用于图像识别的深度残差学习

何恺明 张祥雨 任少卿 孙剑
微软研究院
{kahe, v-xiangz, v-shren, jiansun}@microsoft.com

摘要

更深层的神经网络更难以训练。我们提出了一种残差学习框架，以方便训练比以往使用的网络更深的网络。我们明确地将各层重新表述为参照层输入学习残差函数，而不是学习无参照的函数。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，并能通过大幅提高深度获得准确性。在 ImageNet 数据集上，我们评估了深度高达 152 层的残差网络——比 VGG 网络深 8 倍，但复杂度仍然较低。这些残差网络的集成在 ImageNet 测试集上达到了 3.57%的错误率。这一结果在 ILSVRC 2015 分类任务中获得了第一名。我们还在 CIFAR-10 上展示了 100 层和 1000 层的分析。

表示的深度对于许多视觉识别任务至关重要。仅仅由于我们极其深的表示，我们在 COCO 目标检测数据集上获得了 28%的相对改进。深度残差网络是我们提交给 ILSVRC & COCO 2015 竞赛的基础，在这些竞赛中，我们还在 ImageNet 检测、ImageNet 定位、COCO 检测和 COCO 分割任务中赢得了第一名。

1. 引言

卷积神经网络为图像分类带来了一系列突破，深度网络以端到端的多层方式自然地集中了/低/中/高层特征和分类器，特征的“层次”可以通过堆叠层的数量（深度）来丰富，最新的研究表明，网络的深度是很重要的，在具有挑战性的 ImageNet 数据集上取得领先成果的模型都“非常深”，深度从 16 到 30 不等。许多复杂的的视觉识别任务也从很深的模型中获益匪浅。

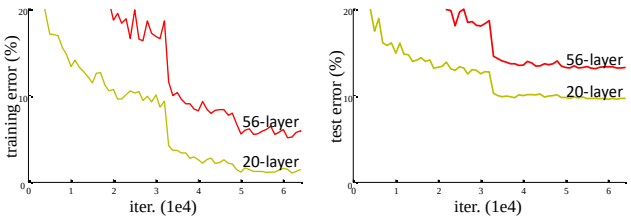

图 1. 20 层和 56 层 "普通 "网络在 CIFAR-10 上的训练误差（左）和测试误差（右）。较深的网络具有较高的训练误差，因此测试误差也较高。图 4 显示了 ImageNet 上的类似现象。

在“深度”这个概念的驱动下，一个问题出现了：想要学习更好的网络，只要简单地堆叠更多的层这么简单就可以吗？回答这个问题的一个例子就是臭名昭著的梯度消失/爆炸问题，这个问题从一开始就阻碍了收敛，然而归一化初始化和中间归一化层在很大程度上解决了这一问题，他们使具有数十个层的网络开始收敛，从而实现随机梯度下降和反向传播。

当更深的网络能开始收敛时，一个退化问题就暴露出来了：随着网络深度的增加，准确度会达到饱和（这可能并不奇怪），然后迅速退化，令人意想不到的是，这种退化并不是由过拟合引起的，在适当的深度模型中增加更多的层会导致更高的训练误差，这一点在文献[11,42]中已有说明，并在我们的实验中得到了充分验证，图 1 就显示了一个典型的例子。

训练精度的降低表明，并非所有的系统都同样容易优化。让我们考虑一个较浅的架构和该结构增加了更多层次的较深架构。通过构造，存在一个针对更深层模型的问题：添加的层是新的参数，非添加的层是从已经学习过的较浅的层复制过来的，这种结构表明，后者深层模型产生的训练误差不应该高于前者较浅的模型。但是实验结果表明，我们目前手头的求解器无法找到这样的解决方案。

