

**OverFeat:**

**使用卷积神经网络进行集成识别定位和检测**

**Pierre Sermanet David Eigen**

**Xiang Zhang Michael Mathieu Rob Fergus Yann LeCun**

纽约大学Courant数学科学研究所

百老汇719 , 12楼, 纽约, 邮政编码10003 sermanet,deigen,xiang,mathieu,fergus,yann@cs.nyu.edu

# Abstract

我们提出了一个使用卷积网络进行分类的集成框架，用于目标的定位和检测，我们展示了如何在卷积神经网络中有效地实现多尺度（可变的检测框范围）和滑动窗口方法，我们还引入了一种新颖的深度学习方法来通过学习预测对象边界来进行定位。然后使用累积边界框的方法（不是抑制边界框），以提高检测置信度。我们展示了可以使用单个共享网络同时学习不同的任务。这一集成框架是ImageNet大规模视觉识别挑战赛2013（ILSVRC2013）定位任务的获胜者，并在检测和分类任务中取得了非常有竞争力的结果。在比赛之后的工作中，我们建立了检测任务的新的最先进技术。最后，我们发布了我们最佳模型OverFeat的特征提取器。

# 介绍

识别图像中主要物体的类别这项任务，长期以来一直是使用CNN来进行的, 无论是手写字符、房屋编号，还是无纹理玩具（比如简单的塑料玩具或颜色均匀的玩具）、交通标志、Caltech-101数据集中的物体，或者包含1000多个类别的imageNet数据集中的物体。

在像Caltech-101这样的小型数据集上，CNN的准确性虽然不错，但并没有创纪录的表现。然而随着更大型数据集的出现（如包含一千多个类别的ImageNet数据集上），CNN已显著提升最新技术水平。

对于这类任务来说，CNN的主要优势在于整个系统是端到端训练的，从原始像素一直训练到到最终类别，从而减轻了手动设计合适的特征提取器的需求。主要的缺点是它们对标记的训练样本有着巨大的需求。本文的主要观点是展示通过训练CNN同时对图像中的物体进行分类、定位和检测，可以提高所有任务的分类准确性以及检测和定位的准确性。本文提出了一种新的集成方法，利用单个CNN进行对象检测、识别和定位。同时还介绍了一种累积预测边界框的新方法进行目标定位和检测。我们建议通过结合多个定位预测，可以在不对背景样本进行训练的情况下进行检测，并且可以避免耗时复杂的引导式训练过程。不对背景进行训练还可以让网络专注于正类别，从而提高准确性。在ImageNet ILSVRC 2012和2013数据集上进行了实验，并在ILSVRC 2013的定位和检测任务上取得了最新的技术成果。虽然ImageNet数据集中的图像大多选择了主要目标大致居中的对象，但目标有时在图像中的大小和位置会有显著变化。解决这一问题的第一个想法是以滑动窗口的方式，在图像中的多个位置和多个尺度上应用CNN。然而采用这种方式，可能有许多窗口包含对象的一个完全可识别的部分（比如狗的头部），但并非整个对象，甚至不是对象的中心。这会导致良好的分类但较差的定位和检测。因此，第二个想法是训练系统不仅为每个窗口生成一个类别分布，还要预测相对于窗口包含对象的边界框的位置和大小。第三个想法是累积每个位置和大小的每个类别的证据。许多作者提出使用CNN结合多尺度的滑动窗口进行检测和定位，早在20世纪90年代就有人用于多字符字符串、人脸和手部的检测。最近，CNN在自然图像中的文本检测、人脸检测和行人检测等方面已经被证明能够达到最先进的性能。

一些作者还提出训练CNN直接预测待定位对象的实例化参数，比如相对于视窗的位置或对象的姿态。例如，Osadchy等人描述了一个用于同时进行人脸检测和姿态估计的CNN。人脸在九维输出空间中被表示为一个三维流形。流形上的位置表示姿态（俯仰、偏航和翻滚）。当训练图像是一张人脸时，网络被训练在已知姿态的位置上产生流形上的点。如果图像不是人脸，则输出被推离流形。在测试时，到流形的距离表示图像是否包含人脸，而流形上最接近点的位置表示姿态。Taylor等人使用CNN来估计身体部位的位置（手、头等），从而推导出人体姿势。他们使用度量学习准则来训练网络在身体姿势流形上产生点。Hinton等人还提出训练网络在识别过程中计算特征的显式实例化参数。其他作者提出通过基于CNN的分割来进行物体定位。最简单的方法是训练CNN将其视窗的中心像素（对于体积图像则是体素）分类为区域边界或非边界。但是当区域必须被分类时，最好执行语义分割。主要思想是训练ConvNet将视窗的中心像素与所属对象的类别进行分类，利用窗口作为决策的上下文。应用范围从生物图像分析、移动机器人的障碍标记到照片标记。这种方法的优势在于边界轮廓不需要是矩形，区域也不需要是明确定义的对象。缺点在于它需要密集的像素级标签进行训练。这种分割预处理或对象提议步骤最近在传统计算机视觉中变得流行，以减少检测的位置、尺度和长宽比的搜索空间[19, 2, 6, 29]。因此，可以在搜索空间中的最佳位置应用昂贵的分类方法，从而提高识别准确性。此外，[29, 1]指出，这些方法通过大幅减少不太可能包含对象的区域，从而减少潜在的误报。然而，我们的密集滑动窗口方法能够在ILSVRC13检测数据集上胜过对象提议方法。  
Krizhevsky等人[15]最近展示了使用大型ConvNet的令人印象深刻的分类性能。作者还参加了ImageNet 2012比赛，赢得了分类和定位挑战。尽管他们展示了令人印象深刻的定位性能，但尚未有公开发表的论文描述他们的方法。因此，我们的论文是第一篇清晰解释了如何利用ConvNets进行ImageNet数据的定位和检测。

在本文中，我们使用"定位"和"检测"这两个术语的方式与它们在ImageNet 2013比赛中的使用一致，即唯一的区别是所使用的评估标准，两者都涉及预测图像中每个对象的边界框。

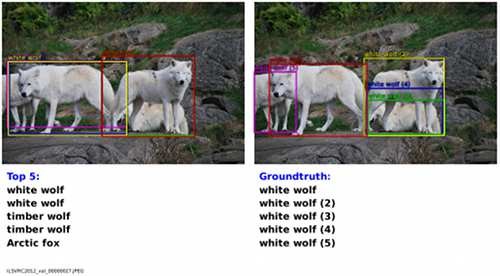


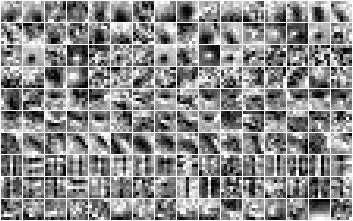
Figure 1: 定位（顶部）和检测任务（底部）。左侧图像包含我们的预测（按置信度降序排列），而右侧图像显示了地面实况标签。检测图像（底部）说明了检测数据集的更高难度，其中可能包含许多小对象，而分类和定位图像通常包含单个大对象。

在本文中，我们探讨了三个计算机视觉任务，难度逐渐增加：(i)分类，(ii)定位，和(iii)检测。每个任务都是下一个任务的子任务。虽然所有任务都使用单一框架和共享的特征学习基础进行处理，但我们将在接下来的部分分别描述它们。

