塔汇集示例[3]。这里sizeX是池窗口的大小。此配置适用于conv 5的特征映射大小为13×13的网络，因此pool3×3，pool2×2和pool 1×1层将分别具有3×3,2×2和1×1个bin。。

The main purpose of our single-size training is to enable the multi-level pooling behavior. Experiments show that this is one reason for the gain of accuracy.

使用单尺度训练的主要原因是能够使用多级别的池化行为。实验表明，这是获得准确性的一个原因。

2.3.2 Multi-Size Training

2.3.2 多尺度训练

Our network with SPP is expected to be applied on images of any sizes. To address the issue of varying image sizes in training, we consider a set of pre-defined sizes. We consider two sizes: 180 × 180 in addition to 224 × 224. Rather than crop a smaller 180 × 180 region, we resize the aforementioned 224 × 224 region to 180 × 180. So the regions at both scales differ only in resolution but not in content/layout. For the network to accept 180 × 180 inputs, we implement another fixed-size-input (180 × 180) network. The feature map size after conv5 is a × a=10× 10 in this case. Then we still use $win=\lceil a/n \rceil$ and $str=\lfloor a/n \rfloor$ to implement each pyramid pooling level. The output of the spatial pyramid pooling layer of this 180-network has the same fixed length as the 224-network. As such, this 180-network has exactly the same parameters as the 224-network in each layer. In other words, during training we implement the varying-input-size SPP-net by two fixed-size networks that share parameters.

带有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$win=\lceil a/n \rceil$和$str=\lfloor a/n \rfloor$，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。

To reduce the overhead to switch from one network (e.g., 224) to the other (e.g., 180), we train each full epoch on one network, and then switch to the other one (keeping all weights) for the next full epoch. This is iterated. In experiments, we find the convergence rate of this multi-size

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。

training to be similar to the above single-size training.

The main purpose of our multi-size training is to simulate the varying input sizes while still leveraging the existing well-optimized fixed-size implementations. Besides the above two-scale implementation, we have also tested a variant using s×s as input where s is randomly and uniformly sampled from [180,224] at each epoch. We report the results of both variants in the experiment section.

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 s×s 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。

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.

注意，上面的单尺寸或多尺寸方案只用于训练。在测试阶段，是直接对各种尺寸的图像应用 SPP-net 的。

3. SPP-Net for Image Classification

3. SPP-NET 用于图像分类

3.1 Experiments on ImageNet 2012 Classification

3.1 ImageNet 2012 分类实验

We train the networks on the 1,000-category training set of ImageNet 2012. Our training algorithm follows the practices of previous work [3], [4], [36]. The images are resized so that the smaller dimension is 256, and a 224 × 224 crop is picked from the center or the four corners from the entire image. The data are augmented by horizontal flipping and color altering [3]. Dropout [3] is used on the two fully-connected layers. The learning rate starts from 0.01, and is divided by 10 (twice) when the error plateaus. Our implementation is based on the publicly available code of cuda-convnet [3] and Caffe [35]. All networks in this paper can be trained on a single GeForce GTX Titan GPU (6 GB memory) within two to four weeks.

我们在 1000 个类别的 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 Baseline Network Architectures

The advantages of SPP are independent of the convolutional network architectures used. We investigate four different network architectures in existing publications [3], [4], [5] (or their modifications), and we show SPP improves the accuracy of all these architectures. These baseline architectures are in Table 1 and briefly introduced below:

- ZF-5. This architecture is based on Zeiler and Fergus's (ZF) "fast" (smaller) model [4]. The number indicates five convolutional layers.
- Convnet*-5. This is a modification on Krizhevsky et al.'s network [3]. We put the two pooling layers after conv2 and conv 3 (instead of after conv1 and conv 2). As a result, the feature maps after each layer have the same size as ZF-5.

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。我们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。
- 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. This architecture is based on the Overfeat paper[5] , with some modifications as in [6]. In contrast to ZF-5/Convnet*-5, this architecture produces a larger feature map (18×18 instead of 13×13) before the last pooling layer. A larger filter number (512) is used in conv 3 and the following convolutional layers. We also investigate a deeper architecture with seven convolutional layers, where conv3 to conv7 have the same structures.

- 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。

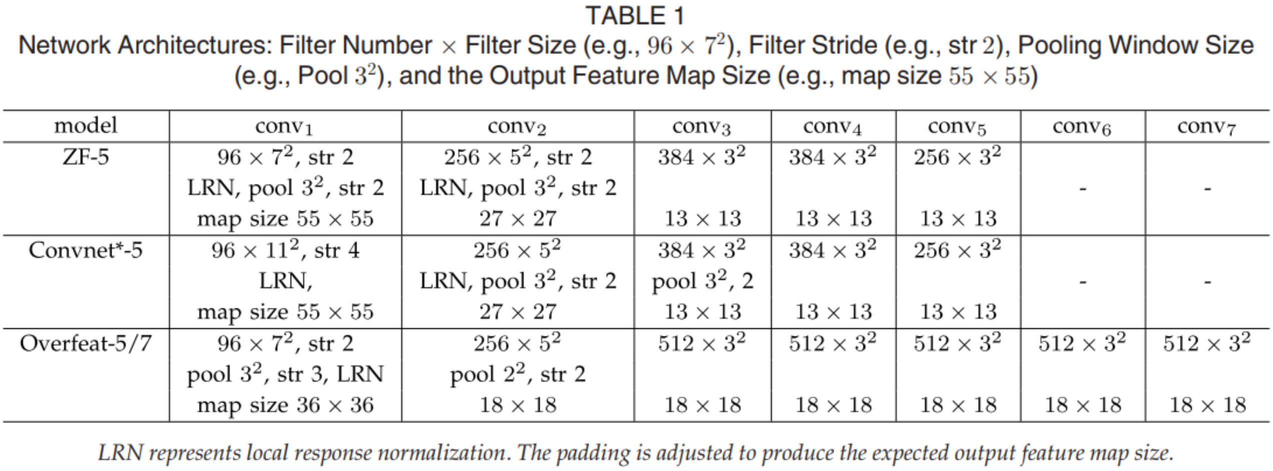


TABLE 1
Network Architectures: Filter Number × Filter Size (e.g., 96×7^2), Filter Stride (e.g., str 2), Pooling Window Size (e.g., Pool 3^2), and the Output Feature Map Size (e.g., map size 55×55)

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

LRN represents local response normalization. The padding is adjusted to produce the expected output feature map size.

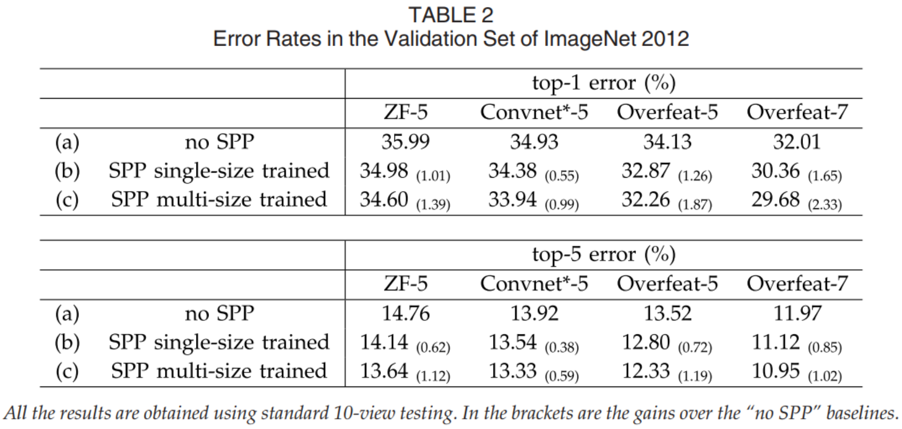


TABLE 2
Error Rates in the Validation Set of ImageNet 2012

		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)

All the results are obtained using standard 10-view testing. In the brackets are the gains over the "no SPP" baselines.

In the baseline models, the pooling layer after the last convolutional layer generates 6×6 feature maps, with two 4096-d fc layers and a 1000-way softmax layer following. Our replications of these baseline networks are in Table 2a. We train 70 epochs for ZF-5 and 90 epochs for the others. Our replication of ZF-5 is better than the one reported in [4] . This gain is because the corner crops are from the entire image, as is also reported in [36].