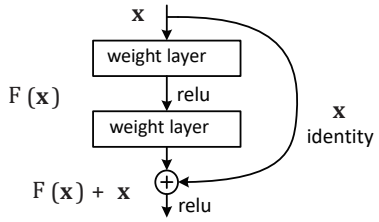

图 2. 残差学习：一个构建块

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，我们不希望每一组堆叠的层直接拟合期望的映射，而是明确地让这些层拟合一个残差映射。在形式上，将期望的映射表示为 H(x)，我们让堆叠的非线性层拟合另一个映射 F(x) := H(x)−x。原始的映射被重铸为 F(x)+x。我们假设，优化残差映射比优化原始的、无参照的映射更容易。极端情况下，如果恒等映射是最优的，那么将残差推至零将比通过堆叠一堆非线性曾来拟合恒等映射要容易得多。

F(x)+x 的公式可以通过带有“捷径连接”的前馈神经网络来实现（图 2）。快捷连接[2, 34, 49]是跳过一层或多层的连接，在我们的例子中，捷径连接只是简单地进行恒等映射，他们的输出被添加到堆叠的输出上（图 2）。恒等快捷连接既不增加额外的参数，也不增加计算复杂性，整个网络仍然可以通过梯度下降与反向传播进行端到端的训练，并且可以在不修改求解器（如 coffe）的情况下轻松实现。

我们在 ImageNet[36]上进行了全面实验，已展示退化问题并评估我们的方法。我们证明了：1.我们的深度残差网络很容易优化，但对应的“普通”网络（简单地堆叠层的网络）在深度增加时会表现出更高的训练误差。2.我们的深度残差网络很容易从深度的大幅增加中获得准确性提升，产生的结果明显优于以前的网络。

类似的现象也在 CIFAR-10 数据集上出现，这表明优化困难和我们方法的效果并不仅限于特定的数据集。我们展示了在这个数据集上成功训练的超过 100 层的模型，并探索了超过 1000 层的模型。

在 ImageNet 分类数据集 [36] 上，我们通过极深的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 上最深的网络，但其复杂度仍低于 VGG 网络 [41]。我们的集合在 ImageNet 测试集上的前五名错误率为 3.57%，并在 ILSVRC 2015 分类竞赛中获得第一名。极深表征在其他识别任务中也具有出色的泛化性能，并使我们在以下任务中进一步赢得了第一名：ImageNet 检测、ImageNet 定位： 在 ILSVRC 和 COCO 2015 比赛中，我们还获得了 ImageNet 检测、ImageNet 定位、COCO 检测和 COCO 分割的第一名。这些强有力的证据表明，残差学习原理是通用的，我们期待它能适用于其他视觉和非视觉问题。

2. 相关工作

残差表示：在图像识别领域，VLAD 是一种通过与字典相关的残差向量进行编码的表示方法，而 Fisher 向量可以被视为 VLAD 的概率形式。这两者都是图像检索和分类中的浅层表示。在向量量化方面，编码残差向量已被证明比编码原始向量更为有效

在像素层面的计算机视觉和计算机图形学中，为了解决偏微分方程（PDEs），一般使用的是多重网格方法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较粗和较细尺度之间的残差解。多网格法的另一种替代方法是分层及预处理法[45,46],他依赖于代表两个尺度之间残差向量的变量。研究表明，这些求解器的收敛速度比标准求解器块很多，因为标准求解器不知道解的残差性质。这些方法表明，良好的重表述或预处理可以简化优化过程。

捷径连接：捷径连接的理论和实践已经研究了很长时间。训练多层感知机(MLP)的早期做法是增加一个从网络输入连接到网络输出的线性层。在文献[44,24]中，一些中间层直接连接到辅助分类器，以解决梯度消失/爆炸问题，文献[39,38,31,47]提出了通过快捷连接实现的方法，用于集中层响应，梯度和传播误差。在[44]中，一个“inception”层由一个快捷分支和几个更深的分支组成。

与我们工作同时进行的“高速公路网络”提出了带有门控函数的快捷连接[15]。这些们是这些门是依赖于数据的，并且带有参数，而我们的捷径连接时没有参数的。当控制捷径“关闭”（趋近于零）的时候，高速公路网络中的层就代表了非剩余函数。与此相反，我们的公式总是能学习残差函数；我们的捷径连接永远不会关闭，所有的信息都会通过，并学习额外的残差函数。此外，高速公路网络在深度增加后（如超过 100 层），准确性没有提高。