在本文中，我们报告了2013年ImageNet大规模视觉识别挑战（ILSVRC2013）的结果。在这一挑战的分类任务中，每个图像被分配一个与图像中主要对象相对应的单个标签。允许进行五次猜测以找到正确答案（这是因为图像也可能包含多个未标记的对象）。定位任务与分类任务类似，每个图像允许进行5次猜测，但另外，每次猜测都必须返回预测对象的边界框。要被认为是正确的，预测的边界框必须至少与地面实况匹配50%（使用PASCAL联合/交集标准），并且必须标记为正确的类别（即每个预测都是一个标签和一个边界框，它们是相关联的）。检测任务与定位任务不同之处在于每个图像中可以包含任意数量的对象（包括零），而误报会受到均值平均精度（mAP）的惩罚。定位任务是分类和检测之间的一个方便的中间步骤，它使我们能够独立评估我们的定位方法，而不受检测特定挑战的影响（比如学习背景类）。在图1中，我们展示了具有我们的定位/检测预测以及相应地面实况的图像示例。请注意，分类和定位共享相同的数据集，而检测还包括对象可能较小的额外数据。检测数据还包含一组图像，其中某些对象不存在。这可以用于引导，但在这项工作中我们没有利用它。

# 分类任务

我们的分类架构类似于Krizhevsky等人在ILSVRC12比赛中最好的架构。然而，我们在网络设计和推理步骤上进行了改进。由于时间限制，Krizhevsky模型中的一些训练特征没有被探索，因此我们希望我们的结果可以进一步改善。这些内容将在未来工作的第6节中讨论。



## 图2：第1层（顶部）和第2层滤波器（底部）。

## 模型设计和训练

我们在ImageNet 2012训练集（120万张图像和1000个类别）上对网络进行训练。我们的模型在训练过程中采用了由Krizhevsky等人提出的固定输入尺寸方法，但在分类过程中采用了多尺度方法，具体描述见下一节。每个图像都被下采样，使得最小的尺寸为256像素。然后我们提取尺寸为221x221像素的5个随机裁剪图像（以及它们的水平翻转），并将它们以大小为128的mini-batch的形式输入到网络中。网络中的权重是随机初始化的，(µ,σ) = (0,1 × 10−2)。它们随后通过随机梯度下降进行更新，伴随着动量项为0.6和ℓ2权重衰减为1 × 10−5。学习率最初为5 × 10−2，在(30,50,60,70,80)个epoch后逐渐减小0.5倍。在分类器的全连接层（第6和第7层）中采用了DropOut [11]，丢弃率为0.5。

我们详细介绍了表1和表3中的架构大小。需要注意的是，在训练过程中，我们将这个架构视为非空间结构（输出大小为1x1），而在推理步骤中，它会产生空间输出。第1至第5层与Krizhevsky等人的架构相似，使用修正线性单元（"relu"）非线性和最大池化，但存在以下差异：(i) 不使用对比度归一化；(ii) 池化区域不重叠；(iii) 我们的模型具有更大的第1层和第2层特征图，这要归功于较小的步幅（2而不是4）。较大的步幅有利于速度，但会降低准确性。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | Output |
| Layer | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Stage | conv + max | conv + max | conv | conv | conv + max | full | full | full |
| # channels | 96 | 256 | 512 | 1024 | 1024 | 3072 | 4096 | 1000 |
| Filter size | 11x11 | 5x5 | 3x3 | 3x3 | 3x3 | - | - | - |
| Conv. stride | 4x4 | 1x1 | 1x1 | 1x1 | 1x1 | - | - | - |
| Pooling size | 2x2 | 2x2 | - | - | 2x2 | - | - | - |
| Pooling stride | 2x2 | 2x2 | - | - | 2x2 | - | - | - |
| Zero-Padding size | - | - | 1x1x1x1 | 1x1x1x1 | 1x1x1x1 | - | - | - |
| Spatial input size | 231x231 | 24x24 | 12x12 | 12x12 | 12x12 | 6x6 | 1x1 | 1x1 |

表1：快速模型的架构细节。特征图的空间大小取决于输入图像的大小，在我们的推理步骤中会发生变化（请参阅附录中的表5）。这里我们展示训练时的空间尺寸。第5层是顶部卷积层。随后的层是全连接层，在测试时以滑动窗口方式应用。全连接层也可以在空间设置中看作是1x1的卷积。准确模型的类似尺寸可以在附录中找到。

在图2中，我们展示了来自前两个卷积层的滤波器系数。第一层滤波器捕捉了定向边缘、图案和斑块。在第二层，滤波器具有各种形式，有些是扩散的，有些具有强烈的线条结构或定向边缘。

## 特征提取器

除了本论文之外，我们还发布了一个名为“OverFeat”的特征提取器，以便为计算机视觉研究提供强大的特征。我们提供了两个模型，一个快速模型和一个准确模型。每个架构都在表1和表3中进行了描述。此外，我们还在表4中比较了它们的参数和连接大小。准确模型比快速模型更准确（表2中的分类错误率为14.18%，而快速模型为16.39%），但它需要的连接数量几乎是快速模型的两倍。使用7个准确模型的委员会达到了13.6%的分类错误率，如图4所示。

## 多尺度分类

在[15]中，使用了多视图投票来提高性能：固定的10个视图（4个角和中心，水平翻转）进行平均。然而，这种方法可能会忽略图像的许多区域，并且在视图重叠时会存在计算冗余。此外，它仅适用于单一尺度，这可能不是ConvNet能够以最佳置信度响应的尺度。 相反，我们通过在每个位置和多个尺度上密集地运行网络来探索整个图像。虽然滑动窗口方法在某些模型类型上可能在计算上具有限制性，但在ConvNets的情况下本质上是高效的（参见第3.5节）。这种方法产生了更多的视图用于投票，从而增加了鲁棒性，同时保持了高效性。在任何尺度上，将ConvNet卷积到任意大小的图像的结果是每个尺度上的C维向量的空间映射。 然而，上述网络中的总下采样比例为2x3x2x3，即36。因此，当密集应用时，这种架构只能在输入维度的每个轴上每36个像素产生一个分类向量。与10个视图方案相比，这种输出的粗糙分布会降低性能，因为网络窗口与图像中的对象不对齐。网络窗口与对象对齐得越好，网络响应的置信度就越高。为了解决这个问题，我们采取了一种类似于Giusti等人[9]介绍的方法，并在每个偏移处应用最后的下采样操作。这可以消除来自这一层的分辨率损失，将总下采样比率从x36改为x12。 现在我们详细解释分辨率增强是如何进行的。我们使用6种输入尺度，导致未汇集的第5层特征图具有不同的分辨率（有关详细信息，请参见表5）。然后，使用以下过程将这些特征图进行汇集并呈现给分类器，如图3所示：

1. For a single image, at a given scale, we start with the unpooled layer 5 feature maps.
2. Each of unpooled maps undergoes a 3x3 max pooling operation (non-overlapping regions),repeated 3x3 times for (∆*x,*∆*y*) pixel offsets of {0*,*1*,*2}.
3. This produces a set of pooled feature maps, replicated (3x3) times for different (∆*x,*∆*y*) combinations.
4. The classifier (layers 6,7,8) has a fixed input size of 5x5 and produces a *C*-dimensional output vector for each location within the pooled maps. The classifier is applied in sliding-window fashion to the pooled maps, yielding *C*-dimensional output maps (for a given (∆*x,*∆*y*) combination).
5. The output maps for different (∆*x,*∆*y*) combinations are reshaped into a single 3D output map (two spatial dimensions x *C* classes).