在基准模型中，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]。

3.1.2 Multi-Level Pooling Improves Accuracy

In Table 2b we show the results using single-size training. The training and testing sizes are both 224 × 224. In these networks, the convolutional layers have the same structures as the corresponding baseline models, whereas the pooling layer after the final convolutional layer is replaced with the SPP layer. For the results in Table 2, we use a four-level pyramid. The pyramid is {6 × 6, 3 × 3, 2 × 2, 1 × 1} (totally

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，{6x6, 3×3, 2×2, 1x1}(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

50 bins). For fair comparison, we still use the standard 10-view prediction with each view a 224 × 224 crop. Our results in Table 2b show considerable improvement over the no-SPP baselines in Table 2a. Interestingly, the largest gain of top-1 error (1.65 percent) is given by the most accurate architecture. Since we are still using the same 10 cropped views as in (a), these gains are solely because of multi-level pooling.

It is worth noticing that the gain of multi-level pooling is not simply due to more parameters; rather, it is because the multi-level pooling is robust to the variance in object deformations and spatial layout [15]. To show this, we train another ZF-5 network with a different 4-level pyramid: {4 × 4, 3 × 3, 2 × 2, 1 × 1} (totally 30 bins). This network has fewer parameters than its no-SPP counterpart, because its fc6 layer has 30 × 256-d inputs instead of 36 × 256-d. The top-1/top-5 errors of this network are 35.06/14.04. This result is similar to the 50-bin pyramid above (34.98/14.14), but considerably better than the no-SPP counterpart (35.99/14.76).

3.1.3 Multi-Size Training Improves Accuracy

Table 2c shows our results using multi-size training. The training sizes are 224 and 180, while the testing size is still 224. We still use the standard 10-view prediction. The top-1/top-5 errors of all architectures further drop. The top-1 error of SPP-net (Overfeat-7) drops to 29.68 percent, which is 2.33 percent better than its no-SPP counterpart and 0.68 percent better than its single-size trained counterpart.

Besides using the two discrete sizes of 180 and 224, we have also evaluated using a random size uniformly sampled from [180,224]. The top-1/5 error of SPP-net (Overfeat-7) is 30.06/10.96 percent. The top-1 error is slightly worse than the two-size version, possibly because the size of 224 (which is used for testing) is visited less. But the results are still better the single-size version.

There are previous CNN solutions [5], [36] that deal with various scales/sizes, but they are mostly based on testing. In Overfeat [5] and Howard’s method [36], the single network is applied at multiple scales in the testing stage, and the scores are averaged. Howard further trains two different networks on low/high-resolution image regions and averages the scores. To our knowledge, our method is the first one that trains a single network with input images of multiple sizes.

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（4×4, 3×3, 2×2, 1×1}，共 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 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。但结果仍然比单尺寸版本要好。

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 Full-Image Representations Improve Accuracy

Next we investigate the accuracy of the full-image views. We resize the image so that min(w,h) = 256 while maintaining its aspect ratio. The SPP-net is applied on this full image to compute the scores of the full view. For fair comparison, we also evaluate the accuracy of the single view in the center 224 × 224 crop (which is used in the above evaluations). The comparisons of single-view testing accuracy are in Table 3. Here we evaluate ZF-5/Overfeat-7. The top-1 error rates are all reduced by the full-view representation. This shows the importance of maintaining the complete content. Even though our network is trained using square images only, it generalizes well to other aspect ratios.

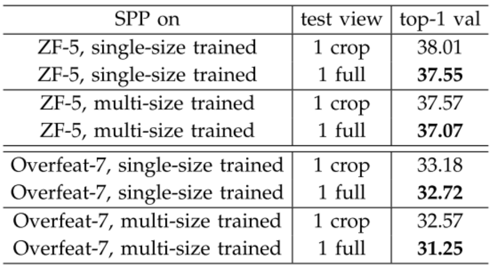
3.1.4 全图像表示提升准确度

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。

Comparing Tables 2 and 3, we find that the combination of multiple views is substantially better than the single full-image view. However, the full-image representations are still of good merits. First, we empirically find that (discussed in the next section) even for the combination of dozens of views, the additional two full-image views (with flipping) can still boost the accuracy by about 0.2 percent. Second, the full-image view is methodologically consistent with the traditional methods [15], [17], [19] where the encoded SIFT vectors of the entire image are pooled together. Third, in other applications such as image retrieval [37], an image representation, rather than a classification score, is required for similarity ranking. A full-image representation can be preferred.

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。

TABLE 3
Error Rates in the Validation Set of ImageNet 2012
Using a Single View



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**

The images are resized so $\min(w,h) = 256$*. The crop view is the central 224 × 224 of the image.*

3.1.5 Multi-View Testing on Feature Maps

Inspired by our detection algorithm (described in the next section), we further propose a multi-view testing method on the feature maps. Thanks to the flexibility of SPP, we can easily extract the features from windows (views) of arbitrary sizes from the convolutional feature maps.

3.1.5 在特征图上的多视图测试

受我们的检测算法的启发（在下一节中介绍），我们进一步提出了在特征映射上的多视图测试方法。感谢 SPP 的灵活性，我们可以从卷积特征映射中轻松地从任意大小的窗口（视图）中提取特征。

On the testing stage, we resize an image so min(w,h)=s where s represents a predefined scale (like 256). Then we

在测试阶段，我们调整图像的大小，使 min(w,h)=s，其中 s 代表预定义比例（如 256）。然后我们计算整个图

compute the convolutional feature maps from the entire image. For the usage of flipped views, we also compute the feature maps of the flipped image. Given any view (window) in the image, we map this window to the feature maps (the way of mapping is in Appendix), and then use SPP to pool the features from this window (see Fig. 5). The pooled features are then fed into the fc layers to compute the softmax score of this window. These scores are averaged for the final prediction. For the standard 10-view, we use s=256 and the views are 224 ×224 windows on the corners or center. Experiments show that the top-5 error of the 10-view prediction on feature maps is within 0.1 percent around the original 10-view prediction on image crops.

像的卷积特征图。对于翻转视图的使用，我们还计算翻转图像的特征映射。给定图像中的任何视图（窗口），我们将这个窗口映射到特征映射（映射的方式在附录中），然后使用 SPP 从这个窗口汇集特征（参见图 5）。汇集的特征然后被馈送到 fc 层以计算该窗口的 softmax 分数。这些分数是最终预测的平均值。对于标准的 10 视图，我们使用 s = 256，并且角落或中心的视图是 224×224 窗口。实验表明，在剪切图像的原始 10 视图预测周围，特征图上 10 视图预测的前 5 个误差在 0.1％以内。

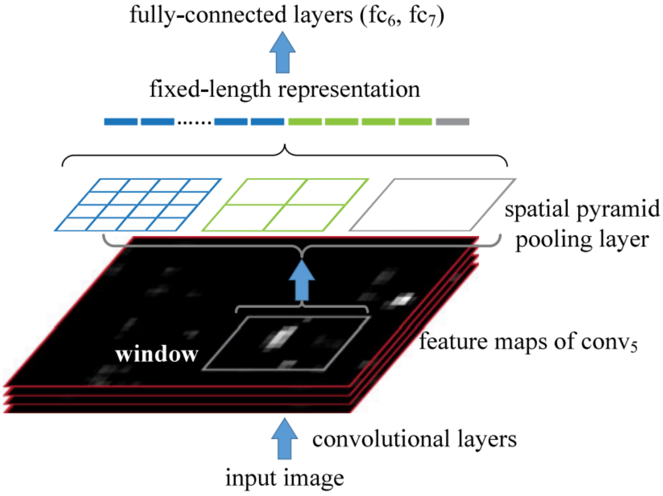

fully-connected layers (fc_6, fc_7)
fixed-length representation
spatial pyramid pooling layer
window
feature maps of $conv_5$
convolutional layers
input image

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.

图5. 在特征图上的任意窗口中汇集特征。特征图从整个图像计算出。池化在候选窗口中执行。

We further apply this method to extract multiple views from multiple scales. We resize the image to six scales s ∈ {224,256,300,360,448,560} and compute the feature maps on the entire image for each scale. We use 224×224 as the view size for any scale, so these views have different relative sizes on the original image for different scales. We use 18 views for each scale: one at the center, four at the corners, and four on the middle of each side, with/without flipping (when s = 224 there are six different views). The combination of these 96 views reduces the top-5 error from 10.95 to 9.36 percent. Combining the two full-image views (with flipping) further reduces the top-5 error to 9.14 percent.