3. 深度残差网络

3.1. 残差学习

让我们把 $\mathbf{H(x)}$看作是几个对叠层拟合的底层映射（不一定是整个网络的底层映射），x 表示这层中第一个层的输入。假设多个非线性层可以渐进地逼近复杂的函数，那么这等价于假设他们可以渐进地逼近残差函数，即 $\mathbf{H(x) - x}$（假设输入和输出具有相同维度）。**因此，我们明确地让这些层逼近一个残差函数** $\mathbf{F(x) := H(x) - x}$，而不是期望堆叠层逼近 H（x）。原始函数因此变成了 $\mathbf{F(x)+x}$,尽管这两种形式都应该能渐渐逼近所需要的函数，但是学习的难易程度不同。

这种重新表述的动机是由于网络退化问题的反直觉现象（图 1，左）。正如我们在介绍中讨论的，如果添加层可以构造为恒等映射，那么较深的模型的训练误差应该不大于较浅的模型。退化问题表明，求解器在逼近多非线性层的恒等映射时可能存在困难。通过残差学习的重新表述，如果恒等映射是最优的，求解器可以简单地将多个非线性层的权值向零逼近，从而逼近恒等映射。

在实际情况中，恒等映射不太可能是最优解，但我们的重新表述可能有助于预先解决问题。如果最优函数更接近于恒等映射而不是零映射，那么求解器应该更容易找到与恒等映射相关的扰动，而不是将该函数作为一个新的函数来学习。我们通过实验（图 7）表明，学习到的残差函数通常具有较小的响应，这表明恒等映射提供了合理的预处理。

3.2.通过捷径连接传递恒等映射

我们对每个模块都是用残差学习，构建块如图 2 所示，本文中我们把构建块定义为：

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

这里的 x 和 y 是模块的输入和输出向量，函数 $F(\mathbf{x},\{W_i\})$表示待学习的残差映射。对于图 2 中有两层的例子，他有两个层：$F = W_2\sigma(W_1\mathbf{x})$，其中 σ 表示 ReLU，为了简化符号，省略了偏置。操作 F+x 是通过一个捷径连接和逐元素加法来执行的。我们在加法后（即 $\sigma(\mathbf{y})$，图 2）采用 ReLU.

方程（1）中的捷径连接既不引入额外的参数，也不增加计算复杂度。这不仅在实践中很有吸引力，同时在我们比较普通网络和残差网络实也很重要。我们可以比较同时具有相同数量的参数、深度、宽度、和计算成

本（除了可以忽略不计的元素加法）的普通/残差网络。

在公式 1 中，x 和 F 的维度必须相等。如果不是这种情况（例如，当改变输入/输出通道数时），我们可以通过捷径连接执行线性投影 w 来匹配。

在公式 1 中，x 和 F 的维度必须相等。如果不是这种情况（例如，当改变输入/输出通道数时），我们可以通过捷径连接执行线性投影 w 来匹配:

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

我们也可以使用公式 1 中的方阵 W，但我们将通过实验证明，恒等映射足以解决网络退化问题，并且花销小，因此 W_s 仅在匹配维度时使用

残差函数 F 的形式是灵活的。本文中的实验设计一个具有两层或三层的函数 F（见图 5），而更多层也是可能的。但是如果 F 只有一个单独的层，公式（1）类似于线性层：$\mathbf{y} = W_1\mathbf{x}+\mathbf{x}$,我们没有观察到这样的 F 有什么优势。

我们还注意到，尽管上述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 F(x,{Wi}) 可以表示多个卷积层。逐元素相加是在两个特征图上逐通道执行的。