(a) 对于单个图像，在给定的尺度上，我们从未汇集的第5层特征图开始。

(b) 每个未汇集的特征图都经历了3x3的最大池化操作（非重叠区域），对于{0,1,2}的像素偏移(∆x,∆y)，重复3x3次。

(c) 这产生了一组汇集的特征图，对于不同的(∆x,∆y)组合进行了(3x3)次复制。

(d) 分类器（第6、7、8层）具有固定的5x5输入尺寸，并对汇集的特征图以滑动窗口的方式进行应用，为汇集的特征图中的每个位置产生一个C维输出向量。对于给定的(∆x,∆y)组合，分类器以滑动窗口的方式应用于汇集的特征图，产生C维输出特征图。 (e) 不同(∆x,∆y)组合的输出特征图被重塑为一个单一的3D输出特征图（两个空间维度 x C类）。

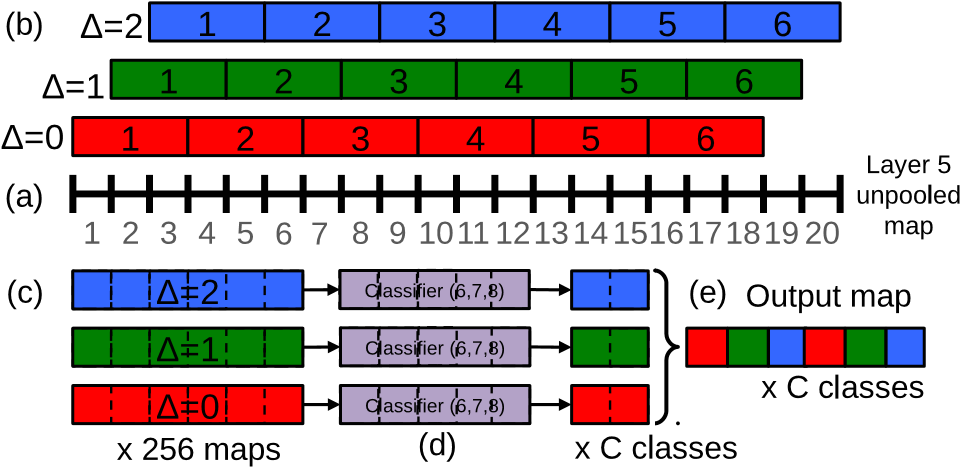


Figure 3: 1D illustration (to scale) of output map computation for classification, using *y*-dimension from scale 2 as an example (see Table 5). (a): 20 pixel unpooled layer 5 feature map. (b): max pooling over non-overlapping 3 pixel groups, using offsets of ∆ = {0*,*1*,*2} pixels (red, green, blue respectively). (c): The resulting 6 pixel pooled maps, for different ∆. (d): 5 pixel classifier (layers 6,7) is applied in sliding window fashion to pooled maps, yielding 2 pixel by *C* maps for each ∆. (e): reshaped into 6 pixel by *C* output maps.

图3: 以比例绘制的一维示例，用于对分类进行输出映射计算，以尺度2的y维度为例（请参见表5）。 (a): 20像素未汇集的第5层特征图。 (b): 对非重叠的3像素组进行最大池化，使用∆ = {0,1,2}像素的偏移（分别为红色、绿色、蓝色）。 (c): 不同∆的结果为6像素的汇集特征图。 (d): 5像素的分类器（第6、7层）以滑动窗口的方式应用于汇集特征图，为每个∆产生了2像素乘以C的特征图。 (e): 重塑为6像素乘以C的输出特征图。

这些操作可以被视为通过池化层每次将分类器的观察窗口向后移动1像素，而不进行下采样，并在下一层使用跳过卷积核（其中邻域中的值是非相邻的）。或者等效地，可以将最终的池化层和全连接堆栈应用于每个可能的偏移，并通过交错输出的方式组装结果。 上述过程对每个图像的水平翻转版本都会重复。然后，我们通过以下步骤产生最终的分类结果：(i) 对每个类别在每个尺度和翻转下取空间最大值；(ii) 对来自不同尺度和翻转的得到的C维向量进行平均；(iii) 从平均类别向量中取出前1或前5个元素（取决于评估标准）。 在直观层面上，网络的两个部分——即特征提取层（1-5）和分类器层（6-输出）——以相反的方式使用。在特征提取部分，滤波器在一次传递中横跨整个图像进行卷积。从计算的角度来看，这比在图像上滑动固定大小的特征提取器然后聚合不同位置的结果要高效得多。然而，这些原则在网络的分类器部分被颠倒。在这里，我们希望在第5层特征图中寻找不同位置和尺度的固定大小表示。因此，分类器具有固定大小的5x5输入，并且被彻底地应用于第5层的特征图。通过使用单像素的移位（∆x,∆y）的穷举式池化方案，确保我们可以在分类器和特征图中对象的表示之间获得良好的对齐。

## 结果

在表2中，我们尝试了不同的方法，并将它们与Krizhevsky等人[15]的单网络模型进行了比较。上述描述的方法，使用6个尺度，实现了13.6%的top-5错误率。如预期的那样，使用较少的尺度会影响性能：单尺度模型的top-5错误率为16.97%。如图3所示的细微步进技术在单尺度情况下带来了相对较小的改进，但对于此处显示的多尺度增益也很重要。

|  |  |  |
| --- | --- | --- |
|  | Top-1 | Top-5 |
| Approach | error % | error % |
| Krizhevsky *et al.* [15] | 40.7 | 18.2 |
| OverFeat - 1 *fast* model, scale 1, coarse stride | 39.28 | 17.12 |
| OverFeat - 1 *fast* model, scale 1, fine stride | 39.01 | 16.97 |
| OverFeat - 1 *fast* model, 4 scales (1,2,4,6), fine stride | 38.57 | 16.39 |
| OverFeat - 1 *fast* model, 6 scales (1-6), fine stride | 38.12 | 16.27 |
| OverFeat - 1 *accurate* model, 4 corners + center + flip | 35.60 | 14.71 |
| OverFeat - 1 *accurate* model, 4 scales, fine stride | 35.74 | 14.18 |
| OverFeat - 7 *fast* models, 4 scales, fine stride | 35.10 | 13.86 |
| OverFeat - 7 *accurate* models, 4 scales, fine stride | 33.96 | 13.24 |

Table 2: **Classification experiments on validation set.** Fine/coarse stride refers to the number of ∆ values used when applying the classifier. Fine: ∆ = 0*,*1*,*2; coarse: ∆ = 0.

