Deep Residual Learning for Image Recognition

Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun

Microsoft Research

{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

Abstract

提示：该*PDF*由*SpeedPaper*生成，版权归原文作者所有。翻译内容仅供参考，请仔细鉴别并以原文为准。

查看更多论文翻译与复现代码：`https://github.com/hanknewbird/SpeedPaper`

神经网络的深度增加会使训练变得更加困难。我们提出了一个残差学习框架，以便于训练比以前更深的网络。我们明确地重新构造层，将其视为学习与层输入相关的残差函数，而不是学习无参考的函数。我们提供了全面的经验证据表明，这些残差网络更容易优化，并且可以通过大幅增加深度来提高准确性。在*ImageNet*数据集上，我们评估了深度长达*152*层的残差网络—比*VGG*网络*[41]*深*8*倍，但复杂度仍较低。这些残差网络的集合在*ImageNet* 测试集上取得了*3.57%*的错误率。这一成绩赢得了*ILSVRC 2015*分类任务的第一名。我们还对具有*100*层和*1000*层的*CIFAR-10*进行了分析。

对于许多视觉识别任务来说，表示的深度是至关重要的。仅仅因为我们的表示非常深，我们在*COCO*目标检测数据集上获得了 *28%* 的相对提升。深度残差网络是我们在*ILSVRC*和*COCO 2015*竞赛中的提交基础，我们在*ImageNet*检测、*ImageNet*定位、*COCO*检测和*COCO*分割任务中也获得了第一名。

1. Introduction

深度卷积神经网络[22, 21]已经引领了一系列图像分类的突破[21, 50, 40]。深度网络自然地将低/中/高级特征[50]和分类器以端到端的多层方式整合在一起，并且特征的“级别”可以通过堆叠层的数量（深度）来丰富。最近的证据[41, 44]表明，网络的深度至关重要，并且在具有挑战性的ImageNet数据集[36]上的领先结果[41, 44, 13, 16]都利用了“非常深”的模型，深度为十六[41]至三十[16]。许多其他非常规的视觉识别任务[8, 12, 7, 32, 27]也极大地受益于非常深的模型。

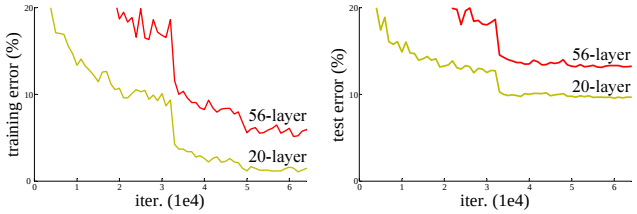

Figure 1. 在 CIFAR-10 数据集上训练错误率（左）和测试错误率（右）分别针对 20 层和 56 层“普通”网络进行比较。更深的网络具有更高的训练错误率，因此测试错误率也更高。在 ImageNet 中出现了类似现象，参见图 4。

在深度的重要性驱动下，一个问题出现了：学习得到更好的网络就像简单地堆叠更多层那样容易吗？回答这个问题的障碍是臭名昭著的梯度消失/梯度爆炸问题[1, 9]，从一开始就妨碍了收敛。然而，通过标准化初始化[23, 9, 37, 13]和中间标准化层[16]，这个问题在很大程度上得到了解决，使得具有数十层的网络可以开始通过随机梯度下降（SGD）和反向传播[22]收敛。

当更深层次的网络能够开始收敛时，一个退化问题浮出水面：随着网络深度的增加，准确度会饱和（这可能不足为奇），然后快速下降。令人意外的是，这种退化不是由过拟合引起的，并且在报告[11, 42]中提到，向一个足够深的模型添加更多层会导致更高的训练误差，并且我们的实验对此进行了彻底验证。图1展示了一个典型例子。

（训练精度的）退化表明，并非所有系统都同样

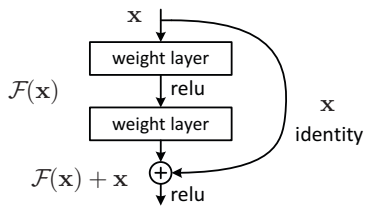

Figure 2. Residual learning: a building block.

容易优化。让我们考虑一个更浅的架构及其加入更多层的更深层次对应物。对更深层次模型存在构建的解决方案：添加的层是恒等映射，其他层从学习的更浅模型中复制而来。这个构建的解决方案的存在表明，更深的模型应该产生的训练误差不应高于更浅的对应模型。但是实验表明当下我们手头上的求解器无法找到与构建的解决方案同样好或更好的解决方案（或者无法在合理的时间内做到）。

在本文中，我们通过引入深度残差学习框架来解决退化问题。我们不再希望每几个堆叠的层直接拟合一个期望的底层映射，而是显式地让这些层拟合一个残差映射。形式上，将期望的底层映射记作$\mathcal{H}(\mathbf{x})$，我们使堆叠的非线性层拟合另外一个映射$\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$。原始映射被重新表述为$\mathcal{F}(\mathbf{x}) + \mathbf{x}$。我们假设优化残差映射比优化原始未引用的映射更容易。极端情况下，如果恒等映射是最优的，将残差推向零比通过一系列非线性层去拟合恒等映射更容易。

$\mathcal{F}(\mathbf{x}) + \mathbf{x}$的形式可以通过带“快捷连接”的前馈神经网络实现（图2）。快捷连接[2, 34, 49]是跳过一个或多个层的连接。在我们的情况中，快捷连接简单执行恒等映射，并将它们的输出添加到堆叠层的输出中（图2）。恒等快捷连接既不增加额外参数也不增加计算复杂度。整个网络仍可以通过使用SGD和反向传播端对端训练，并可以使用常见的库（例如Caffe [19]）轻松实现而无需修改求解器。

我们在ImageNet上进行了全面的实验，展示降级问题并评估我们的方法。我们发现：1）我们的极深残差网络易于优化，但对应的“普通”网络（简单堆叠层的网络）在深度增加时训练误差较高；2）我们的深度残差网络能够轻松地从大幅增加深度中获得精度提升，产生的结果明显优于之前的网络。

类似的现象也在CIFAR-10数据集上得到展示，表明优化困难和我们方法的影响不仅仅适用于特定数据集。我们在该数据集上成功训练的模型具有超过100层，并探索了具有超过1000层的模型。

在ImageNet分类数据集上，我们通过极深残差网络获得了出色的结果。我们的152层残差网络是迄今为止在ImageNet上展示的最深的网络，而复杂度仍低于VGG网络。我们的集成模型在ImageNet测试集上的top-5误差为**3.57%**，并获得了ILSVRC 2015分类竞赛的第一名。这些极深的表示也在其他识别任务上具有出色的泛化性能，并且使我们在ILSVRC＆COCO 2015竞赛中进一步获得了ImageNet检测、ImageNet定位、COCO检测和COCO分割的第一名。这些有力的证据表明残差学习原则是通用的，并且我们期望它能够应用于其他视觉和非视觉问题。