3.3 网络体系结构

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。

普通网络：我们的普通基准模型（图 3，中间）主要受到 vgg 网络（图 3，左边）的启发。卷积层主要使用 3x3 的滤波器，，并遵循两个简单的设计规则：1.对于相同的输出特征大小，层具有相同数量的滤波器；2.如果特征图大小减半，滤波器的数量加倍，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，网络以全局平均池化和一个具有 softmax 的 1000 路全连接层结束。图 3（中间）中的加权层总数为 34

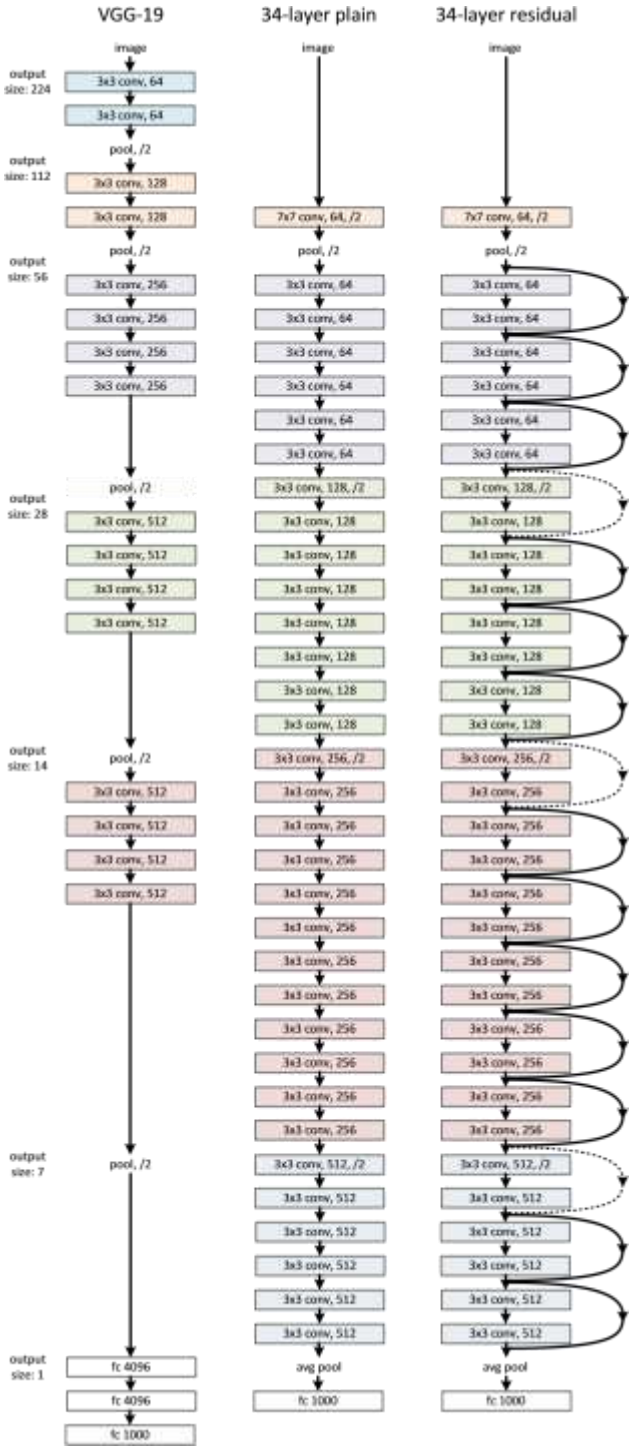

图 3. ImageNet 的示例网络架构。左侧：VGG-19 模型[41]（196 亿次 FLOPs）作为参考。中间：具有 34 个参数层（36 亿次 FLOPs）的普通网络。右侧：具有 34 个参数层（36 亿次 FLOPs）的残差网络。虚线捷径连接增加了维度。表 1 显示了更多细节和其他变体。

值得注意的是，我们的模型具有比 VGG 网络[41]（图 3，左边）更少的滤波器和更低的复杂度。我们的 34 层基准模型具有 36 亿次 FLOPs（乘加运算），仅为 VGG-19（196 亿次 FLOPs）的 18%。