我们进一步应用此方法从多个尺度提取多个视图。我们将图像的大小调整为六个尺度 s ∈{224,256,300,360,448,560}，并为每个尺度计算整个图像的特征图。我们使用 224×224 作为任何比例的视图大小，因此对于不同比例，这些视图在原始图像上具有不同的相对大小。我们对每个比例使用 18 个视图：一个在中心，四个在角落，四个在每一侧的中间，有/没有翻转（当 s = 224 时，有六个不同的视图）。这 96 个视图的组合将前 5 个错误从 10.95 减少到 9.36％。组合两个完整图像视图（翻转）进一步将前 5 个错误减少到 9.14％。

In the Overfeat paper [5], the views are also extracted from the convolutional feature maps instead of image crops. However, their views cannot have arbitrary sizes; rather, the windows are those where the pooled features match the desired dimensionality. We empirically find that these restricted windows are less beneficial than our flexibly

在 Overfeat 论文[5]中，视图也是从卷积特征图而不是图像裁剪中提取的。但是，他们的观点不能有任意大小;相反，窗口是汇集的特征与所需维度相匹配的窗口。我们凭经验发现这些受限制的窗户比我们灵活定位/大小的窗户更不利。

located/sized windows.

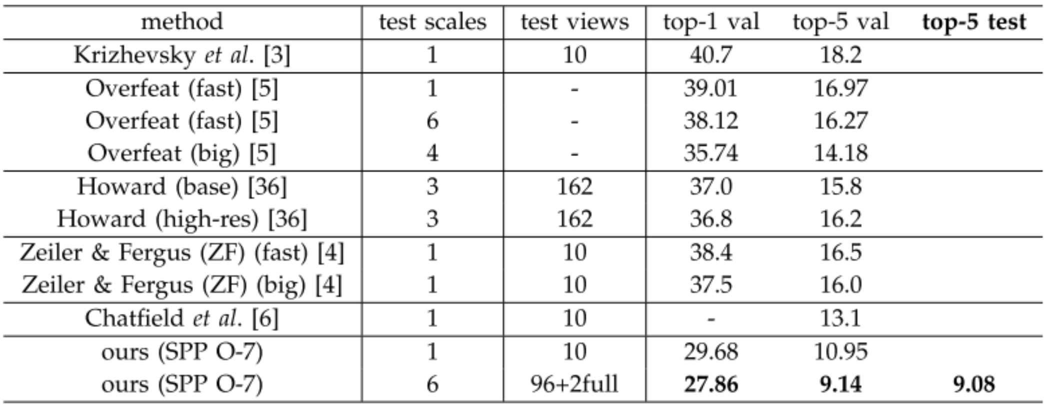
3.1.6 Summary and Results for ILSVRC 2014

In Table 4 we compare with previous state-of-the-art methods. Krizhevsky et al.'s [3] is the winning method in ILSVRC 2012; Overfeat [5], Howard's [36], and Zeiler and Fergus's [4] are the leading methods in ILSVRC 2013. We only consider single-network performance for manageable comparisons.

3.1.6 2014 年 ILSVRC 的总结和结果

在表 4 中，我们与先前的最新方法进行了比较。Krizhevsky 等人[3]是 ILSVRC 2012 中的获胜方法; Overfeat [5]，Howard's [36]和 Zeiler 和 Fergus [4]是 ILSVRC 2013 中的主要方法。我们只考虑单网络性能进行可管理的比较。

TABLE 4
Error Rates in ImageNet 2012



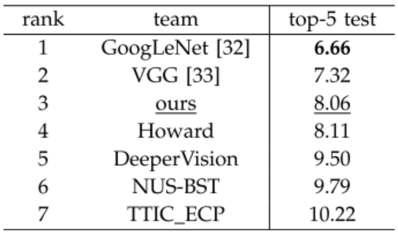
method	test scales	test views	top-1 val	top-5 val	**top-5 test**
Krizhevsky *et al.* [3]	1	10	40.7	18.2	
Overfeat (fast) [5]	1	-	39.01	16.97	
Overfeat (fast) [5]	6	-	38.12	16.27	
Overfeat (big) [5]	4	-	35.74	14.18	
Howard (base) [36]	3	162	37.0	15.8	
Howard (high-res) [36]	3	162	36.8	16.2	
Zeiler & Fergus (ZF) (fast) [4]	1	10	38.4	16.5	
Zeiler & Fergus (ZF) (big) [4]	1	10	37.5	16.0	
Chatfield *et al.* [6]	1	10	-	13.1	
ours (SPP O-7)	1	10	29.68	10.95	
ours (SPP O-7)	6	96+2full	**27.86**	**9.14**	**9.08**

*All the results are based on **a single network**. The number of views in Overfeat depends on the scales and strides, for which there are several hundreds at the finest scale.*

Our best single network achieves 9.14 percent top-5 error on the validation set. This is exactly the single-model entry we submitted to ILSVRC 2014 [26]. The top-5 error is 9.08 percent on the testing set (ILSVRC 2014 has the same training/validation/testing data as ILSVRC 2012). After combining eleven models, our team's result (8.06 percent) is ranked #3 among all 38 teams attending ILSVRC 2014 (Table 5). Since the advantages of SPP-net should be in general independent of architectures, we expect that it will further improve the deeper and larger convolutional architectures [32], [33] .

我们最好的单一网络在验证集上实现了 9.14％的 top-5 错误。这正是我们提交给 ILSVRC 2014 [26]的单一模型条目。测试集中的前 5 个错误为 9.08％（ILSVRC 2014 具有与 ILSVRC 2012 相同的培训/验证/测试数据）。在结合 11 个模型之后，我们团队的结果（8.06％）在参加 ILSVRC 2014 的所有 38 个团队中排名第 3（表 5）。由于 SPP-net 的优点通常应该独立于体系结构，我们期望它将进一步改进更深和更大的卷积体系结构[32]，[33]。

TABLE 5
The Competition Results of ILSVRC 2014 Classification [26]



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

The best entry of each team is listed.

3.2 Experiments on VOC 2007 Classification

Our method can generate a full-view image representation. With the above networks pre-trained on ImageNet, we extract these representations from the images in the target datasets and re-train SVM classifiers [38]. In the SVM training, we intentionally do not use any data augmentation (flip/multi-view). We l2-normalize the features for SVM training.

3.2 VOC 2007 分类任务的实验

我们的方法可以生成全视图图像表示。通过在 ImageNet 上预训练的上述网络，我们从目标数据集中的图像中提取这些表示并重新训练 SVM 分类器[38]。在 SVM 训练中，我们故意不使用任何数据扩充（翻转/多视图）。我们将 SVM 训练的特征标准化。

The classification task in Pascal VOC 2007 [22] involves 9,963 images in 20 categories. 5,011 images are for training, and the rest are for testing. The performance is evaluated by mean Average Precision (mAP). Table 6 summarizes the results.

Pascal VOC 2007 [22]中的分类任务涉及 20 个类别中的 9,963 个图像。5,011 张图片用于训练，其余图片用于测试。性能通过平均精度（mAP）评估。表 6 总结了结果。

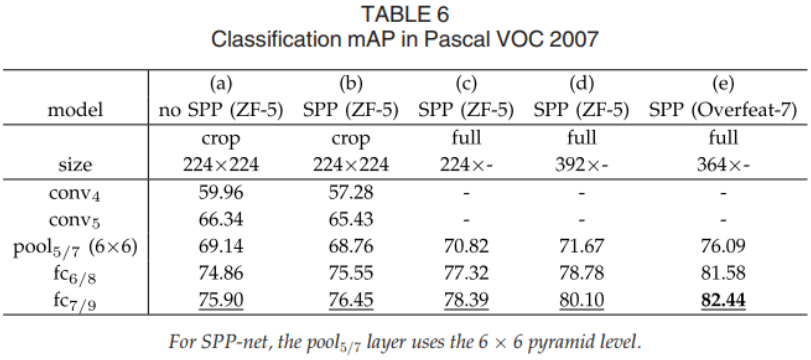


TABLE 6
Classification mAP in Pascal VOC 2007

model	(a) no SPP (ZF-5)	(b) SPP (ZF-5)	(c) SPP (ZF-5)	(d) SPP (ZF-5)	(e) SPP (Overfeat-7)
size	crop 224×224	crop 224×224	full 224×-	full 392×-	full 364×-
conv_4	59.96	57.28	-	-	-
conv_5	66.34	65.43	-	-	-
$\text{pool}_{5/7}$ (6×6)	69.14	68.76	70.82	71.67	76.09
$\text{fc}_{6/8}$	74.86	75.55	77.32	78.78	81.58
$\text{fc}_{7/9}$	75.90	76.45	78.39	80.10	**82.44**

For SPP-net, the $\text{pool}_{5/7}$ layer uses the 6 × 6 pyramid level.