2. Related Work

残差表示。 在图像识别中，VLAD [18] 是一种表示形式，它通过相对于字典的残差向量进行编码，而 Fisher Vector [30] 可以被形式化为 VLAD 的概率版本 [18]。它们都是针对图像检索和分类的强大浅层表示 [4, 48]。对于矢量量化，显示编码残差向量 [17] 相较于编码原始向量会更有效。

在低层次视觉和计算机图形学中，为了解决偏微分方程（PDEs），广泛采用的多重网格方法 [3] 将系统重新表述为多个尺度上的子问题，其中每个子问题负责在更粗糙和更精细的尺度之间的残差解。多重网格的一种替代方法是分层基础预处理 [45, 46]，它依赖于代表两个尺度之间残差向量的变量。研究表明 [3, 45, 46]，这些求解器的收敛速度比不了解解决方案残差性质的标准求解器快得多。这些方法表明，良好的重新表述或预处理可以简化优化过程。

快捷连接。 长期以来，已研究了导致快捷连接的实践和理论[2, 34, 49]。早期训练多层感知器（MLPs）的一种实践是添加一个从网络输入到输出的线性层[34, 49]。在[44, 24]中，几个中间层直接连接到辅助分类器，以解决梯度消失/爆炸问题。在[39, 38, 31, 47]的论文中，提出了一些关于通过快捷连接实现中心化层响应、梯度和传播错误的方法。在[44]中，一个“inception”层由一个快捷分支和几个更深的分支组成。

与我们的工作同步进行，“高速公路网络”[42, 43]引入了带有门控函数的快捷连接[15]。这些门是数据相关的，并具有参数，与我们的无参数身份快捷连接形

成对比。当门控式快捷连接“关闭”（接近零）时，高速公路网络中的层表示非残差函数。相反，我们的表述始终学习残差函数；我们的身份快捷连接永远不会关闭，所有信息始终被传递，需要学习额外的残差函数。此外，高速公路网络并未展示出在深度极大增加时的准确性提高（例如，超过100层）。

3. Deep Residual Learning

3.1. Residual Learning

考虑$\mathcal{H}(\mathbf{x})$作为一个基础映射，由一些堆叠的层（不一定是整个网络）拟合，其中$\mathbf{x}$表示输入到这些层中的第一个。如果假设多个非线性层可以渐近地逼近复杂函数[1]，那么等同于假设它们可以渐近地逼近残差函数，即$\mathcal{H}(\mathbf{x})-\mathbf{x}$（假定输入和输出具有相同维度）。

因此，与其期望堆叠的层逼近$\mathcal{H}(\mathbf{x})$，我们明确地让这些层逼近一个残差函数$\mathcal{F}(\mathbf{x}):=\mathcal{H}(\mathbf{x})-\mathbf{x}$。原始函数因此变为$\mathcal{F}(\mathbf{x})+\mathbf{x}$。尽管这两种形式都应该能够渐近地逼近所期望的函数（如假设的那样），但学习的难易程度可能有所不同。

这种重新表述是受到关于退化问题的反直觉现象的启发（图 1，左侧）。正如我们在引言中讨论的那样，如果额外的层可以构建为恒等映射，那么一个更深的模型的训练误差不应该大于其较浅的对应物。退化问题表明，求解器可能会在通过多个非线性层逼近恒等映射时遇到困难。通过残差学习的重新表述，如果恒等映射是最优选择，那么求解器可能会简单地使多个非线性层的权重趋近于零以逼近恒等映射。

在实际情况中，恒等映射可能不是最佳选择，但我们的重新表述可能有助于预处理问题。如果最佳函数更接近于一个恒等映射而不是零映射，那么对求解器来说，找到与恒等映射相关的扰动应该比学习一个新函数更容易。我们通过实验（图 7）显示，一般学得的残差函数具有较小的响应，表明恒等映射提供了合理的预处理。

3.2. Identity Mapping by Shortcuts

我们在每几个堆叠层中采用残差学习。图 2 显示了一个构建块。形式上，在本文中，我们将一个构建块定义为：

$$\mathbf{y}=\mathcal{F}(\mathbf{x},\{W_i\})+\mathbf{x}. \tag{1}$$

在这里，$\mathbf{x}$ 和 $\mathbf{y}$ 是所考虑层的输入和输出向量。函数 $\mathcal{F}(\mathbf{x},\{W_i\})$ 代表要学习的残差映射。对于图 2 中有两层的例子，$\mathcal{F}=W_2\sigma(W_1\mathbf{x})$，其中 σ 表示ReLU [29]，为简化符号，省略了偏差。操作 $\mathcal{F}+\mathbf{x}$ 通过快捷连接和逐元素加法执行。我们在加法之后采用第二个非线性（即，$\sigma(\mathbf{y})$，见图 2）。

式 (1) 中的快捷连接既不引入额外的参数，也不增加计算复杂性。这不仅在实践中具有吸引力，还在纯网络和残差网络之间的比较中至关重要。我们可以公平地比较同时具有相同参数数量、深度、宽度和计算成本的纯/残差网络（除了可忽略的逐元素加法）。

在式 (1) 中，$\mathbf{x}$ 和 $\mathcal{F}$ 的维度必须相等。如果不是这种情况（例如，更改输入/输出通道时），我们可以通过快捷连接执行线性投影 W_s 来匹配维度：

$$\mathbf{y}=\mathcal{F}(\mathbf{x},\{W_i\})+W_s\mathbf{x}. \tag{2}$$

我们也可以在公式(1)中使用一个方阵W_s。但我们将通过实验表明，恒等映射足以解决退化问题且经济实惠，因此只有在匹配维度时才使用W_s。

残差函数$\mathcal{F}$的形式是灵活的。本文的实验涉及到一个具有两到三个层次的函数$\mathcal{F}$（如图 5所示），更多层次也是可能的。但如果$\mathcal{F}$只有一个单层，公式(1)类似于一个线性层：$\mathbf{y}=W_1\mathbf{x}+\mathbf{x}$，我们并未观察到其优势。

我们还指出，尽管以上符号是为了简单起见而涉及全连接层，但它们也适用于卷积层。函数$\mathcal{F}(\mathbf{x},\{W_i\})$可以表示多个卷积层。对两个特征图进行逐元素相加，逐通道执行。

3.3. Network Architectures

我们已经测试了各种普通/残差网络，并观察到了一致的现象。为了提供讨论的实例，我们描述了两个用于ImageNet的模型如下。

简单网络. 我们的简单基线模型（见图 3，中）主要受 VGG 网络哲学的启发[41]（见图 3，左）。卷积层大多采用 3×3 的滤波器，并遵循两个简单设计原则：（i）对于相同的输出特征图大小，各层拥有相同数量的滤波器；以及（ii）如果特征图大小减半，滤波器数量