表2：在验证集上进行的分类实验。细/粗步进是指应用分类器时使用的∆值的数量。细：∆ = 0,1,2；粗：∆ = 0。

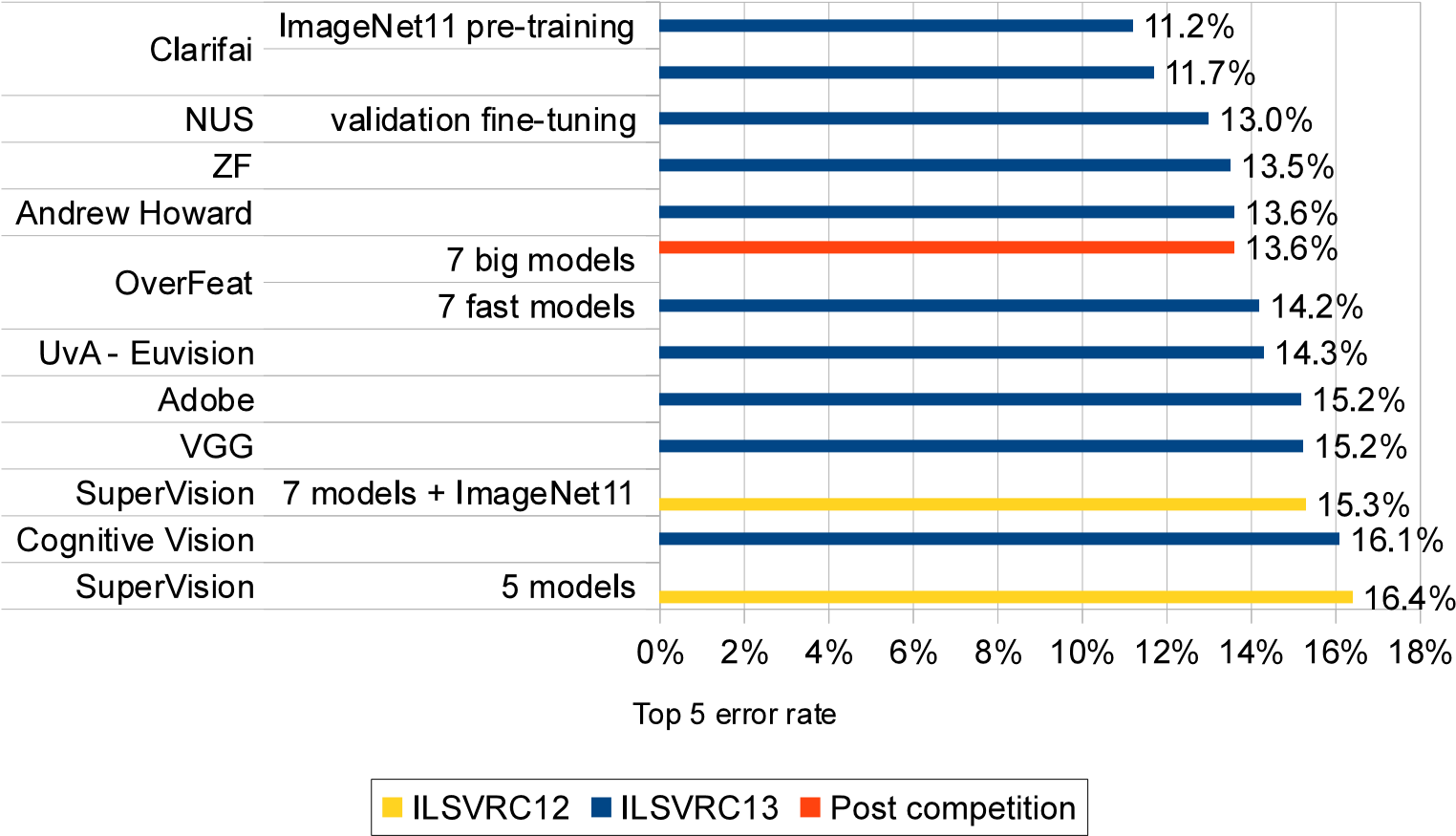


Figure 4: **Test set classification results.** During the competition, OverFeat yielded 14.2% top 5 error rate using an average of 7 fast models. In post-competition work, OverFeat ranks fifth with 13.6% error using bigger models (more features and more layers).

图4：测试集分类结果。在比赛期间，OverFeat使用7个快速模型的平均值产生了14.2%的top 5错误率。在比赛后的工作中，使用更大的模型（更多特征和更多层）的OverFeat以13.6%的错误率排名第五。

我们在图4中报告了2013年比赛的测试集结果，我们的模型（OverFeat）通过对7个ConvNets的投票（每个使用不同的初始化进行训练）获得了14.2%的准确率，并在18个团队中排名第5。仅使用ILSVRC13数据的最佳准确率为11.7%。通过使用来自ImageNet Fall11数据集的额外数据进行预训练，将这一数字提高到了11.2%。在比赛后的工作中，通过使用更大的模型（更多特征和更多层），我们将OverFeat的结果改进到了13.6%的错误率。由于时间限制，这些更大的模型并未完全训练，预计会有更多的改进出现。

## 卷积神经网络和滑动窗口的效率

与许多滑动窗口方法相比，这些方法一次只计算输入的每个窗口的整个流水线，卷积神经网络在滑动方式下具有固有的高效性，因为它们自然地共享重叠区域的常见计算。在测试时将我们的网络应用于更大的图像时，我们只需将每个卷积应用于整个图像的范围。这将扩展每一层的输出以覆盖新的图像尺寸，最终产生一个包含输出类别预测的地图，其中每个输入的“窗口”（视野）都对应一个空间位置。