We start from a baseline in Table 6a. The model is ZF-5 without SPP. To apply this model, we resize the image so that its smaller dimension is 224, and crop the center 224 × 224 region. The SVM is trained via the features of a layer. On this dataset, the deeper the layer is, the better the result is. In Table 6b, we replace the no-SPP net with our SPP-net. As a first-step comparison, we still apply the SPP-net on the center 224 ×224 crop. The results of the fc layers improve. This gain is mainly due to multi-level pooling.

我们从表 6a 中的基线开始。该模型为 ZF-5，不含 SPP。要应用此模型，我们调整图像大小以使其较小的尺寸为 224，并裁剪中心 224×224 区域。SVM 通过图层的功能进行训练。在此数据集上，图层越深，结果越好。在表 6b 中，我们用 SPP-net 替换了无 SPP 网。作为第一步比较，我们仍然将 SPP-net 应用于中心 224×224 作物。fc 层的结果得到改善。此增益主要归因于多级池化。

Table 6c shows our results on full images, where the images are resized so that the shorter side is 224. We find that the results are considerably improved (78.39 versus 76.45 percent). This is due to the full-image representation that maintains the complete content.

表 6c 显示了我们在完整图像上的结果，其中图像被调整大小以使较短边为 224.我们发现结果得到显着改善（78.39 对 76.45％）。

Because the usage of our network does not depend on scale, we resize the images so that the smaller dimension is s and use the same network to extract features. We find that s=392 gives the best results (Table 6d) based on the validation set. This is mainly because the objects occupy smaller regions in VOC 2007 but larger regions in ImageNet, so the relative object scales are different between the two sets. These results indicate scale matters in the classification tasks, and SPP-net can partially address this “scale mismatch” issue.

这是由于维护完整内容的全图像表示。由于我们网络的使用并不依赖于规模，因此我们调整图像大小以使较小的维度为 s 并使用相同的网络来提取特征。我们发现 s = 392 基于验证集给出了最好的结果（表 6d）。这主要是因为物体在 VOC 2007 中占据较小的区域，而在 ImageNet 中占据较大的区域，因此两组之间的相对物体尺度不同。这些结果表明分类任务中的比例问题，SPP-net 可以部分解决这种“规模不匹配”问题。

In Table 6e the network architecture is replaced with our best model (Overfeat-7, multi-size trained), and the mAP increases to 82.44 percent. Table 8 summarizes our results and the comparisons with the state-of-the-art methods. Among these methods, VQ [15], LCC [18], and FK [19] are all based on spatial pyramids matching, and [4], [6] , [13], [34] are based on deep networks. In these results, Oquab et al.’s (77.7 percent) and Chatfield et al.’s (82.42 percent) are

在表 6e 中，网络架构被替换为我们的最佳模型（Overfeat-7，多尺寸训练），并且 mAP 增加到 82.44％。表 8 总结了我们的结果以及与最先进方法的比较。在这些方法中，VQ [15]，LCC [18]和 FK [19]均基于空间金字塔匹配，[4]，[6]，[13]，[34]基于深度网络。在这些结果中，Oquab 等人（77.7％）和 Chatfield 等人（82.42％）是通过网络微调和多视图测试获得的。我们的结果与现有技术相当，仅使用单个全图像表示而无需微调。

obtained by network fine-tuning and multi-view testing. Our result is comparable with the state of the art, using only a single full-image representation and without fine-tuning.

3.3 Experiments on Caltech101

3.3 Caltech101 上的实验

The Caltech101 dataset [21] contains 9,144 images in 102 categories (one background). We randomly sample 30 images per category for training and up to 50 images per category for testing. We repeat 10 random splits and average the accuracy. Table 7 summarizes our results.

Caltech101 数据集[21]包含 102 个类别（一个背景）中的 9,144 个图像。我们为每个类别随机抽样 30 张图片进行培训，每个类别最多 50 张图片进行测试。我们重复 10 次随机分割并平均准确度。表 7 总结了我们的结果。

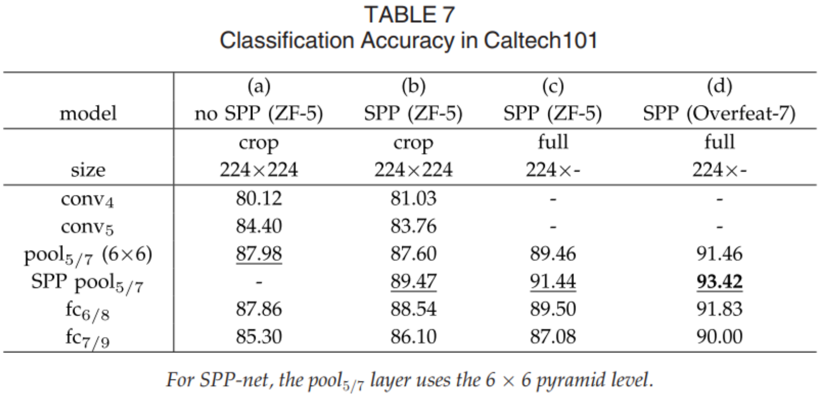


TABLE 7
Classification Accuracy in Caltech101

	(a)	(b)	(c)	(d)
model	no SPP (ZF-5)	SPP (ZF-5)	SPP (ZF-5)	SPP (Overfeat-7)
	crop	crop	full	full
size	224×224	224×224	224×-	224×-
conv$_4$	80.12	81.03	-	-
conv$_5$	84.40	83.76	-	-
pool$_{5/7}$ (6×6)	87.98	87.60	89.46	91.46
SPP pool$_{5/7}$	-	89.47	91.44	**93.42**
fc$_{6/8}$	87.86	88.54	89.50	91.83
fc$_{7/9}$	85.30	86.10	87.08	90.00

For SPP-net, the pool$_{5/7}$ layer uses the 6 × 6 pyramid level.

There are some common observations in the Pascal VOC 2007 and Caltech101 results: SPP-net is better than the no-SPP net (Table 7b versus Table 7 a), and the full-view representation is better than the crop (Table 7 c versus Table 7b). But the results in Caltech101 have some differences with Pascal VOC. The fully-connected layers are less accurate, and the SPP layers are better. This is possibly because the object categories in Caltech101 are less related to those in ImageNet, and the deeper layers are more category-specialized. Further, we find that the scale 224 has the best performance among the scales we tested on this dataset. This is mainly because the objects in Caltech101 also occupy large regions of the images, as is the case of ImageNet.

在 Pascal VOC 2007 和 Caltech101 结果中有一些共同的观察结果：SPP-net 优于无 SPP 网（表 7b 与表 7a），全视图表示优于作物（表 7c 对比）表 7b）。但 Caltech101 的结果与 Pascal VOC 有一些差异。完全连接的层不太准确，SPP 层更好。这可能是因为 Caltech101 中的对象类别与 ImageNet 中的对象类别关联性较低，而较深层的类别更加类别化。此外，我们发现，在我们在该数据集上测试的尺度中，尺度 224 具有最佳性能。这主要是因为 Caltech101 中的对象也占据了图像的大部分区域，就像 ImageNet 的情况一样。

Besides cropping, we also evaluate warping the image to fit the 224×224 size. This solution maintains the complete content, but introduces distortion. On the SPP (ZF-5) model, the accuracy is 89.91 percent using the SPP layer as features—lower than 91.44 percent which uses the same model on the undistorted full image.

除了裁剪，我们还评估变形图像以适应 224×224 大小。此解决方案保留了完整的内容，但引入了失真。在 SPP(ZF-5)模型中，使用 SPP 层作为特征的准确度为 89.91％ - 低于 91.44％，在未失真的完整图像上使用相同的模型。

Table 8 summarizes our results compared with the state-of-the-art methods on Caltech101. Our result (93.42 percent) exceeds the previous record (88.54 percent) by a substantial margin (4.88 percent).