[1]然而,这一假设仍是一个未决问题。见[28]。

翻倍以保持每层的时间复杂度。我们通过步幅为 2 的卷积层直接进行下采样。网络以一个全局平均池化层和一个具有 softmax 的 1000 路全连接层结束。如图 3（中）所示，加权层的总数为 34 层。

值得注意的是，我们的模型比 VGG 网络[41]（见图 3，左）拥有更少的滤波器和更低的复杂度。我们的 34 层基线模型的浮点运算总量为 36 亿次（乘加运算），仅为 VGG-19（196 亿次）的 18%。

4. 残差网络。

基于上述普通网络，在其中插入快捷连接（图 3，右），使网络转变为其残差版本。当输入和输出具有相同维度时，可以直接使用恒等快捷方式（公式(1)），如图 3 中的实线快捷方式。当维度增加时（图 3 中的虚线快捷方式），我们考虑两个选项： (A) 快捷方式仍执行恒等映射，为增加的维度填充额外的零条目。该选项不引入额外参数； (B) 使用公式(2) is used to match dimensions (done by 1×1 convolutions). For both options, when the shortcuts go across feature maps of two sizes, they are performed with a stride of 2.

4.1. Implementation

我们对ImageNet的实现遵循了[21, 41]中的做法。图像的调整大小在规模增强中随机采样其较短边在$[256, 480]$之间[41]。从图像或其水平翻转中随机采样出一个224×224的裁剪图像，并减去每个像素的均值[21]。采用了[21]中的标准颜色增强。我们在每个卷积层之后和激活函数之前采用批归一化（Batch Normalization, BN）[16]，遵循[16]的做法。我们将权重初始化为[13]中的方法，并从头开始训练所有的纯粹/残差网络。我们使用的是带有256个样本的小批量随机梯度下降（Stochastic Gradient Descent, SGD）。学习率从0.1开始，当误差停滞时除以10，模型训练时间长达60×10^4次迭代。我们采用了0.0001的权重衰减和0.9的动量。我们不使用丢弃（Dropout）[14]，遵循[16]的惯例。

在测试阶段，为进行比较研究，我们采用了标准的10次裁剪测试[21]。为了获得最佳结果，我们采用了[41, 13]中的全卷积形式，并对多个尺度的得分进行平均处理（图像调整大小使较短边处于$\{224, 256, 384, 480, 640\}$）。

5. Experiments

5.1. ImageNet Classification

我们在包含1000个类别的ImageNet 2012分类数据集[36]上评估了我们的方法。模型在128万张训练图片上进行训练，并在5万张验证图片上进行评估。我们还在测试服务器报告的10万张测试图片上得到最终结果。我们评估了top-1和top-5错误率。

简单网络。 我们首先评估了18层和34层的简单网络。34层的简单网络如图 3（中）所示。18层的简单网络具有类似的结构。详细的网络架构请参见表 1。

表 2中的结果显示，更深的34层简单网络比较浅的18层简单网络有更高的验证错误。为了揭示原因，我们在图 4（左）中比较它们在训练过程中的训练/验证错误。我们观察到退化问题 - 34层的简单网络在整个训练过程中都有更高的训练错误，即使18层简单网络的解空间是34层网络的子空间。

我们认为，这种优化困难不太可能是由消失的梯度引起的。这些普通网络是用BN [16]进行训练的，这确保了前向传播信号具有非零方差。我们还验证了通过BN传播的反向梯度具有健康的范数。因此，前向和后向信号都不会消失。实际上，34层的普通网络仍然能够实现竞争力的准确性（表 3），表明求解器在某种程度上有效。我们推测深层的普通网络可能具有指数低的收敛速率，这会影响训练误差的降低[2]。这种优化困难的原因将在今后研究中探讨。

残差网络。

接下来，我们评估了18层和34层的残差网络（*ResNets*）。基线架构与上文的普通网络相同，只是在每一对3×3的滤波器中添加了一条快捷连接，如图 3（右侧）所示。在第一个比较中（表 2 和图 4 右侧），我们对所有快捷连接使用标识映射，对增加维度使用零填充（选项A）。因此，与普通网络相比，它们没有额外的参数。

从表 2 和图 4 中，我们有三个主要观察。首先，在残差学习中，情况与之前相反——34层ResNet优于18层ResNet（提高了2.8%）。更重要的

[2]我们进行了更多的训练迭代（3×）,仍然观察到了退化问题,这表明通过简单增加迭代次数不能解决这个问题。

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1. ImageNet的架构。构建模块如方括号所示（另见图 5），并显示堆叠的模块数。通过conv3_1、conv4_1和conv5_1执行降采样，步幅为2。

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Table 2. 在ImageNet验证中的Top-1错误率（%，10-crop测试）。与它们的普通对应模型相比，ResNets没有额外的参数。图 4展示了训练过程。

是，34层ResNet表现出较低的训练误差，并且可以泛化至验证数据。这表明在这种情况下成功地解决了退化问题，并且我们通过增加深度获得了准确度的提升。

其次，与普通对应物相比，34层ResNet将top-1误差降低了3.5%（表 2），这是由于成功减少了训练误差（图 4 右侧与左侧的对比）。这种比较验证了残差学习对极深系统的有效性。

最后，我们还注意到，18层的普通/残差网络在准确度上相当（表 2），但18层ResNet收敛速度更快（图 4 右侧与左侧的对比）。当网络“不是过于深”时（这里是18层），当前的随机梯度下降求解器仍然能够找到普通网络的良好解决方案。在这种情况下，ResNet通过在早期阶段提供更快的收敛来简化优化过程。

身份*vs.*投影快捷方式。 我们已经证明，无参数的身份快捷方式有助于训练。接下来我们将研究投影快捷方式（式（2））。在表 3 中，我们比较了三种选项：（A）对于增加维度，使用零填充快捷方式，并且所有快捷方式均无参数（与表 2 和图 4 右侧相同）；（B）对于增加维度，使用投影快捷方式，其余快捷方式为身份；以及（C）所有快捷方式均为投影。

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Table 3. 在ImageNet验证集上的错误率（%，**10-crop** 测试）。VGG-16是基于我们的测试。ResNet-50/101/152采用仅利用投影以增加维度的选项B。

表 3 显示，这三种选项均比简单对应项明显更好。B 稍微优于 A。我们认为这是因为在 A 中的零填充维度确实没有残差学习。C 稍微好于 B，我们将此归因于许多（十三个）投影快捷方式引入的额外参数。但 A/B/C 之间的微小差异表明，投影快捷方式并非解决退化问题的必要条件。因此我们在本文的其余部分中不使用选项 C，以减少内存/时间复杂度和模型大小。对于不增加引入的瓶颈结构的复杂性，身份快捷方式尤为重要。

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4. 在ImageNet验证集上的**单模型**结果的错误率（%，†除外，报告在测试集上）。

method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Table 5. **集成**的错误率（%）。前五个错误在ImageNet的测试集上，由测试服务器报告。