input

st stage

1

output

classifier

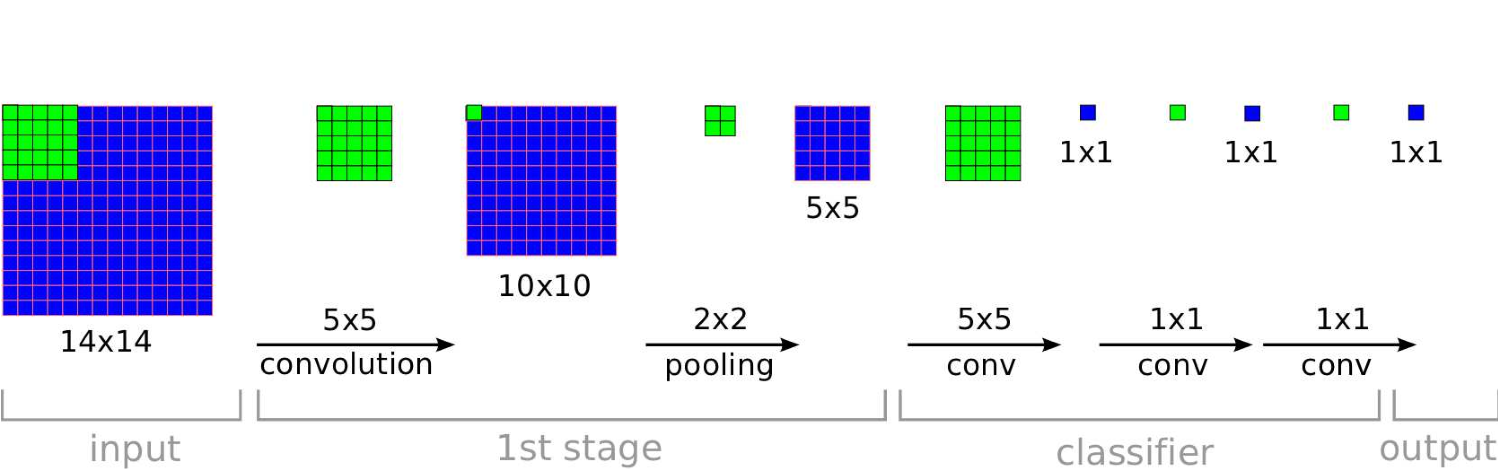
convolution

pooling

conv

conv

conv



input

1

st stage

output

classifier

convolution

pooling

conv

conv

conv

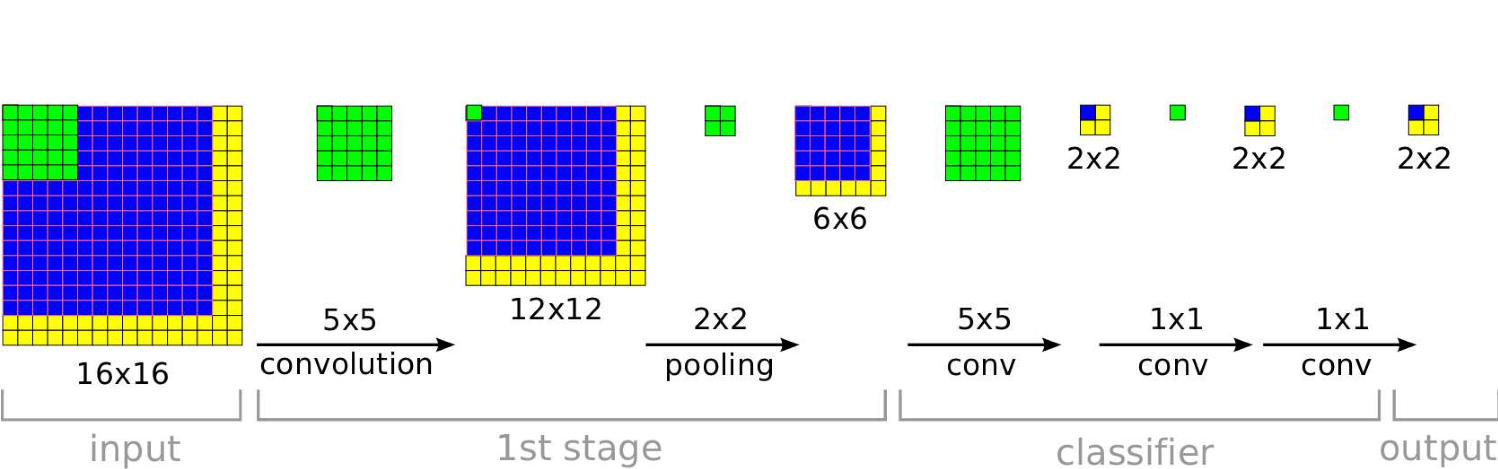


图5：卷积神经网络在检测中的效率。在训练期间，卷积神经网络仅产生单个空间输出（顶部）。但当在测试时应用于更大的图像时，它会产生一个空间输出地图，例如2x2（底部）。由于所有层都是以卷积方式应用的，因此对于更大图像所需的额外计算仅限于黄色区域。为简单起见，该图表省略了特征维度。

这在图5中有所体现。卷积是自底向上应用的，因此邻近窗口共同的计算只需要执行一次。

请注意，我们架构的最后几层是全连接的线性层。在测试时，这些层实际上被具有1x1空间范围的卷积操作所替代。整个ConvNet就是一系列仅包含卷积、最大池化和阈值操作的序列。

# 定位

从我们经过分类训练的网络开始，我们将分类器层替换为回归网络，并训练它在每个空间位置和尺度上预测物体边界框。然后，我们将回归预测与每个位置的分类结果结合在一起，接下来我们将详细描述这一过程。

## 生成预测

为了生成物体边界框的预测，我们同时在所有位置和尺度上运行分类器和回归器网络。由于它们共享相同的特征提取层，因此在计算分类网络后，只需要重新计算最终的回归层。在每个位置的最终softmax层的输出对于类别c提供了一个置信度分数，表示相应视野中存在类别c的对象的置信度（尽管不一定完全包含）。因此，我们可以为每个边界框分配一个置信度。

## 回归器训练

回归网络以第5层的池化特征图作为输入。它包括2个分别为4096和1024个通道的全连接隐藏层。最终的输出层有4个单元，用于指定边界框边缘的坐标。与分类一样，这也是由于∆x,∆y的偏移而产生的（3x3）个副本。该架构如图8所示。

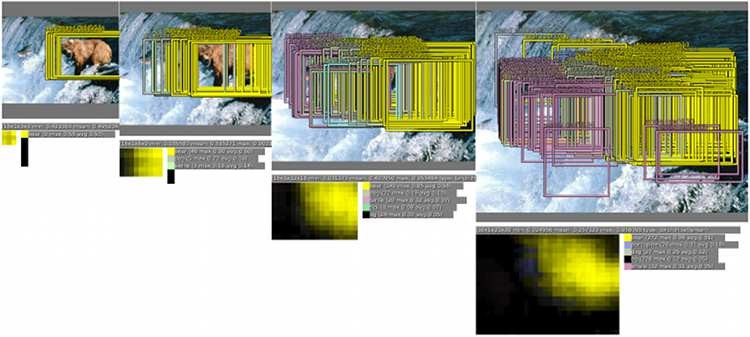
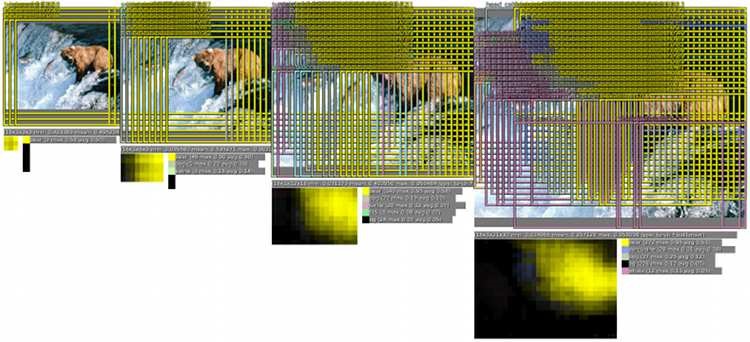
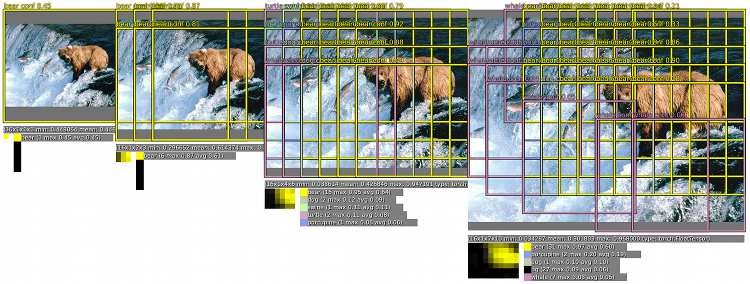


图6：定位/检测流程。原始的分类器/检测器为每个位置输出一个类别和置信度（第1张图）。可以使用第3.3节中描述的方法来增加这些预测的分辨率（第2张图）。然后，回归器预测对象相对于每个窗口的位置尺度（第3张图）。然后这些边界框被合并并累积到少量的对象中（第4张图）。

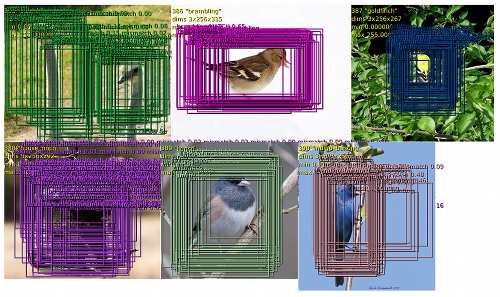


图7：回归网络生成的边界框示例，在合并为最终预测之前。这里显示的示例是在单个尺度上。根据物体的不同，预测可能在其他尺度上更为准确。在这里，大多数最初以网格形式组织的边界框会收敛到单个位置和尺度。这表明网络对对象的位置非常自信，而不是随机分散。左上方的图像显示，如果存在多个对象，它也能正确识别多个位置。预测的边界框的各种长宽比表明网络能够处理各种对象姿势。

我们固定了分类网络的特征提取层（1-5），并使用每个示例的预测边界框和真实边界框之间的ℓ2损失来训练回归网络。最终的回归器层是特定于类别的，有1000个不同版本，每个类别一个。我们使用与第3节中描述的相同尺度集合来训练这个网络。我们将回归网络在每个空间位置的预测与位于参考帧内的真实边界框进行比较，参考了卷积中的回归器平移偏移（参见图8）。然而，我们不会对与输入视野重叠度低于50%的边界框进行训练：因为对象大部分在这些位置之外，所以通过包含对象的回归窗口更好地处理这些情况。

以多尺度方式训练回归器对于跨尺度预测组合至关重要。在单个尺度上训练会在该尺度上表现良好，并且在其他尺度上也能表现合理。然而，多尺度训练将使预测在不同尺度上正确匹配，并且指数级增加合并预测的置信度。反过来，这使我们能够仅使用少量尺度表现良好，而不是像通常的检测情况下那样需要许多尺度。行人检测中一个尺度到另一个尺度的典型比率大约为1.05到1.1[25]，但在这里我们使用大约1.4的较大比率（由于尺度的维度被调整以确切适应我们网络的步幅，因此每个尺度的这个数字会有所不同），这使我们的系统运行更快。

## 组合预测

We combine the individual predictions (see Fig. 7) via a greedy merge strategy applied to the regressor bounding boxes, using the following algorithm.