残差网络：基于上述普通网络，我们插入了捷径连接（图 3，右边），将网络转变为其对应的残差版本。当输入和输出具有相同的维度时，可以直接使用恒等捷径连接（方程 1）（图 3 中的实线为捷径连接）。当维度增加时间（图 3 中的虚线捷径连接），我们考虑两种选项：a.捷径连接仍然执行恒等映射，增加的额度使用零填充，这个选项不会引入额外的参数；b.使用方程（2）中的投影捷径连接来匹配维度（使用 1x1 的卷积完成）。对于这两种选项，当捷径连接跨越两种尺寸的特征图时，他们会以步幅 2 执行。

3.4. 实现

我们对 ImageNet 的实现遵循[21,41]中的实践。在[256,480]中随机采样图像的短边以进行缩放[41]。从图像或其水平翻转中随机采样 224×224 裁剪，并减去每像素平均值[21]。使用[21]中的标准颜色增强。我们在每次卷积之后和激活之前采用批归一化(BN)[16]。我们像[13]中那样初始化权重，并从头开始训练所有的 plain/residual 网络。我们使用 SGD 的小批量大小为 256。学习率从 0.1 开始，当误差趋于平稳时除以 10，并且模型被训练到 60×104 迭代。我们使用 0.0001 的权重衰减和 0.9 的动量。遵循文献[16]的做法，我们没有使用 dropout[14]。

在测试中，我们采用标准的 10 种作物测试进行比较研究[21]。为了获得最佳结果，我们采用了[41,13]中的全卷积形式，并在多个尺度上平均得分(图像被调整大小，使较短的一面在{224,256,384,480,640}中)。

4.实验

4.1. ImageNet 分类

我们在包含 1000 个类的 ImageNet 2012 分类数据集[36]上评估我们的方法。模型在 128 万张训练图像上进行训练，并在 5 万张验证图像上进行评估。我们还获得了由测试服务器报告的 100k 测试映像的最终结果。我们评估前 1 和前 5 的错误率。

普通网络：我们首先评估了 18 层和 34 层的普通网。34 层平面网如图 3(中)所示。18 层的平面网也是类似的形式。请参见表 1 了解详细的体系结构。

表 2 中的结果显示，更深的 34 层普通网络的验证误差高于较浅的 18 层普通网络。为了揭示原因，在图 4（左侧）中，我们比较它们在训练过程中的训练/验证误差。我们观察到了退化问题。

34 层普通网络在整个训练过程中的训练误差都更高，即使 18 层普通网络的解空间是 34 层网络的子空间。我们认为这种优化困难不太可能是由梯度消失引起的。这些普通网络是使用 BN[16]进行训练的，这确保了前向传播信号具有非零方差。我们还验证了使用 BN 时反向传播的梯度具有健康的范数。因此，前向和后向信号都不会消失。事实上，34 层普通网络仍能够达到竞争性的准确性（表 3），这表明求解器在一定程度上是有效的。我们推测深层普通网络可能具有指数级别的低收敛速率，这影响了训练误差的降低[我们尝试了更多的训练迭代（3×），仍然观察到了退化问题，这表明这个问题不能简单地通过增加迭代次数来解决。]。这种优化困难的原因将在未来进行研究。

残差网络。接下来我们评估 18 层和 34 层的残差网络（ResNets）。基准架构与上述普通网络相同，只是在每对 3×3 滤波器中添加了一个快捷连接，如图 3（右侧）所示。在第一次比较中（表 2 和图 4 右侧），我们对