表 8 总结了我们与 Caltech101 上最先进的方法相比的结果。我们的结果(93.42％)大幅超出之前的记录(88.54％)，超出(4.88％)。

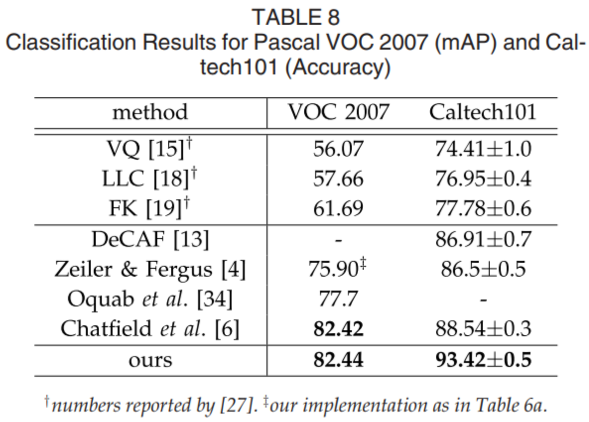


TABLE 8
Classification Results for Pascal VOC 2007 (mAP) and Caltech101 (Accuracy)

method	VOC 2007	Caltech101
VQ [15]†	56.07	74.41±1.0
LLC [18]†	57.66	76.95±0.4
FK [19]†	61.69	77.78±0.6
DeCAF [13]	-	86.91±0.7
Zeiler & Fergus [4]	75.90‡	86.5±0.5
Oquab *et al.* [34]	77.7	-
Chatfield *et al.* [6]	**82.42**	88.54±0.3
ours	**82.44**	**93.42±0.5**

†*numbers reported by [27].* ‡*our implementation as in Table 6a.*

4. SPP-Net for Object Detection

Deep networks have been used for object detection. We briefly review the recent state-of-the-art R-CNN method [7]. R-CNN first extracts about 2,000 candidate windows from each image via selective search (SS) [20]. Then the image region in each window is warped to a fixed size (227 × 227). A pre-trained deep network is used to extract the feature of each window. A binary SVM classifier is then trained on these features for detection. R-CNN generates results of compelling quality and substantially outperforms previous methods. However, because R-CNN repeatedly applies the deep convolutional network to about 2,000 windows per image, it is time-consuming. Feature extraction is the major timing bottleneck in testing.

Our SPP-net can also be used for object detection. We extract the feature maps from the entire image only once (possibly at multiple scales). Then we apply the spatial pyramid pooling on each candidate window of the feature maps to pool a fixed-length representation of this window (see Fig. 5). Because the time-consuming convolutions are only applied once, our method can run orders of magnitude faster.

Our method extracts window-wise features from regions of the feature maps, while R-CNN extracts directly from image regions. In previous works, the Deformable Part Model (DPM) [23] extracts features from windows in HOG [24] feature maps, and the Selective Search method [20] extracts from windows in encoded SIFT feature maps. The Overfeat detection method [5] also extracts from windows of deep convolutional feature maps, but needs to pre-define the window size. On the contrary, our method enables feature extraction in arbitrary windows from the deep convolutional feature maps.

4.1 Detection Algorithm

4. SPP-NET 用于目标检测

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提取每个窗口的特征。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。在测试阶段的特征提取是主要的耗时瓶颈。

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 region proposal 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

We use the “fast” mode of selective search [20] to generate about 2,000 candidate windows per image. Then we resize the image such that min(w,h)=s , and extract the feature maps from the entire image. We use the SPP-net model of ZF-5 (single-size trained) for the time being. In each candidate window, we use a four-level spatial pyramid (1 × 1, 2 × 2, 3 × 3, 6 × 6, totally 50 bins) to pool the features. This generates a 12,800-d (256 × 50) representation for each window. These representations are provided to the fully-connected layers of the network. Then we train a binary linear SVM classifier for each category on these features.

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。

Our implementation of the SVM training follows [7], [20]. We use the ground-truth windows to generate the positive samples. The negative samples are those overlapping a positive window by at most 30 percent (measured by the intersection-over-union (IoU) ratio). Any negative sample is removed if it overlaps another negative sample by more than 70 percent. We apply the standard hard negative mining [23] to train the SVM. This step is iterated once. It takes less than 1 hour to train SVMs for all 20 categories. In testing, the classifier is used to score the candidate windows. Then we use non-maximum suppression [23] (threshold of 30 percent) on the scored windows.

我们的 SVM 实现追随了[7], [20]。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 小时。测试阶段，分类器用来对候选窗口打分。然后在打分窗口上使用非极大值抑制[23]算法（30%的阈值）。

Our method can be improved by multi-scale feature extraction. We resize the image such that min(w,h)=s∈S={480,576,688,864,1,200}, and compute the feature maps of conv5 for each scale. One strategy of combining the features from these scales is to pool them channel-by-channel. But we empirically find that another strategy provides better results. For each candidate window, we choose a single scale s∈S such that the scaled candidate window has a number of pixels closest to 224 × 224. Then we only use the feature maps extracted from this scale to compute the feature of this window. If the pre-defined scales are dense enough and the window is approximately square, our method is roughly equivalent to resizing the window to 224 × 224 and then extracting features from it. Nevertheless, our method only requires computing the feature maps once (at each scale) from the entire image, regardless of the number of candidate windows.

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∈S={480,576,688,864,1,200}，然后针对每个尺度计算 conv5 的特征图。一个结合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∈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。

We also fine-tune our pre-trained network, following [7]. Since our features are pooled from the conv5 feature maps from windows of any sizes, for simplicity we only fine-tune the fully-connected layers. In this case, the data layer accepts the fixed-length pooled features after conv5, and the fc6,7 layers and a new 21-way (one extra negative category)

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 的特征图中池化来得到特征的，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

fc8 layer follow. The fc 8 weights are initialized with a Gaussian distribution of σ = 0.01. We fix all the learning rates to 1e-4 and then adjust to 1e-5 for all three layers. During fine-tuning, the positive samples are those overlapping with a ground-truth window by [0.5,1] , and the negative samples by [0.1,0.5). In each mini-batch, 25 percent of the samples are positive. We train 250k mini-batches using the learning rate 1e-4, and then 50 k mini-batches using 1e-5. Because we only fine-tune the fc layers, the training is very fast and takes about 2 hours on the GPU (excluding pre-caching feature maps which takes about 1 hour). Also following [7] , we use bounding box regression to post-process the prediction windows. The features used for regression are the pooled features from conv5 (as a counterpart of the pool 5 features used in [7]). The windows used for the regression training are those overlapping with a ground-truth window by at least 50 percent.

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。

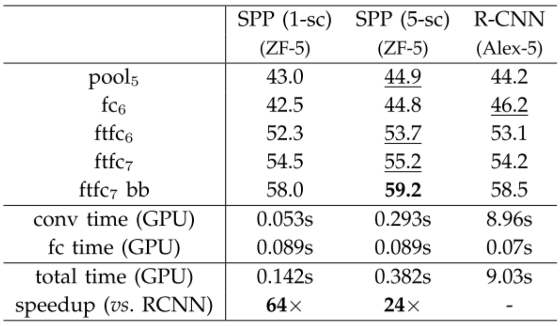
4.2 Detection Results

We evaluate our method on the detection task of the Pascal VOC 2007 dataset. Table 9 shows our results on various layers, by using one-scale (s = 688) or five-scale. Here the R-CNN results are as reported in [7] using the AlexNet [3] with five conv layers. Using the pool5 layers (in our case the pooled features), our result (44.9 percent) is comparable with R-CNN’s result (44.2 percent). But using the non-fine-tuned fc6 layers, our results are inferior. An explanation is that our fc layers are pre-trained using image regions, while in the detection case they are used on the feature map regions. The feature map regions can have strong activations near the window boundaries, while the image regions may not. This difference of usages can be addressed by fine-tuning. Using the fine-tuned fc layers (ftfc 6,7), our results are comparable with or slightly better than the fine-tuned results of R-CNN. After bounding box regression, our five-scale result (59.2 percent) is 0.7 percent better than R-CNN (58.5 percent), and our one-scale result (58.0 percent) is 0.5 percent worse.

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。经过约束框回归，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%。

TABLE 9
Detection Results (mAP) on Pascal VOC 2007



	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**×	-

“ft” and “bb” denote fine-tuning and bounding box regression.

In Table 10 we further compare with R-CNN using the same pre-trained model of SPPnet (ZF-5). In this case, our method and R-CNN have comparable averaged scores. The R-CNN result is boosted by this pre-trained model. This is because of the better architecture of ZF-5 than AlexNet, and also because of the multi-level pooling of SPPnet (if using the no-SPP ZF-5, the R-CNN result drops). Table 11 shows the results for each category.

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）。

TABLE 10
Detection Results (mAP) on Pascal VOC 2007, **Using the Same Pre-Trained Model** of SPP (ZF-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 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
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 also includes additional methods. Selective Search [20] applies spatial pyramid matching on SIFT feature maps. DPM [23] and Regionlet [39] are based on HOG features [24]. The Regionlet method improves to 46.1 percent [8] by combining various features including conv5 . DetectorNet [40] trains a deep network that outputs pixel-wise object masks. This method only needs to apply the deep network once to the entire image, as is the case for our method. But this method has lower mAP (30.5 percent).