我们通过贪婪合并策略将个别预测（见图7）组合起来，使用以下算法。

1. Assign to *Cs* the set of classes in the top *k* for each scale *s* ∈ 1*...*6, found by taking the maximum detection class outputs across spatial locations for that scale.
   1. 对于每个尺度 s ∈ 1...6，通过获取该尺度空间位置上的最大检测类别输出，为每个尺度s分配前k个类别的集合Cs。
2. Assign to *Bs* the set of bounding boxes predicted by the regressor network for each class in *Cs*, across all spatial locations at scale *s*.
   1. 为Cs中的每个类别，对于每个尺度s，在所有空间位置上，通过回归器网络预测的边界框集合为Bs。

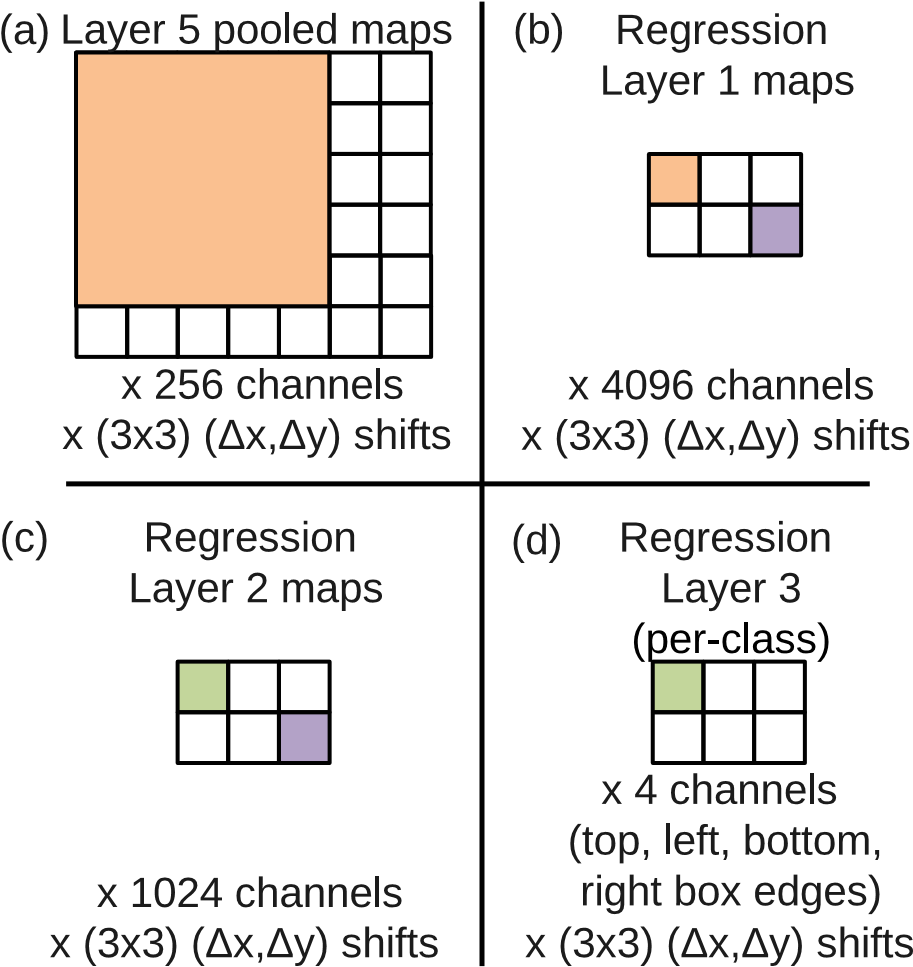


图8：将回归网络应用于第5层特征，在尺度2上的示例。 (a) 在该尺度上，回归器的输入为每个(3x3) ∆x,∆y偏移的6x7像素空间和256个通道。 (b) 回归网络的第一层中的每个单元连接到第5层地图中的一个5x5空间邻域，以及所有256个通道。移动5x5邻域会导致每个4096个通道的图像具有2x3的空间范围，以及每个(3x3) ∆x,∆y偏移。 (c) 第二个回归层有1024个单元，并且是全连接的（即紫色元素只与(b)中的紫色元素连接，跨越所有4096个通道）。 (d) 回归网络的输出是一个4维向量（指定边界框的边缘）对于2x3地图中的每个位置，以及每个(3x3) ∆x,∆y偏移。

1. Assign *B* ←S*s Bs*
2. Repeat merging until done:

) = argmin*b*16=*b*2∈*B*matchscore(b1*,*b2)

1. If matchscore, stop.
2. Otherwise, setboxmerge

In the above, we compute matchscore using the sum of the distance between centers of the two bounding boxes and the intersection area of the boxes. boxmerge compute the average of the bounding boxes’ coordinates.

The final prediction is given by taking the merged bounding boxes with maximum class scores. This is computed by cumulatively adding the detection class outputs associated with the input windows from which each bounding box was predicted. See Fig. 6 for an example of bounding boxes merged into a single high-confidence bounding box. In that example, some *turtle* and *whale* bounding boxes appear in the intermediate multi-scale steps, but disappear in the final detection image. Not only do these bounding boxes have low classification confidence (at most 0.11 and 0.12 respectively), their collection is not as coherent as the *bear* bounding boxes to get a significant confidence boost. The *bear* boxes have a strong confidence (approximately 0.5 on average per scale) and high matching scores. Hence after merging, many *bear* boundingboxes are fused into a single very high confidence box, while false positives disappear below the detection threshold due their lack of bounding box coherence and confidence. This analysis suggest that our approach is naturally more robust to false positives coming from the pure-classification model than traditional non-maximum suppression, by rewarding bounding box coherence.

在上述过程中，我们使用两个边界框中心之间的距离和边界框交集区域的总和来计算匹配分数。边界框合并计算边界框坐标的平均值。最终预测是通过选择具有最大类别分数的合并边界框来确定的。这是通过累加与预测每个边界框的输入窗口相关联的检测类别输出来计算的。请参见图6，其中显示了边界框合并为单个高置信度边界框的示例。在该示例中，一些乌龟和鲸鱼的边界框出现在中间的多尺度步骤中，但在最终的检测图像中消失了。这些边界框不仅具有较低的分类置信度（最多分别为0.11和0.12），它们的收集也不像熊的边界框那样一致，无法获得显著的置信度提升。熊的边界框具有很高的置信度（每个尺度平均约0.5），并且具有较高的匹配分数。因此，在合并后，许多熊的边界框被融合为一个非常高置信度的边界框，而假阳性由于缺乏边界框一致性和置信度而消失在检测阈值以下。这一分析表明，相比传统的非极大值抑制方法，我们的方法自然更加稳健，通过奖励边界框的一致性来抵御来自纯分类模型的假阳性。