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

表 1.图像网络的架构。括号中显示的是构建块（另见图 5），以及堆叠的构建块数。conv31、conv41 和 conv51 进行向下采样，采样率为 2。

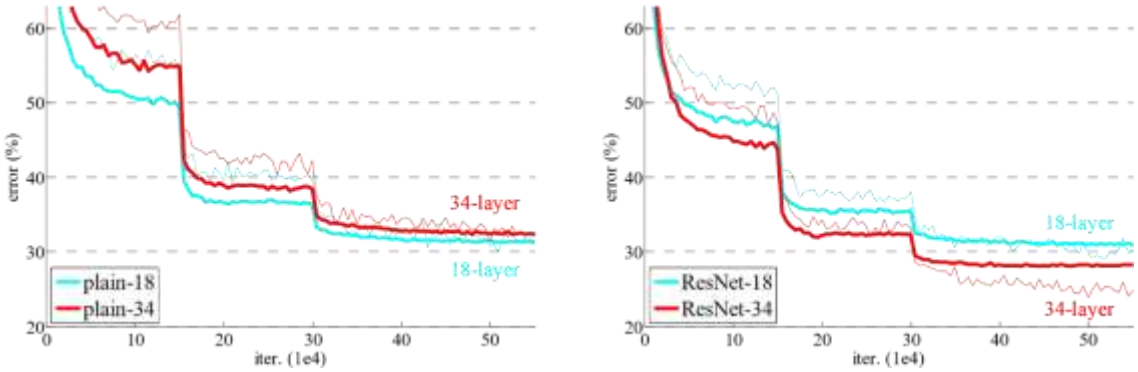

图 4.图像网络的训练。左图：18 层和 34 层的普通网络。 右图：18 层和 34 层的 ResNet。在此图中，残差网络的参数与普通网络的参数相差甚远。在这幅图中，残差网络的参数与普通网络的参数相比没有超出。

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

表 2. ImageNet 验证的前 1 名误差（%，10 次裁剪测试）。与普通网络相比，ResNets 没有额外的参数。图 4 显示了训练过程。

所有快捷连接使用恒等映射，并对增加的维度进行零填充（选项 A）。因此，与普通对应物相比，它们没有额外的参数。

尽管 18 层普通网络的解空间是 34 层网络的子空间，但在整个训练过程中，34 层普通网络的训练误差更大。

我们认为这种优化困难不太可能是由梯度消失引起的。这些普通网络采用了批量归一化（BN）[16]，这确保了前向传播的信号具有非零方差。我们还验证了使用 BN 的后向传播梯度表现出健康的范数。因此，无论是前向还是后向信号都没有消失。实际上，34 层的普通网络仍然能够达到有竞争力的准确度（表 3），这表明求解器在某种程度上是有效的。我们推测，深层普通网络可能具有指数级低的收敛速度，这影响了训练误差的减少。这种优化困难的原因将在未来进行研究。

残差网络：接下来，我们评估 18 层和 34 层的残差网络（ResNets）。基准架构与上述普通网络相同，只是如图 3（右）所示，每对 3×3 滤波器之间添加了一个快捷连接。在第一次比较中（表 2 和图 4 右），我们对所有快捷连接使用恒等映射，并对增加的维度使用零填充（选项 A）。因此，它们与普通网络相比没有额外的参数。

从表 2 和图 4 中，我们得到了三个主要的观察结果。首先，随着残差学习的引入，情况发生了逆转——34 层的 ResNet 比 18 层的 ResNet 表现更好（高出 2.8%）。更重要的是，34 层的 ResNet 在训练误差上显著降低，并且能够推广到验证数据上。这表明在这种设置下，退化问题得到了很好的解决，并且我们能够从增加的深度中获得准确性的提升。

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

表 3. 在 ImageNet 验证集上的错误率（%，10-crop 测试）。VGG-16 基于我们的测试。ResNet-50/101/152 采用选项 B，仅在增加维度时使用投影。

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**

表 4. 单一模型在 ImageNet 验证集（不包括报告在测试集上的结果）上的错误率（%）

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**

表 5. 集成模型的错误率（%）。top-5 错误率是在 ImageNet 测试集上，由测试服务器报告的。

其次，与它的普通网络对应物相比，34 层的 ResNet 将 top-1 错误率降低了 3.5%（表 2），这是由于成功降低了训练误差（图 4 右与左对比）。这一比较验证了残差学习在极深系统中的有效性。