更深的瓶颈结构。

接下来我们描述了适用于ImageNet的更深层的网络。由于我们担心我们所能承受的训练时间，我们将构建块修改为瓶颈设计[3]。对于每个残差函数$\mathcal{F}$，我们使用一堆3层而不是2层（图 5）。这三层是1×1、3×3和1×1的卷积层，其中1×1层负责减少然后增加（恢复）维度，使3×3层成为具有较小输入/输出维度的瓶颈。图 5展示了一个例子，其中两种设计具有类似的时间复杂性。

对于瓶颈结构，无参数的身份快捷方式特别重要。如果图 5（右侧）中的身份快捷方式被替换为投影，可以证明时间复杂性和模型大小都会增加一倍，因为快捷方式连接到两个高维端点。因此，对于瓶颈设计，身份快捷方式可导致更高效的模型。

50层ResNet: 我们用这个3层瓶颈块替换34层网络中的每个2层块，得到一个50层ResNet（表 1）。我们选择选项B来增加维度。该模型有38亿次浮点运算（FLOPs）。

101层 和152层ResNets: 我们通过使用更多的3层块构建了101层和152层的ResNets（表 1）。值得注意的是，尽管深度明显增加，152层的ResNet（113亿FLOPs）仍然比VGG-16/19网络（153亿/196亿FLOPs）具有更低的复杂度。

50/101/152层ResNets比34层的准确性有显着的差距（表 3 和表 4）。我们没有观察到退化问题，因此从明显增加的深度中获得了显著的准确性提升。深度的优势体现在所有评估指标上（表 3 和表 4）。

与最先进方法的比较

在表 4 中，我们将与先前最佳的单模型结果进行比较。我们基准的34层ResNets取得了非常有竞争力的准确性。我们的152层ResNet在单模型下的前五验证错误率为4.49%。这一单模型结果胜过了所有先前的集成结果（表 5）。我们将六个不同深度的模型组合成一个集成（在提交时只有两个152层模型）。这在测试集上导致了**3.57%** 的前五错误率（表 5）。这个条目赢得了*2015*年*ILSVRC*比赛第一名。

5.2. CIFAR-10 and Analysis

我们在CIFAR-10数据集[20]上进行了更多研究，该数据集包含10个类别中的50,000张训练图像和10,000张测试图像。我们展示了在训练集上训练并在测试集上评估的实验。

我们的重点在于极深网络的行为，而不是追求最先进的结果，因此我们有意使用简单的结构，如下所示。

纯粹/残差结构遵循图 3（中/右）所示的形式。网络输入是32×32的图像，减去每个像素的均值。第一层是3×3的卷积。然后我们在分别为$\{32, 16, 8\}$大小的特征图上使用一堆$6n$层，其中每个特征图大小有$2n$层的卷积。滤波器的数量分别为$\{16, 32, 64\}$。下采样通过步长为2的卷积实现。网络以全局平均池化、一个10路全连接层和softmax结束。总共有$6n$+2堆叠的加权层。以下表总结了架构：

[3]更深层的非瓶颈ResNet（例如，图 5 左侧）也通过增加深度（如在CIFAR-10中所示）获得准确性，但不像瓶颈ResNets那样经济。因此，瓶颈设计的使用主要是出于实际考虑。我们进一步指出，普通网络的退化问题也出现在瓶颈设计中。

method			error (%)
Maxout [10]			9.38
NIN [25]			8.81
DSN [24]			8.22
	# layers	# params	
FitNet [35]	19	2.5M	8.39
Highway [42, 43]	19	2.3M	7.54 (7.72±0.16)
Highway [42, 43]	32	1.25M	8.80
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43** (6.61±0.16)
ResNet	1202	19.4M	7.93

Table 6. 在**CIFAR-10**测试集上的分类错误。所有方法均采用数据增强。对于ResNet-110，我们运行了5次，并显示“最好（均值±标准差）”如[43]中所示。

output map size	32×32	16×16	8×8
# layers	1+2n	2n	2n
# filters	16	32	64

当使用快捷连接时，它们连接到3×3层的一对（总共3n个快捷连接）。在这个数据集中，我们在所有情况下使用恒等快捷连接（即，选项A），因此我们的残差模型与普通对应模型具有完全相同的深度、宽度和参数数量。

我们使用了权重衰减为0.0001和动量为0.9，并采用了[13]中的权重初始化和[16]中的BN，但没有使用dropout。这些模型在两个GPU上以大小为128的mini-batch进行训练。我们从学习率0.1开始，分别在32k和48k次迭代时除以10，训练在64k次迭代时终止，这是在45k/5k的训练/验证集拆分上确定的。我们遵循[24]中的简单数据增强方法进行训练：在每一侧填充4个像素，并从填充图像或其水平翻转中随机选取一个32×32的裁剪图像。在测试时，我们只评估原始32×32图像的单个视图。

我们比较了$n = \{3, 5, 7, 9\}$，导致了20、32、44和56层网络。图 6（左）展示了普通网络的行为。深层普通网络在增加深度时遇到了训练错误增加的问题。这种现象类似于在ImageNet（图 4，左）和MNIST（见[42]）上的情况，表明这种优化困难是一个基本问题。

图 6（中）展示了ResNets的行为。和ImageNet情况相似（图 4，右），我们的ResNets设法克服了优化困难，并在深度增加时表现出准确性提高。

我们进一步探索了导致一个110层ResNet的$n = 18$。在这种情况下，我们发现初始学习率0.1稍微过大，无法开始收敛[4]。因此，我们使用0.01来热身训练，直到训练错误率低于80%（大约400次迭代），然后恢复到0.1继续训练。其余学习计划如前所述。这个110层网络收敛良好（图 6，中）。它的参数比其他深且细的网络（如FitNet [35]和Highway [42]）要少（表 6），但是在最新的研究成果中表现出色（6.43%，表 6）。

分析层响应。 图 7 显示了层响应的标准差（std）。这些响应是每个3×3层的输出，在BN之后，非线性激活函数（ReLU/addition）之前。对于ResNets，这种分析揭示了残差函数的响应强度。图 7 表明，与普通对应模型相比，ResNets的响应通常更小。这些结果支持我们的基本动机（第3.1节），即残差函数可能总体上比非残差函数更接近零。我们还注意到，更深的ResNet的响应幅度更小，这可以从图 7 中ResNet-20、56和110之间的比较中看出。当层数更多时，ResNets的单个层倾向于修改信号更少。

6. 探索超过1000层。

我们探索了一个超过1000层的极深模型。我们设置 $n = 200$，形成一个1202层的网络，按照以上描述进行训练。我们的方法显示没有优化困难，这个10^3层网络能够实现训练误差¡0.1%（图 6，右侧）。其测试误差仍然相当不错（7.93%，表 6）。