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%)。

4.3 Complexity and Running Time

Despite having comparable accuracy, our method is much faster than R-CNN. The complexity of the convolutional feature computation in R-CNN is O(n·2272) with the window number n (~2,000). This complexity of our method is O(r·s2) at a scale s, where r is the aspect ratio. Assume r is about 4/3. In the single-scale version when s=688 , this complexity is about 1/160 of R-CNN’s; in the five-scale version, this complexity is about 1/24 of R-CNN’s.

In Table 10, we provide a fair comparison on the running

4.3 复杂度和运行时间

尽管具有可比较的准确度，但我们的方法比 R-CNN 快得多。R-CNN 中卷积特征计算的复杂度为 O（n·2272），窗口数为 n（~2,000）。我们方法的这种复杂性是在尺度 s 下的 O（r·s2），其中 r 是纵横比。假设 r 约为 4/3。在 s = 688 的单尺度版本中，这种复杂性约为 R-CNN 的 1/160；在五级版本中，这种复杂性约为 R-CNN 的 1/24。

在表 10 中，我们使用相同的 SPP（ZF-5）模型对特征计

time of the feature computation using the same SPP (ZF-5) model. The implementation of R-CNN is from the code published by the authors implemented in Caffe [35]. We also implement our feature computation in Caffe. In Table 10 we evaluate the average time of 100 random VOC images using GPU. R-CNN takes 14.37s per image for convolutions, while our one-scale version takes only 0.053 s per image. So ours is 270× faster than R-CNN. Our five-scale version takes 0.293 s per image for convolutions, so is 49 × faster than R-CNN. Our convolutional feature computation is so fast that the computational time of fc layers takes a considerable portion. Table 10 shows that the GPU time of computing the 4,096-d fc7 features is 0.089 s per image. Considering both convolutional and fully-connected features, our one-scale version is 102× faster than R-CNN and is 1.2 percent inferior; our five-scale version is 38 × faster and has comparable results.

We also compares the running time in Table 9 where R-CNN uses AlexNet [3] as is in the original paper [7] . Our method is 24× to 64 × faster. Note that the AlexNet [3] has the same number of filters as our ZF-5 on each conv layer. The AlexNet is faster because it uses splitting on some layers, which was designed for two GPUs in [3].

We further achieve an efficient full system with the help of the recent window proposal method [25]. The Selective Search proposal [20] takes about 1-2 seconds per image on a CPU. The method of EdgeBoxes [25] only takes ∼0.2 s. Note that it is sufficient to use a fast proposal method during testing only. Using the same model trained as above (using SS), we test proposals generated by EdgeBoxes only. The mAP is 52.8 without bounding box regression. This is reasonable considering that EdgeBoxes are not used for training. Then we use both SS and EdgeBox as proposals in the training stage, and adopt only EdgeBoxes in the testing stage. The mAP is 56.3 without bounding box regression, which is better than 55.2 (Table 10) due to additional training samples. In this case, the overall testing time is ∼0.5 s per image including all steps (proposal and recognition). This makes our method practical for real-world applications. Fig. 6 shows some visual examples of our results.

4.4 Model Combination for Detection

Model combination is an important strategy for boosting CNN-based classification accuracy [3]. We propose a simple combination method for detection.

We pre-train another network in ImageNet, using the same

算的运行时间进行了公平的比较。R-CNN 的实现来自Caffe [35]中实现的作者发布的代码。我们还在 Caffe 中实现了我们的特征计算。在表 10 中，我们使用 GPU 评估 100 个随机 VOC 图像的平均时间。R-CNN 每张图像需要 14.37 秒进行卷积，而我们的单比例版本每张图像仅需 0.053 秒。所以我们比 R-CNN 快 270 倍。对于卷积，我们的五个版本每张图像需要 0.293 秒，因此比 R-CNN 快 49 倍。我们的卷积特征计算速度非常快，以至于 fc 层的计算时间占用了相当大的一部分。表 10 显示计算 4,096-d fc7 特征的 GPU 时间为每个图像 0.089 秒。考虑到卷积和完全连接的特性，我们的单尺度版本比 R-CNN 快 102 倍，低于 1.2%；我们的五级版本快 38 倍，并且具有可比较的结果。

我们还比较了表 9 中的运行时间，其中 R-CNN 使用 AlexNet [3]，如原始论文[7]中所述。我们的方法是 24×到 64 倍更快。请注意，AlexNet [3]在每个转换层上具有与 ZF-5 相同数量的滤波器。AlexNet 更快，因为它在某些层上使用拆分，这是为[3]中的两个 GPU 设计的。

借助最近的窗口提议方法[25]，我们进一步实现了一个高效的完整系统。选择性搜索提议[20]在 CPU 上每张图像大约需要 1-2 秒。EdgeBoxes [25]的方法只需要~0.2 秒。请注意，仅在测试期间使用快速提议方法就足够了。使用与上述相同的模型（使用 SS），我们仅测试由 EdgeBoxes 生成的提议。mAP 是 52.8，没有边界框回归。考虑到 EdgeBoxes 不用于训练，这是合理的。然后我们在训练阶段使用 SS 和 EdgeBox 作为建议，并且在测试阶段仅采用 EdgeBoxes。由于额外的训练样本，mAP 为 56.3，没有边界框回归，优于 55.2（表 10）。在这种情况下，每个图像的整体测试时间约为 0.5 秒，包括所有步骤（建议和识别）。这使我们的方法适用于实际应用。图 6 显示了我们结果的一些可视化示例。

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

structure but different random initializations. Then we repeat the above detection algorithm. Table 12 (SPP-net (2)) shows the results of this network. Its mAP is comparable with the first network (59.1 versus 59.2 percent), and outperforms the first network in 11 categories.