最后，我们也注意到 18 层的普通网络/残差网络在准确性上是相当的（表 2），但 18 层的 ResNet 收敛得更快（图 4 右与左对比）。当网络"不是特别深"（此处为 18 层）时，当前的 SGD 求解器仍然能够为普通网络找到良好的解决方案。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。

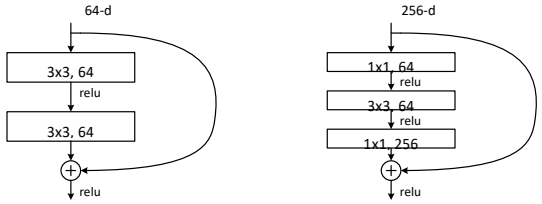

图 5. ImageNet 的更深层残差函数 F。左图：ResNet34 中使用的构建块（在 56×56 特征图上），如 Fig. 3 所示。右图：ResNet-50/101/152 中使用的"瓶颈"构建块。

恒等捷径连接与投影捷径连接。我们已经展示了无参数的恒等捷径连接有助于训练。接下来，我们研究投影捷径连接（公式.(2)）。在表 3 中，我们比较了三个选项：(A) 对于增加的维度使用零填充快捷连接，所有快捷连接都是无参数的（与表 2 和图 4 右相同）；(B) 对于增加的维度使用投影快捷连接，其他快捷连接是恒等的；以及(C) 所有快捷连接都是投影的。

表 3 显示，所有三种选项都明显优于普通的对应网络。选项 B 略优于选项 A。我们认为这是因为选项 A 中零填充的维度实际上

没有进行残差学习。选项 C 略优于选项 B，我们将其归因于由许多（十三次）投影快捷连接引入的额外参数。但是，A/B/C 之间的微小差异表明，投影快捷连接对于解决退化问题并不是必需的。因此，我们在本文的其余部分不使用选项 C，以减少内存/时间复杂度和模型大小。对于下面介绍的瓶颈架构，恒等快捷连接尤其重要，因为它不会增加复杂性。

更深的瓶颈架构。接下来，我们描述我们为 ImageNet 设计的更深的网络。由于对我们可以承受的训练时间的担忧，我们将构建块修改为瓶颈设计。对于每个残差函数 F，我们使用三层堆栈而不是两层（图 5）。这三层是 1×1、3×3 和 1×1 卷积，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，其中两种设计具有相似的时间复杂度。

无参数的恒等快捷连接对于瓶颈架构尤其重要。如果图 5（右）中的恒等快捷连接被投影快捷连接替换，可以证明时间复杂度和模型大小都会翻倍，因为快捷连接连接到了两个高维度的末端。因此，恒等快捷连接为瓶颈设计带来了更高效的模型。

50 层 ResNet：我们将 34 层网络中的每个 2 层块替换为这种 3 层瓶颈块，从而得到一个 50 层 ResNet（表 1）。我们对于增加的维度使用选项 B（即使用投影快捷连接）。这个模型有 38 亿次浮点运算（FLOPs）。

101 层和 152 层 ResNet：我们通过使用更多的 3 层瓶颈块来构建 101 层和 152 层 ResNet（表 1）。值得注意的是，尽管深度显著增加，但 152 层 ResNet（113 亿次浮点运算）的复杂度仍然低于 VGG-16/19 网络（153 亿/196 亿次浮点运算）。

50/101/152 层 ResNet 比 34 层 ResNet 在准确性上有显著的提升（表 3 和表 4）。我们没有观察到退化问题，因此从显著增加的深度中获得了显著的准确性提升。深度的益处在所有评估指标中都得到了体现（表 3 和表 4）。