然而，在这种极深模型上仍然存在一些问题。这个1202层网络的测试结果比我们的110层网络差，尽管它们的训练误差相似。我们认为这是由于过拟合产生的。对于这个小数据集来说，1202层网络可能过于庞大（19.4M）。为了获得最佳结果，我们采用了强大的正则化方法，如maxout [10] 或dropout [14] ([10, 25, 24,

[4]使用初始学习率0.1后,几个周期后开始收敛（¡90%的错误率）,但仍然达到类似的准确性。

training data	07+12	07++12
test data	VOC 07 test	VOC 12 test
VGG-16	73.2	70.4
ResNet-101	**76.4**	**73.8**

Table 7. 在PASCAL VOC 2007/2012测试集上，使用**基准** Faster R-CNN 进行目标检测mAP（%）。更好的结果请参见表10和11。

metric	mAP@.5	mAP@[.5, .95]
VGG-16	41.5	21.2
ResNet-101	**48.4**	**27.2**

Table 8. 在COCO验证集上，使用**基准** Faster R-CNN 进行目标检测 mAP（百分比）。更好结果请参阅表9。

35])。在本文中，我们没有使用maxout/dropout，而仅通过设计采用了深且狭窄的架构来实施正则化，不会分散对优化困难的关注。但结合更强的正则化可能会改善结果，这是我们将来会研究的部分。

6.1. Object Detection on PASCAL and MS COCO

以下是一篇英语学术论文中的一节。

我们的方法在其他识别任务上具有良好的泛化性能。表 7 和 8 展示了在PASCAL VOC 2007和2012[5]以及COCO[26]上的目标检测基线结果。我们采用*Faster R-CNN* [32] 作为检测方法。在这里，我们关注用ResNet-101替换VGG-16[41] with ResNet-101. The detection implementation (see appendix) of using both models is the same, so the gains can only be attributed to better networks. Most remarkably, on the challenging COCO dataset we obtain a 6.0% increase in COCO's standard metric (mAP@[.5, .95]), which is a 28% relative improvement. This gain is solely due to the learned representations.

Based on deep residual nets, we won the 1st places in several tracks in ILSVRC & COCO 2015 competitions: ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation. The details are in the appendix.

References

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[2] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[3] W. L. Briggs, S. F. McCormick, et al. *A Multigrid Tutorial*. Siam, 2000.

[4] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, pages 303–338, 2010.

[6] S. Gidaris and N. Komodakis. Object detection via a multi-region & semantic segmentation-aware cnn model. In *ICCV*, 2015.

[7] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[10] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *arXiv:1302.4389*, 2013.

[11] K. He and J. Sun. Convolutional neural networks at constrained time cost. In *CVPR*, 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[17] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *TPAMI*, 33, 2011.

[18] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *TPAMI*, 2012.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.

[20] A. Krizhevsky. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

[21] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

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

[23] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. Efficient backprop. In *Neural Networks: Tricks of the Trade*, pages 9–50. Springer, 1998.

[24] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. *arXiv:1409.5185*, 2014.

[25] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv:1312.4400*, 2013.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*. 2014.

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[28] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. In *NIPS*, 2014.

[29] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[30] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

[31] T. Raiko, H. Valpola, and Y. LeCun. Deep learning made easier by linear transformations in perceptrons. In *AISTATS*, 2012.

[32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[33] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. *arXiv:1504.06066*, 2015.

[34] B. D. Ripley. *Pattern recognition and neural networks*. Cambridge university press, 1996.

[35] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv:1409.0575*, 2014.

[37] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv:1312.6120*, 2013.

[38] N. N. Schraudolph. Accelerated gradient descent by factor-centering decomposition. Technical report, 1998.

[39] N. N. Schraudolph. Centering neural network gradient factors. In *Neural Networks: Tricks of the Trade*, pages 207–226. Springer, 1998.

[40] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014.

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[42] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv:1505.00387*, 2015.

[43] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. *1507.06228*, 2015.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[45] R. Szeliski. Fast surface interpolation using hierarchical basis functions. *TPAMI*, 1990.

[46] R. Szeliski. Locally adapted hierarchical basis preconditioning. In *SIGGRAPH*, 2006.

[47] T. Vatanen, T. Raiko, H. Valpola, and Y. LeCun. Pushing stochastic gradient towards second-order methods–backpropagation learning with transformations in nonlinearities. In *Neural Information Processing*, 2013.

[48] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[49] W. Venables and B. Ripley. Modern applied statistics with s-plus. 1999.

[50] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In *ECCV*, 2014.

A. Object Detection Baselines

在本节中，我们介绍基于基准 Faster R-CNN [32] 系统的检测方法。这些模型是通过 ImageNet 分类模型进行初始化，然后在目标检测数据上进行微调。我们在 ILSVRC COCO 2015 检测竞赛时尝试了使用 ResNet-50/101。

与 [32] 中使用的 VGG-16 不同，我们的 ResNet 没有隐藏的全连接层。我们采用了“Networks on Conv feature maps”（NoC）[33] 这一概念来解决这个问题。我们使用那些在图像上步幅不大于16像素的层（如 conv1、conv2_x、conv3_x 和 conv4_x，在 ResNet-101 中共有 91 个卷积层）计算全图共享的卷积特征图（参见表 1）。我们将这些层视为类似于 VGG-16 中的 13 个卷积层，通过这种方式，ResNet 和 VGG-16 都具有相同总步幅（16像素）的卷积特征图。这些层被区域建议网络（RPN，生成 300 个建议）[32] 和 Fast R-CNN 检测网络[7]所共享。在 conv5_1 之前执行 RoI pooling[7]。在此 RoI-pooled 特征上，每个区域都采用 conv5_x 及更高层，担任 VGG-16 的全连接层角色。最终的分类层被两个兄弟层（分类层和框回归[7]）替代。

对于 BN 层的使用，在预训练之后，我们在 ImageNet 训练集上计算每一层的 BN 统计量（均值和方差）。然后，在目标检测期间，BN 层被固定。这样一来，BN 层变成具有恒定偏移和比例的线性激活，并且 BN 统计量在微调过程中不会更新。我们主要固定 BN 层是为了减少 Faster R-CNN 训练时的内存消耗。

PASCAL VOC

根据[7, 32]，对于PASCAL VOC 2007的测试集，我们使用VOC 2007中的5k *trainval* 图像和VOC 2012中的16k *trainval* 图像进行训练（“07+12”）。对于PASCAL VOC 2012的测试集，我们使用VOC 2007中的10k *trainval+test* 图像和VOC 2012中的16k *trainval* 图像进行训练（“07++12”）。训练Faster R-CNN的超参数与[32]中相同。表 7显示了结果。ResNet-101使mAP提高了大于3%，这一增益完全是由于ResNet学习到的改进特征。