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。

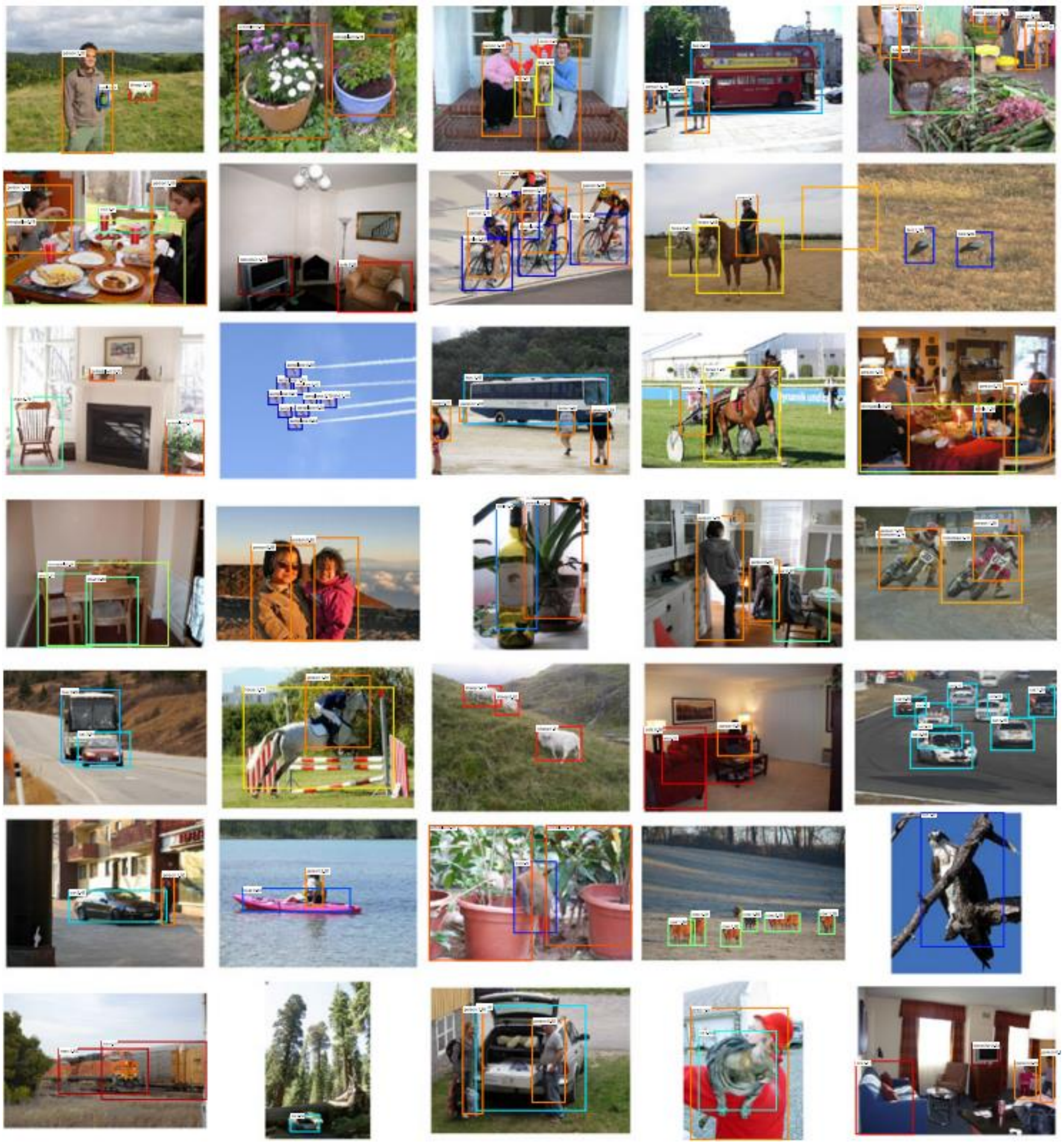


Figure 6. Example detection results of "SPP-net ftfc7 bb" on the Pascal VOC 2007 testing set (59.2 percent mAP). All windows with scores > 0 are shown. The predicted category/score are marked. The window color is associated with the predicted category. These images are manually selected because we find them impressive. Visit our project website to see all 4,952 detection results in the testing set.

图6. 在 Pascal VOC 2007 测试装置上的"SPP-net ftfc7 bb"的实例检测结果（59.2 %mAP ）。显示分数> 0 的所有窗口。预测的类别/分数被标记。窗口颜色与预测的类别相关联。手动选择这些图像是因为我们发现它们令人印象深访问我们的项目网站，查看测试集中的所有 4,952 个检测结果。

TABLE 12
Detection Results on VOC 2007 Using Model Combination

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**

The results of both models use "ftfc$_7$ bb".

Given the two models, we first use either model to score all candidate windows on the test image. Then we perform non-maximum suppression on the union of the two sets of candidate windows (with their scores). A more confident window given by one method can suppress those less confident given by the other method. After combination, the mAP is boosted to 60.9 percent (Table 12). In 17 out of all 20 categories the combination performs better than either individual model. This indicates that the two models are complementary.

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。

We further find that the complementarity is mainly because of the convolutional layers. We have tried to combine two randomly initialized fine-tuned results of the same convolutional model, and found no gain.

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

4.5 ILSVRC 2014 上的检测

The ILSVRC 2014 detection [26] task involves 200 categories. There are ∼450 k/20 k/40 k images in the training/validation/testing sets. We focus on the task of the provided-data-only track (the 1000-category CLS training data is not allowed to use).

ILSVRC 2014检测[26]任务涉及200个类别。训练/验证/测试集中有~450 k / 20 k / 40 k图像。我们专注于提供的仅数据轨道的任务（不允许使用1000类CLS训练数据）。

There are three major differences between the detection (DET) and classification (CLS) training datasets, which greatly impacts the pre-training quality. First, the DET training data is merely 1/3 of the CLS training data. This seems to be a fundamental challenge of the provided-data-only DET task. Second, the category number of DET is 1/5 of CLS. To overcome this problem, we harness the provided subcategory labels for pre-training. There are totally 499 non-overlapping subcategories (i.e., the leaf nodes in the provided category hierarchy). So we pre-train a 499-category network on the DET training set. Third, the distributions of object scales are different between DET/CLS training sets. The dominant object scale in CLS is about 0.8 of the image length, but in DET is about 0.5. To address the scale difference, we resize each training image to min(w,h)=400 (instead of 256), and randomly crop 224×224 views for training. A crop is only used when it overlaps with a ground truth object by at least 50 percent.

检测（DET）和分类（CLS）训练数据集之间存在三个主要差异，这极大地影响了训练前的质量。首先，DET训练数据仅为CLS训练数据的1/3。这似乎是仅提供数据的DET任务的基本挑战。其次，DET的类别编号是CLS的1/5。为了克服这个问题，我们利用提供的子类别标签进行预训练。总共有499个非重叠子类别（即，所提供的类别层次结构中的叶节点）。因此，我们在DET训练集上预先训练499类网络。第三，DET / CLS训练集之间的对象尺度分布是不同的。CLS中的主要对象尺度约为图像长度的0.8，但在DET中约为0.5。为了解决比例差异，我们将每个训练图像的大小调整为min（w，h）=400（而不是256），并随机裁剪224×224个视图以进行训练。只有在与真值对象重叠至少50％时才使用裁剪。

We verify the effect of pre-training on Pascal VOC 2007. For a CLS-pre-training baseline, we consider the pool 5 features (mAP 43.0 percent in Table 9). Replaced with a 200-category network pre-trained on DET, the mAP significantly drops to 32.7 percent. A 499-category pre-trained network improves the result to 35.9 percent. Interestingly, even if the amount of training data do not

我们验证了预训练对Pascal VOC 2007的影响。对于CLS训练前基线，我们考虑了池5的特征（表9中的mAP为43.0％）。替换为在DET上预训练的200类网络，mAP显着降至32.7％。499类预训练网络将结果提高到35.9％。有趣的是，即使培训数据量没有增加，培训更多类别的网络也会提高功能质量。最后，使用min（w，h）= 400而不是256的训练进一步将mAP提高到37.8％。

increase, training a network of more categories boosts the feature quality. Finally, training with min(w,h)=400 instead of 256 further improves the mAP to 37.8 percent. Even so, we see that there is still a considerable gap to the CLS-pre-training result. This indicates the importance of big data to deep learning.

For ILSVRC 2014, we train a 499-category Overfeat-7 SPP-net. The remaining steps are similar to the VOC 2007 case. Following [7], we use the validation set to generate the positive/negative samples, with windows proposed by the selective search fast mode. The training set only contributes positive samples using the ground truth windows. We fine-tune the fc layers and then train the SVMs using the samples in both validation and training sets. The bounding box regression is trained on the validation set.

Our single model leads to 31.84 percent mAP in the ILSVRC 2014 testing set [26]. We combine six similar models using the strategy introduced in this paper. The mAP is 35.11 percent in the testing set [26]. This result ranks #2 in the provided-data-only track of ILSVRC 2014 (Table 13) [26]. The winning result is 37.21 percent from NUS, which uses contextual information.

Our system still shows great advantages on speed for this dataset. It takes our single model 0.6 seconds (0.5 for conv, 0.1 for fc, excluding proposals) per testing image on a GPU extracting convolutional features from all five scales. Using the same model, it takes 32 seconds per image in the way of RCNN. For the 40 k testing images, our method requires 8 GPU·hours to compute convolutional features, while RCNN would require 15 GPU·days.

5. Conclusion

SPP is a flexible solution for handling different scales, sizes, and aspect ratios. These issues are important in visual recognition, but received little consideration in the context of deep networks. We have suggested a solution to train a deep network with a spatial pyramid pooling layer. The resulting SPP-net shows outstanding accuracy in classification/detection tasks and greatly accelerates DNN-based detection. Our studies also show that many time-proven techniques/insights in computer vision can still play important roles in deep-networks-based recognition.

6. APPENDIX

In the Appendix, we describe a few technical details that may impact the accuracy.

即便如此，我们仍然看到 CLS 预训练结果仍然存在相当大的差距。这表明大数据对深度学习的重要性。

对于 ILSVRC 2014，我们训练了 499 类 Overfeat-7 SPP-net。其余步骤类似于 VOC 2007 案例。在[7]之后，我们使用验证集来生成正/负样本，其中窗口由选择性搜索快速模式提出。训练集仅使用真值窗口提供正样本。我们微调 fc 层，然后使用验证和训练集中的样本训练 SVM。在验证集上训练边界框回归。

我们的单一模型在 ILSVRC 2014 测试集中导致 31.84%的 mAP [26]。我们使用本文介绍的策略组合了六个相似的模型。测试集中的 mAP 为 35.11%[26]。该结果在 ILSVRC 2014 的提供数据专用轨道中排名第 2（表 13）[26]。获胜的结果是新加坡国立大学的 37.21%，它使用了背景信息。

我们的系统在这个数据集的速度上仍然显示出很大的优势。在 GPU 上提取所有五个尺度的卷积特征时，每个测试图像需要我们的单个模型 0.6 秒（对于转换为 0.5，对于 fc 为 0.1，不包括建议）。使用相同的模型，RCNN 的每个图像需要 32 秒。对于 40 k 测试图像，我们的方法需要 8 个 GPU·小时来计算卷积特征，而 RCNN 需要 15 个 GPU·days。