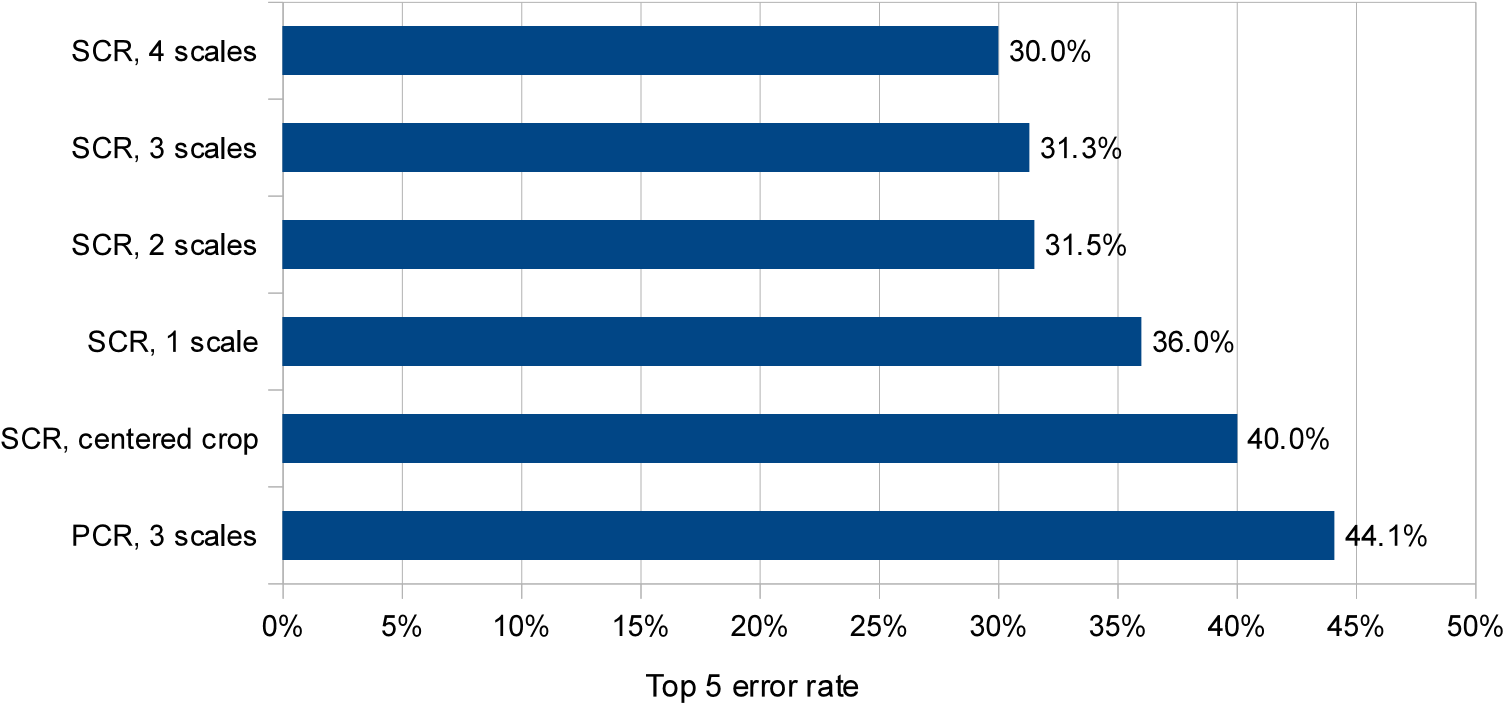


Figure 9: **Localization experiments on ILSVRC12 validation set.** We experiment with different number of scales and with the use of single-class regression (SCR) or per-class regression (PCR).

图9：在ILSVRC12验证集上进行的定位实验。我们尝试不同数量的尺度，并使用单类别回归（SCR）或每类别回归（PCR）。

## 实验

我们将我们的网络应用于Imagenet 2012验证集，使用了比赛规定的定位标准。这些结果如图9所示。图10展示了2012年和2013年的定位比赛结果（这两年的训练和测试数据是相同的）。我们的方法在2013年的比赛中获胜，错误率为29.9%。 我们的多尺度和多视角方法对于获得良好性能至关重要，如图9所示：仅使用单个中心裁剪，我们的回归网络达到了40%的错误率。通过组合两个尺度所有空间位置上的回归器预测，我们实现了31.5%的大幅改善的错误率。增加第三和第四个尺度进一步提高到30.0%的错误率。 对于回归网络中的每个类别使用不同的顶层（图9中的Per-Class Regressor (PCR)）令人惊讶地没有超过仅使用在所有类别之间共享的单个网络（44.1% vs. 31.3%）。这可能是因为在训练集中，每个类别的带有边界框注释的示例相对较少，而网络的顶层参数则多1000倍，导致训练不足。这种方法可能通过仅在相似类别之间共享参数（例如，为所有狗类别训练一个网络，为车辆类别训练另一个网络等）来改进。

# Detection

检测训练与分类训练类似，但是以空间方式进行。图像的多个位置可以同时进行训练。由于模型是卷积的，所有权重在所有位置之间共享。与定位任务的主要区别在于，在没有物体存在时需要预测背景类。传统上，负例最初是随机选择进行训练的。然后，在引导式训练中，将最严重的负误差添加到训练集中。独立的引导式训练使得训练复杂化，并可能导致负例集合和训练时间之间的潜在不匹配。此外，需要调整引导式训练的规模，以确保训练不会过度拟合于一个小集合。为了规避所有这些问题，我们通过在图像中选择一些有趣的负例，比如随机的或者最严重的负例，来实时进行负训练。这种方法在计算上更昂贵，但使得该过程更加简单。而且，由于特征提取最初是通过分类任务进行训练的，因此检测微调也不会太长。

在图11中，我们报告了我们的检测系统在ILSVRC 2013比赛中排名第三，均值平均精度（mAP）为19.4%。后来，我们建立了一个新的检测技术水平，mAP为24.3%。请注意，前三名方法和其他团队之间存在很大的差距（第四名...）。



图10：ILSVRC12和ILSVRC13比赛结果（测试集）。我们的参赛作品在ILSVRC13定位比赛中获得了29.9%的错误率（前5名）。请注意，这两年的训练和测试数据是相同的。OverFeat参赛作品使用了4个尺度和单类别回归方法。



Figure 11: **ILSVRC13 test set Detection results.** During the competition, UvA ranked first with 22.6% mAP. In post competition work, we establish a new state of the art with 24.3% mAP. Systems marked with \* were pre-trained with the ILSVRC12 classification data.

我们的方法得到了11.5%的平均精度（mAP）。此外，我们的方法与排名前两的系统有很大不同，它们使用初始分割步骤将候选窗口数量从大约200,000个减少到2,000个。这种技术加快了推断速度，并大大减少了潜在的误报。[29, 1]指出，与选择性搜索丢弃不太可能包含目标的位置以减少假阳性的方法相比，使用密集滑动窗口会降低检测准确度。结合我们的方法，我们可能会观察到类似的改进，就像传统的密集方法和基于分割的方法之间所见到的改进一样。值得注意的是，我们没有在检测验证集上进行微调，而NEC和UvA进行了微调。验证集和测试集的分布与训练集有明显的差异，这本身就使结果提高了大约1个百分点。图11中OverFeat两个结果之间的改进是由于更长的训练时间和上下文的使用，即每个尺度也使用较低分辨率的尺度作为输入。