MS COCO

MS COCO数据集涉及80个物体类别[26]。我们评估PASCAL VOC指标（mAP @ IoU = 0.5）和标准COCO指标（mAP @ IoU = .5:.05:.95）。我们使用训练集上的80k张图像进行训练，使用验证集上的40k张图像进行评估。我们针对COCO的检测系统与PASCAL VOC的系统类似。我们使用8个GPU进行COCO模型的训练，因此RPN步骤的小批量大小为8张图像（即，每个GPU 1张），Fast R-CNN步骤的小批量大小为16张图像。RPN步骤和Fast R-CNN步骤均进行了240k次迭代训练，学习率为0.001，然后进行了80k次迭代训练，学习率为0.0001。

表 8显示了MS COCO验证集的结果。ResNet-101的mAP@[.5, .95]比VGG-16提高了6%，相对改善了28%，这完全是由更好的网络学习到的特征贡献的。值得注意的是，mAP@[.5, .95]的绝对增长（6.0%）几乎与mAP@.5（6.9%）一样大。这表明更深的网络能够提高识别和定位的能力。

B. Object Detection Improvements

为了完整起见，我们报告了为比赛做出的改进。这些改进基于深度特征，因此应该受益于残差学习。

MS COCO

盒子细化。 我们的盒子细化部分地遵循了[6]中的迭代定位方法。在Faster R-CNN中，最终的输出是一个经过回归的盒子，与其提议的盒子不同。因此，在推断过程中，我们从回归的盒子中提取一个新的特征，得到一个新的分类得分和一个新的回归盒子。我们将这300个新的预测结果与原始的300个预测结果相结合。在预测框的并集上应用IoU阈值为0.3的非最大抑制(NMS)[8]，然后进行盒子投票[6]。盒子细化将平均精度均值（mAP）提高了约2个点（表9）。

全局背景。 我们在快速R-CNN步骤中结合全局背景。给定完整图像的卷积特征图，我们通过全局空间金字塔池化[12]（采用“单级”金字塔）来池化特征，可以将其实现为以整个图像的边界框作为感兴趣区域的“RoI”池化。这个池化特征被送入RoI之后的层以获得全局背景特征。全局特征与原始区域特征拼接后，经过兄弟分类和框回归层。这种新结构被端到端地训练。全局背景可以使mAP@0.5提高约1个点（表9）。

多尺度测试。 在上述实验中，所有结果均通过单尺度训练/测试获得，如[32]中所示，其中图像的较短边长为$s = 600$像素。多尺度训练/测试已经在[12, 7]中发展出来，通过从特征金字塔中选择一个尺度，以及

training data	COCO train		COCO trainval	
test data	COCO val		COCO test-dev	
mAP	@.5	@[.5, .95]	@.5	@[.5, .95]
baseline Faster R-CNN (VGG-16)	41.5	21.2		
baseline Faster R-CNN (ResNet-101)	48.4	27.2		
+box refinement	49.9	29.9		
+context	51.1	30.0	53.3	32.2
+multi-scale testing	53.8	32.5	**55.7**	**34.9**
ensemble			**59.0**	**37.4**

Table 9. 使用Faster R-CNN和ResNet-101在MS COCO数据集上改进目标检测。

system	net	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
baseline	VGG-16	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
baseline	ResNet-101	07+12	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	**89.8**	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.3	72.0
baseline+++	ResNet-101	COCO+07+12	**85.6**	**90.0**	**89.6**	**87.8**	**80.8**	**76.1**	**89.9**	**89.9**	89.6	**75.5**	**90.0**	**80.7**	**89.6**	**90.3**	**89.1**	**88.7**	**65.4**	**88.1**	**85.6**	**89.0**	**86.8**

Table 10. 在PASCAL VOC 2007测试集上的检测结果。基准是Faster R-CNN系统。系统“baseline+++”包括表9中的边界框细化、上下文和多尺度测试。

system	net	data	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
baseline	VGG-16	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
baseline	ResNet-101	07++12	73.8	86.5	81.6	77.2	58.0	51.0	78.6	76.6	93.2	48.6	80.4	59.0	92.1	85.3	84.8	80.7	48.1	77.3	66.5	84.7	65.6
baseline+++	ResNet-101	COCO+07++12	**83.8**	**92.1**	**88.4**	**84.8**	**75.9**	**71.4**	**86.3**	**87.8**	**94.2**	**66.8**	**89.4**	**69.2**	**93.9**	**91.9**	**90.9**	**89.6**	**67.9**	**88.2**	**76.8**	**90.3**	**80.0**

Table 11. 在PASCAL VOC 2012测试集（`http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=4`）上的检测结果。基准是Faster R-CNN系统。表 9 中的系统“baseline+++”包括边界框细化、上下文和多尺度测试。

在[33]中通过使用maxout层。在我们目前的实现中，我们按照[33]进行了多尺度 测试；由于时间有限，我们尚未进行多尺度训练。此外，我们仅对Fast R-CNN步骤进行了多尺度测试（尚未对RPN步骤进行）。使用训练好的模型，我们在图像金字塔上计算了卷积特征图，其中图像的较短边长为$s \in \{200, 400, 600, 800, 1000\}$。我们按照[33]从金字塔中选择了两个相邻尺度。 RoI池化和后续层在这两个尺度的特征图上执行[33]，并通过maxout进行合并，如[33]所示。多尺度测试使mAP提高了超过2个百分点（表 9）。

使用验证数据。接下来，我们使用80k+40k的trainval集进行训练，使用20k的test-dev集进行评估。 test-dev集没有公开可用的标准答案，结果是通过评估服务器报告的。在这种设置下，结果是mAP@.5为55.7%，mAP@[.5, .95]为34.9%（表 9）。这是我们的单模型结果。

集成。 在 Faster R-CNN 中，系统被设计为学习区域提议和物体分类器，因此可以使用集成来提升这两个任务。我们使用一个集成来提出区域，提议的并集由一个每个区域分类器的集成处理。表 9 显示了基于 3 个网络集成的结果。 mAP 在测试开发集上分别为 59.0% 和 37.4%。 这个结果在 *COCO 2015* 的检测任务中获得了第一名。

PASCAL VOC

基于上述模型，我们重新审视了PASCAL VOC数据集。我们使用在COCO数据集上的单一模型（在表9中为55.7% mAP@.5）对PASCAL VOC数据集进行微调。我们还采用了边界框细化、上下文和多尺度测试的改进方法。通过这样做，我们在PASCAL VOC 2007上达到了85.6%的mAP（表10），在PASCAL VOC

	val2	test
GoogLeNet [44] (ILSVRC'14)	-	43.9
our single model (ILSVRC'15)	60.5	58.8
our ensemble (ILSVRC'15)	**63.6**	**62.1**

Table 12. 我们在ImageNet检测数据集上的结果（mAP，%）。我们的检测系统是使用ResNet-101的Faster R-CNN [32]，其中包括表9中的改进。