5. 结论

SPP 是一种灵活的解决方案，用于处理不同的比例，尺寸和纵横比。这些问题在视觉识别中很重要，但在深度网络环境中却很少考虑。我们已经提出了一种解决方案来训练具有空间金字塔池层的深层网络。由此产生的 SPP-net 在分类/检测任务中表现出极高的准确性，并大大加速了基于 DNN 的检测。我们的研究还表明，许多经过时间验证的计算机视觉技术/见解仍然可以在基于深度网络的识别中发挥重要作用。

6. 附录

在附录中，我们描述了一些可能影响准确性的技术细节。

Mapping a window to feature maps. In the detection algorithm (and multi-view testing on feature maps), a window is given in the image domain, but we use it to crop the convolutional feature maps (e.g., conv 5) which have been sub-sampled several times. So we need to align the window on the feature maps. 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. This projection depends on the network architecture. For the ZF-5 network, a pixel in conv5 corresponds to a 139 × 139-pixel receptive field in the image domain, and the effective stride of conv5 in the image domain is 16. Denote x and xconv5 as the coordinates in the image domain and conv5 (we use the MATLAB convention, i.e., x starts from 1). We project the top-left corner by: $xconv5=\lfloor(x-139/2+63)/16\rfloor+1$. 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: $xconv5=\lceil(x+139/2-75)/16\rceil-1$. Here 75 is the offset computed by the bottom-right corner of the receptive field. The mapping of other network architectures can be derived in a similar way.

将窗口映射到特征图。在检测算法（以及特征图上的多视图测试）中，在图像域中给出窗口，但是我们使用它来裁剪已经多次子采样的卷积特征图（例如，conv5）。所以我们需要在特征图上对齐窗口。在我们的实现中，我们将窗口的角点投影到特征图中的像素上，使得该角点（在图像域中）最接近该像素的感受域的中心。此预测取决于网络架构。对于 ZF-5 网络，conv5 中的像素对应于图像域中的 139×139 像素感受野，并且图像域中 conv5 的有效步幅为 16.将 x 和 xconv5 表示为图像域中的坐标和 conv5（我们使用 MATLAB 约定，即 x 从 1 开始）。我们通过以下方式投射左上角：$xconv5=\lfloor(x-139/2+63)/16\rfloor+1$。这里 139/2 是感受野的半径，16 是有效步幅，63 是偏移。偏移量由图像域中感知字段的左上角计算。同样，我们通过以下方式投影右下角：$xconv5=\lceil(x+139/2-75)/16\rceil-1$。这里 75 是由感受野的右下角计算的偏移量。可以以类似的方式导出其他网络架构的映射。

Implementation of pooling bins. We use the following implementation to handle all bins when applying the network. Denote the width and height of the conv5 feature maps (can be the full image or a window) as w and h . For a pyramid level with n ×n bins, the (i,j)th bin is in the range of $[\lfloor(i-1)w/n\rfloor,\lceil iw/n\rceil]\times[\lfloor(i-1)h/n\rfloor,\lceil ih/n\rceil]$. Intuitively, if rounding is needed, we take the floor operation on the left/top boundary and ceiling on the right/bottom boundary.

池化箱的实施。我们使用以下实现来处理应用网络时的所有箱。将 conv5 特征图（可以是完整图像或窗口）的宽度和高度表示为 w 和 h。对于具有 n×n 个二进制位的金字塔等级，第（i，j）个二进制位在$[\lfloor(i-1)w/n\rfloor,\lceil iw/n\rceil]\times[\lfloor(i-1)h/n\rfloor,\lceil ih/n\rceil]$的范围内。直观地说，如果需要舍入，我们在左/上边界向下取整和在右/下边界向上取整。

Mean subtraction. The 224 × 224 cropped training/testing images are often pre-processed by subtracting the per-pixel mean [3]. When input images are in any sizes, the fixed-size mean image is not directly applicable. In the ImageNet dataset, we warp the 224 × 224 mean image to the desired size and then subtract it. In Pascal VOC 2007 and Caltech101, we use the constant mean (128) in all the experiments.

平均减法。224×224 裁剪的训练/测试图像通常通过减去每像素平均值来预处理[3]。当输入图像为任何尺寸时，固定尺寸的平均图像不能直接应用。在 ImageNet 数据集中，我们将 224×224 平均图像变形到所需大小，然后减去它。在 Pascal VOC 2007 和 Caltech101 中，我们在所有实验中使用常数均值（128）。

References

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, "Backpropagation applied to handwritten zip code recognition", Neural Comput., vol. 1, no. 4, pp. 541-551, 1989.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, "Imagenet: A large-scale hierarchical image database", Proc. Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 248-255, 2009.

[3] A. Krizhevsky, I. Sutskever, G. Hinton, "Imagenet classification with deep convolutional neural networks", Proc. Adv. Neural Inf. Process. Syst., pp. 1106-1114, 2012.

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

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

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

[7] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014.

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

[9] A. S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition", Proc. IEEE Conf. Comput. Vis. Pattern Recognit. DeepVision Workshop, pp. 806-813, 2014.

[10] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, "Deepface: Closing the gap to human-level performance in face verification", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1701-1708, 2014.

[11] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, L. Bourdevr, "Panda: Pose aligned networks for deep attribute modeling", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1637-1644, 2014.

[12] Y. Gong, L. Wang, R. Guo, S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features", arXiv:1403.1840, 2014.

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

[14] K. Grauman, T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features", Proc. 10th IEEE Int. Conf. Comput. Vis., pp. 1458-1465, 2005.

[15] S. Lazebnik, C. Schmid, J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 2169-2178, 2006.

[16] J. Sivic, A. Zisserman, "Video google: A text retrieval approach to object matching in videos", Proc. 9th IEEE Int. Conf. Comput. Vis., pp. 1470, 2003.

[17] J. Yang, K. Yu, Y. Gong, T. Huang, "Linear spatial pyramid matching using sparse coding for image classification", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1794-1801, 2009.

[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, "Locality-constrained linear coding for image classification", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 3306, 2010.

[19] F. Perronnin, J. Sánchez, T. Mensink, "Improving the Fisher kernel for large-scale image classification", Proc. 11th Eur. Conf. Comput. Vis., pp. 143-156, 2010.

[20] K. E. van de Sande, J. R. Uijlings, T. Gevers, A. W. Smeulders, "Segmentation as selective search for object recognition", Proc. 9th IEEE Int. Conf. Comput. Vis., pp. 1879-1886, 2011.

[21] L. Fei-Fei, R. Fergus, P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories", Comput. Vis. Image Understanding, vol. 106, no. 1, pp. 59-70, 2007.

[22] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, "The pascal visual object classes (VOC) challenge", 2010.

[23] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, "Object detection with discriminatively trained part-based models", IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1627-1645, Sep. 2010.

[24] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 886-893, 2005.

[25] C. L. Zitnick, P. Dollár, "Edge boxes: Locating object proposals from edges", Proc. 10th Eur. Conf. Comput. Vis., pp. 391-405, 2014.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, "Imagenet large scale visual recognition challenge", arXiv:1409.0575, 2014.

[27] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, "The devil is in the details: An evaluation of recent feature encoding methods", Proc. British Mach. Vis. Conf., pp. 1-12, 2011.

[28] A. Coates, A. Ng, "The importance of encoding versus training with sparse coding and vector quantization", Proc. 28th Int. Conf. Mach. Learn., pp. 921-928, 2011.

[29] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", Int. J. Comput. Vis., vol. 60, no. 2, pp. 91-110, 2004.

[30] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, A. W. Smeulders, "Kernel codebooks for scene categorization", Proc. 10th Eur. Conf. Comput. Vis., pp. 696-709, 2008.

[31] M. Lin, Q. Chen, 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, A. Rabinovich, "Going deeper with convolutions", arXiv:1409.4842, 2014.

[33] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv:1409.1556, 2014.

[34] M. Oquab, L. Bottou, I. Laptev, J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 1717–1724.

[35] Y. Jia, 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, C. Schmid, "Aggregating local image descriptors into compact codes", IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 9, pp. 1704-1716, Sep. 2012.

[38] C.-C. Chang, C.-J. Lin, "Libsvm: A library for support vector machines", ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, pp. 27, 2011.

[39] X. Wang, M. Yang, S. Zhu, Y. Lin, "Regionlets for generic object detection", Proc. IEEE Int. Conf. Comput. Vis., 2013, pp. 17–24.

[40] C. Szegedy, A. Toshev, D. Erhan, "Deep neural networks for object detection", Proc. Adv. Neural Inf. Process. Syst., 2013.