# 讨论

我们提出了一种多尺度滑动窗口方法，可用于分类、定位和检测。我们将其应用于ILSVRC 2013数据集，目前在分类中排名第4，在定位中排名第1，在检测中排名第1。我们论文的第二个重要贡献是解释了如何有效地利用ConvNets进行检测和定位任务。这些在[15]中从未被解决过，因此我们是第一个在ImageNet 2012的背景下解释如何做到这一点的人。我们提出的方案对设计用于分类的网络进行了重大修改，但清楚地表明ConvNets能够胜任这些更具挑战性的任务。我们的定位方法赢得了2013年ILSVRC比赛，并明显优于所有2012年和2013年的方法。在比赛期间，检测模型是表现最好的之一，在比赛后的结果中排名第一。我们提出了一个集成的流水线，可以执行不同的任务，同时共享一个从像素直接学习的通用特征提取基础。 我们的方法仍然可以在几个方面进行改进。(i) 对于定位，我们目前并没有通过整个网络进行反向传播；这样做很可能会提高性能。(ii) 我们使用的是ℓ2损失，而不是直接优化用于衡量性能的交并比（IOU）准则。将损失转换为这个准则应该是可能的，因为IOU仍然是可微的，只要有一些重叠。(iii) 边界框的替代参数化可能有助于去相关化输出，这将有助于网络训练。

# References

1. J. Carreira, F. Li, and C. Sminchisescu. Object recognition by sequential figure-ground ranking. *International journal of computer vision*, 98(3):243–262, 2012.
2. J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation, release 1. http://sminchisescu.ins.uni-bonn.de/code/cpmc/.
3. D. C. Ciresan, J. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, 2012.
4. M. Delakis and C. Garcia. Text detection with convolutional neural networks. In *International Conference on Computer Vision Theory and Applications (VISAPP 2008)*, 2008.
5. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
6. I. Endres and D. Hoiem. Category independent object proposals. In *Computer Vision–ECCV 2010*, pages 575–588. Springer, 2010.
7. C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. in press.
8. C. Garcia and M. Delakis. Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.
9. A. Giusti, D. C. Ciresan, J. Masci, L. M. Gambardella, and J. Schmidhuber. Fast image scanning with deep max-pooling convolutional neural networks. In *International Conference on Image Processing (ICIP)*, 2013.
10. R. Hadsell, P. Sermanet, M. Scoffier, A. Erkan, K. Kavackuoglu, U. Muller, and Y. LeCun. Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26(2):120–144, February 2009.
11. G. Hinton, N. Srivastave, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580, 2012.
12. G. E. Hinton, A. Krizhevsky, and S. D. Wang. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 44–51. Springer Berlin Heidelberg, 2011.
13. V. Jain, J. F. Murray, F. Roth, S. Turaga, V. Zhigulin, K. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung. Supervised learning of image restoration with convolutional networks. In *ICCV’07*.
14. K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *Proc. International Conference on Computer Vision (ICCV’09)*. IEEE, 2009.
15. A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.
16. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems (NIPS 1989)*, volume 2, Denver, CO, 1990. Morgan Kaufman.
17. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.
18. Y. LeCun, F.-J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of CVPR’04*. IEEE Press, 2004.
19. S. Manen, M. Guillaumin, and L. Van Gool. Prime object proposals with randomized prims algorithm. In *International Conference on Computer Vision (ICCV)*, 2013.
20. O. Matan, J. Bromley, C. Burges, J. Denker, L. Jackel, Y. LeCun, E. Pednault, W. Satterfield, C. Stenard, and T. Thompson. Reading handwritten digits: A zip code recognition system. *IEEE Computer*, 25(7):59– 63, July 1992.
21. F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371, September 2005. Special issue on Molecular and Cellular Bioimaging.
22. S. Nowlan and J. Platt. A convolutional neural network hand tracker. pages 901–908, San Mateo, CA, 1995. Morgan Kaufmann.
23. M. Osadchy, Y. LeCun, and M. Miller. Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research*, 8:1197–1215, May 2007.
24. P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In *International Conference on Pattern Recognition (ICPR 2012)*, 2012.
25. P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun. Pedestrian detection with unsupervised multistage feature learning. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR’13)*. IEEE, June 2013.
26. P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *Proceedings of International Joint Conference on Neural Networks (IJCNN’11)*, 2011.
27. G. Taylor, R. Fergus, G. Williams, I. Spiro, and C. Bregler. Pose-sensitive embedding by nonlinear nca regression. In *NIPS*, 2011.
28. G. Taylor, I. Spiro, C. Bregler, and R. Fergus. Learning invarance through imitation. In *CVPR*, 2011.
29. J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.
30. R. Vaillant, C. Monrocq, and Y. LeCun. Original approach for the localisation of objects in images. *IEE Proc on Vision, Image, and Signal Processing*, 141(4):245–250, August 1994.

# Appendix: Additional Model Details

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | Output |
| Layer | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Stage | conv + max | conv + max | conv | conv | conv | conv + max | full | full | full |
| # channels | 96 | 256 | 512 | 512 | 1024 | 1024 | 4096 | 4096 | 1000 |
| Filter size | 7x7 | 7x7 | 3x3 | 3x3 | 3x3 | 3x3 | - | - | - |
| Conv. stride | 2x2 | 1x1 | 1x1 | 1x1 | 1x1 | 1x1 | - | - | - |
| Pooling size | 3x3 | 2x2 | - | - | - | 3x3 | - | - | - |
| Pooling stride | 3x3 | 2x2 | - | - | - | 3x3 | - | - | - |
| Zero-Padding size | - | - | 1x1x1x1 | 1x1x1x1 | 1x1x1x1 | 1x1x1x1 | - | - | - |
| Spatial input size | 221x221 | 36x36 | 15x15 | 15x15 | 15x15 | 15x15 | 5x5 | 1x1 | 1x1 |

Table 3: **Architecture specifics for *accurate* model.** It differs from the *fast* model mainly in the stride of the first convolution, the number of stages and the number of feature maps.

|  |  |  |
| --- | --- | --- |
| model | # parameters (in millions) | # connections (in millions) |
| Krizhevsky | 60 | - |
| *fast* | 145 | 2810 |
| *accurate* | 144 | 5369 |

Table 4: **Number of parameters and connections** for different models.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Input | Layer 5 | Layer 5 | Classifier | Classifier |
| Scale | size | pre-pool | post-pool | map (pre-reshape) | map size |
| 1 | 245x245 | 17x17 | (5x5)x(3x3) | (1x1)x(3x3)x*C* | 3x3x*C* |
| 2 | 281x317 | 20x23 | (6x7)x(3x3) | (2x3)x(3x3)x*C* | 6x9x*C* |
| 3 | 317x389 | 23x29 | (7x9)x(3x3) | (3x5)x(3x3)x*C* | 9x15x*C* |
| 4 | 389x461 | 29x35 | (9x11)x(3x3) | (5x7)x(3x3)x*C* | 15x21x*C* |
| 5 | 425x497 | 32x35 | (10x11)x(3x3) | (6x7)x(3x3)x*C* | 18x24x*C* |
| 6 | 461x569 | 35x44 | (11x14)x(3x3) | (7x10)x(3x3)x*C* | 21x30x*C* |

Table 5: **Spatial dimensions of our multi-scale approach**. 6 different sizes of input images are used, resulting in layer 5 unpooled feature maps of differing spatial resolution (although not indicated in the table, all have 256 feature channels). The (3x3) results from our dense pooling operation with (∆*x,*∆*y*) = {0*,*1*,*2}. See text and Fig. 3 for details for how these are converted into output maps.

表5：我们的多尺度方法的空间维度。使用了6种不同尺寸的输入图像，导致了第5层未池化的特征图具有不同的空间分辨率（尽管表中没有指出，但所有特征图都具有256个特征通道）。(3x3)是我们密集池化操作的结果，其中(∆x,∆y) = {0,1,2}。有关这些如何转换为输出图的详细信息，请参见正文和图3。