LOC method	LOC network	testing	LOC error on GT CLS	classification network	top-5 LOC error on predicted CLS
VGG's [41]	VGG-16	1-crop	33.1 [41]		
RPN	ResNet-101	1-crop	13.3		
RPN	ResNet-101	dense	11.7		
RPN	ResNet-101	dense		ResNet-101	14.4
RPN+RCNN	ResNet-101	dense		ResNet-101	**10.6**
RPN+RCNN	ensemble	dense		ensemble	**8.9**

Table 13. 在ImageNet验证集上的定位错误率（%）。在"LOC error on GT class"一栏中（[41]），使用了真实类别。在"testing"一栏中，"1-crop"表示在一个中心裁剪的224×224像素上进行测试，"dense"表示密集（全卷积）和多尺度测试。

2012上达到了83.8%（表11）[5]。PASCAL VOC 2012的结果比先前的最先进结果高了10个百分点[6]。

ImageNet检测

ImageNet检测（DET）任务涉及200个目标类别。准确性通过mAP@.5进行评估。我们针对ImageNet DET的目标检测算法与表9中针对MS COCO的算法相同。这些网络是在1000类别的ImageNet分类集上进行预训练，并在DET数据上进行微调。我们按照[8]将验证集分成两部分（val1/val2）。我们使用DET训练集和val1集对检测模型进行微调，val2集用于验证。我们不使用其他ILSVRC 2015数据。我们的单一模型使用ResNet-101在DET测试集上获得58.8%的mAP，我们的3个模型集成在DET测试集上获得62.1%的mAP（表12）。这一结果在*ILSVRC 2015*的*ImageNet*检测任务中获得第一名，超过第二名**8.5个百分点**（绝对值）。

C. ImageNet Localization

ImageNet定位（LOC）任务[36]要求对对象进行分类和定位。在[40, 41]的研究基础上，我们假设首先采用图像级分类器来预测图像的类别标签，然后定位算法仅基于预测的类别来预测边界框。我们采用"按类回归"（PCR）策略[40, 41]，为每个类别学习一个边界框回归器。我们预先对用于ImageNet分类的网络进行了训练，然后再对其进行定位微调。我们在提供的1000类ImageNet训练集上训练网络。

method	top-5 localization err	
	val	test
OverFeat [40] (ILSVRC'13)	30.0	29.9
GoogLeNet [44] (ILSVRC'14)	-	26.7
VGG [41] (ILSVRC'14)	26.9	25.3
ours (ILSVRC'15)	**8.9**	**9.0**

Table 14. 在ImageNet数据集上与最先进方法的定位错误率（%）比较。

我们的定位算法基于[32]的RPN框架进行了一些修改。与[32]以类别无关的方式不同，我们的定位RPN设计成按类形式。这个RPN以两个同级的1×1卷积层结束，用于二元分类（*cls*）和框回归（*reg*），与[32]类似。*cls*和*reg*层都是按类形式的，与[32]不同。具体来说，*cls*层输出为1000维，每个维度是用于预测是否为对象类别的二元逻辑回归；*reg*层输出为1000×4维，包含1000个类别的框回归器。与[32]类似，我们的边界框回归是参考每个位置上的多个平移不变的"锚定"框。

像我们的ImageNet分类训练一样（第4.1节），我们随机采样224×224的裁剪来进行数据增强。我们在微调时使用256个图像的小批量大小。为避免负样本占主导地位，我们为每个图像随机采样8个锚点，其中来自采样的正负锚点比率为1:1[32]。在测试时，网络进行全卷积地应用于图像。

表 13 比较了定位结果。按照[41]，我们首先进行了"神谕"测试，使用地面真实类别作为分类预测。VGG 的论文[41] 报告了使用地面真实类别的中心裁剪错误率为33.1%（表 13）。在相同设置下，我们的 RPN 方法使用 ResNet-101 网络，将中心裁剪错误显著降低至13.3%。这个比较展示了我们框架的优异表现。通过密集（全卷积）和多尺度测试，我们的 ResNet-101 使用地面真实类别达到了11.7%的错误率。使用 ResNet-101 进行类别预测（5项中的前4项错误率为4.6%，表4），则删除前5项的定位错误率为14.4%。

[5] http://host.robots.ox.ac.uk:8080/anonymous/3OJ4OJ.html，提交日期为2015年11月26日

上述结果仅基于 Faster R-CNN [32] 中的提议网络（RPN）。人们可以在 Faster R-CNN 中使用检测网络（Fast R-CNN [7]）来改善结果。但我们注意到，在这个数据集中，一幅图像通常只包含一个主要对象，提议区域高度重叠，并且具有非常相似的 RoI 池化特征。因此，Fast R-CNN [7] 的图像中心训练生成具有小变异的样本，这可能不适用于随机训练。受此激励，我们当前的实验中使用原始 R-CNN [8]，它是 RoI 中心的，代替 Fast R-CNN。

我们的 R-CNN 实现如下。我们在训练图像上应用了上述训练的每类 RPN，以预测地面真实类别的边界框。这些预测框充当与类别相关的提议。对于每幅训练图像，从提议中提取最高得分的 200 个样本作为训练样本，用于训练 R-CNN 分类器。图像区域从提议中裁剪，调整为 224 × 224 像素，并与 R-CNN [8] 中一样，馈送到分类网络中。这个网络的输出包括*cls*和*reg*的两个同级 fc 层，也以每类形式存在。这个 R-CNN 网络通过 RoI 中心的方式在训练集上微调，使用一个 mini-batch 大小为 256。在测试时，RPN 为每个预测类别生成得分最高的 200 个提议，R-CNN 网络用于更新这些提议的得分和框位置。

这种方法将删除前5项的定位错误率降低至10.6%（表 13）。这是我们在验证集上的单模型结果。通过对分类和定位进行网络集成，我们在测试集上实现了 9.0% 的删除前5项定位错误率。这个数字明显优于 ILSVRC 14 结果（表 14），显示出错误率的相对减少率为 64%。*这个结果赢得了 ILSVRC 2015 年 ImageNet 定位任务第一名 *。

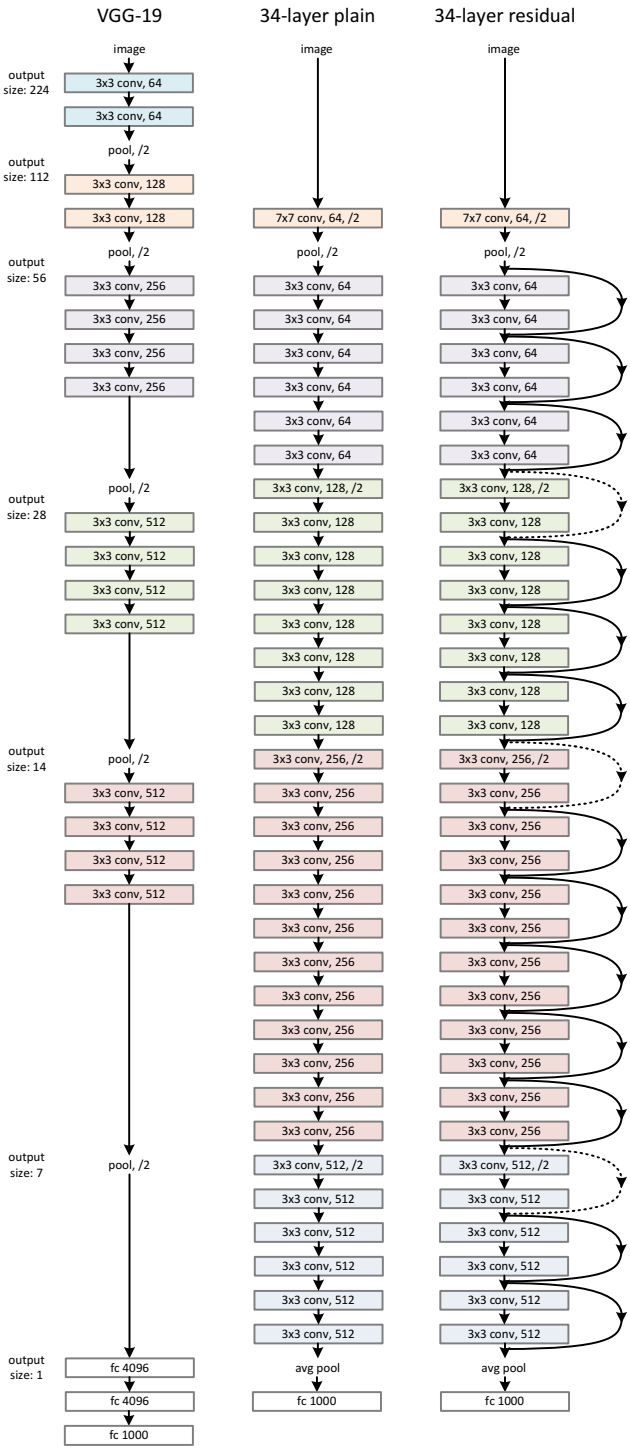

Figure 3. ImageNet的示例网络架构。 **左侧**：VGG-19模型[41]（196亿 FLOPs）作为参考。 **中间**：一个具有34个参数层（36亿 FLOPs）的普通网络。 **右侧**：一个具有34个参数层（36亿 FLOPs）的残差网络。虚线的快捷方式增加了维度。**表 1** 显示更多细节和其他变体。

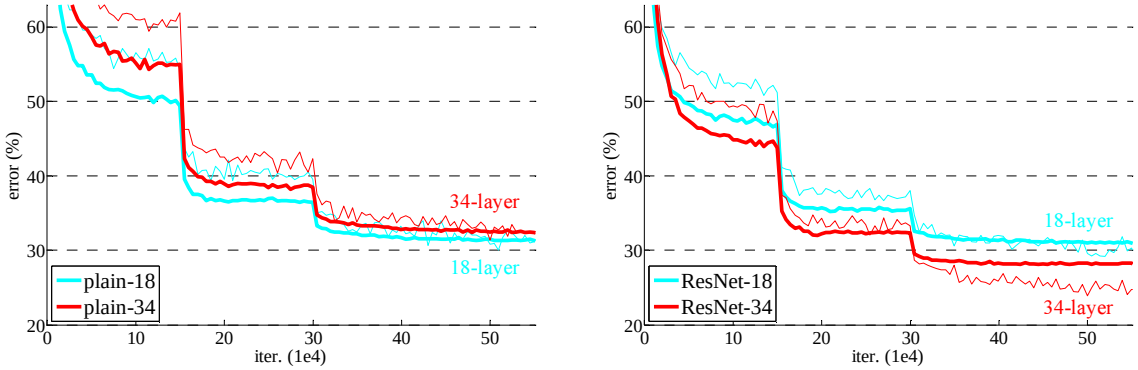

Figure 4. 在**ImageNet**上训练。细曲线表示18层和34层的普通网络的训练误差，粗曲线表示中心裁剪的验证误差。左图：18层和34层的普通网络。右图：18层和34层的ResNets。在这个图中，相比普通网络，残差网络没有额外的参数。

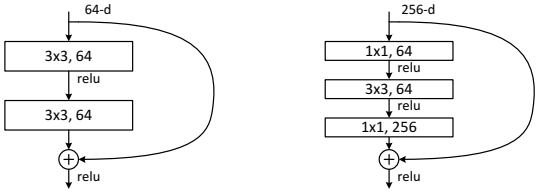

Figure 5. 对于ImageNet，一个更深层的残差函数$\mathcal{F}$。左图：ResNet-34中的一个构建块（位于56×56特征图上），如图 3所示。右图：ResNet-50/101/152中的“瓶颈”构建块。

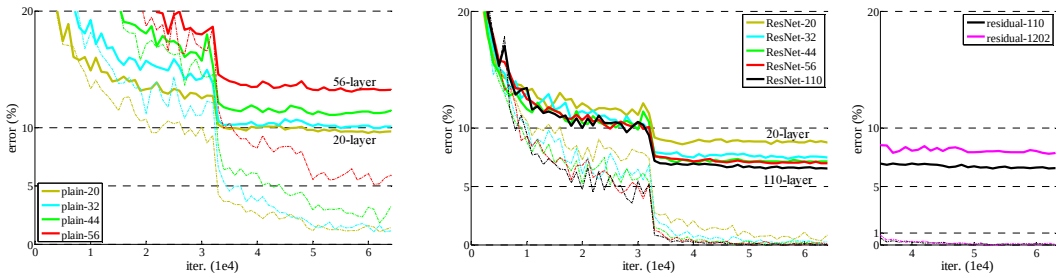

Figure 6. 在 **CIFAR-10**数据集上进行训练。虚线表示训练误差，粗线表示测试误差。 **左**：普通网络。plain-110的误差高于60%，未显示。 **中**：ResNets。 **右**：具有110层和1202层的ResNets。

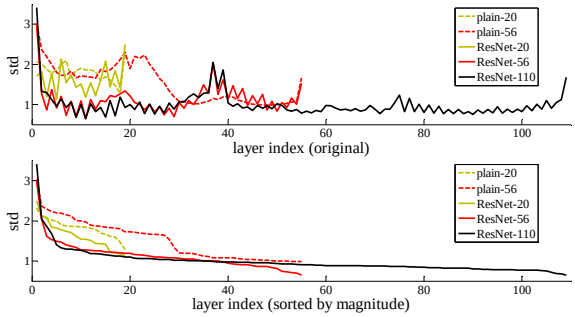

Figure 7. CIFAR-10数据集上各层响应的标准差（std）。这些响应是每个3×3层的输出，在BN之后以及非线性之前。 **顶部**：层按原始顺序排列。 **底部**：响应按降序排列。