与最先进方法的比较。在表 4 中，我们与之前最佳的单模型结果进行了比较。我们的基准 34 层 ResNet 已经达到了非常有竞争力的准确性。我们的 152 层 ResNet 的单模型前 5 验证错误率为 4.49%。这个单模型结果超过了所有之前的集成结果（表 5）。我们组合了六个不同深度的模型来形成一个集成（在提交时只有两个 152 层模型）。这导致了测试集上的 3.57%前 5 错误率（表 5）。这一参赛作品在 ILSVRC 2015 中获得了第一名。

4.2. CIFAR-10 和分析

我们对 CIFAR-10 数据集进行了更多的研究 [20]，该数据集包含 10 个类别的 5 万张训练图像和 1 万张测试图像。我们对训练集上的模型进行训练，并在测试集上对其进行评估。我们的关注点在于极深网络的行为，而不是追求最先进的结果，因此我们故意使用了以下简单的网络结构。

平滑/残差网络的结构如图 3（中间/右侧）所示。网络的输入是 32×32 的图像，并且对每个像素都进行了减均值处理。第一层是 3×3 的卷积层。然后我们使用一个堆叠层，共有 6n 层，每层都包含 3×3 的卷积操作，对大小分别为 32、16、8 的特征图进行操作，每种特征图大小有 2n 层。滤波器的数量分别为 16、32、64。采用 2×2 的卷积进行下采样。网络的最后是全局平均池化、一个 10 维的全连接层和 softmax 操作。整个网络共有 6n+2 层加权层。下面的表格总结了网络结构：

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

当使用捷径连接时，它们连接到 3×3 层对（总共有 3n 个捷径连接）。在该数据集上，我们始终使用身份捷径（即选项 A），因此我们的残差网络与普通网络具有完全相同的深度、宽度和参数数量。

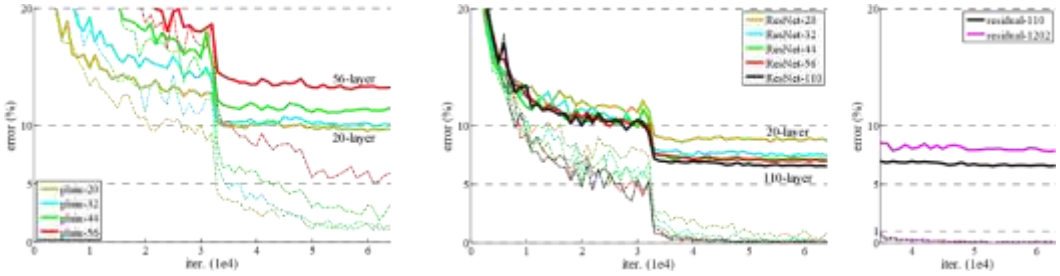

图 6. 在 CIFAR-10 上进行的训练。虚线表示训练误差，粗线表示测试误差。左图：普通网络。plain-110 的误差高于 60%，故未显示。中间：ResNets： ResNets. 右图 具有 110 层和 1202 层的 ResNets。

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

表 6：CIFAR-10 测试集中的分类错误。对于 ResNet-110，我们进行了 5 次单元测试，并显示了 "最佳（平均值）"。表 6：CIFAR-10 测试集的分类误差。

我们使用了 0.0001 的权重衰减和 0.9 的动量，并采用了 [13] 和 BN [16] 中的权重初始化，但没有丢弃。这些模型在两台 GPU 上以 128 的最小批量进行训练。我们从 0.1 的学习率开始，在 32k 和 48k 次迭代时将学习率除以 10，在 64k 次迭代时终止训练，这是由 45k/5k train/val 分割决定的。我们采用 [24] 中的简单数据增强方法进行训练：每边填充 4 个像素，从填充图像或其水平翻转图像中随机抽取 32×32 的裁剪。测试时，我们只评估原始 32×32 图像的单一视图。

我们比较了 n = {3,5,7,9}，从而得出 20、32、44 和 56 层网络。图 6（左）显示了普通网络的行为。深度平原网络会受到深度增加的影响，当深度增加时会表现出更高的训练误差。这一现象与 ImageNet 上的情况类似

(图 4，左）和 MNIST（见 [42]）上，表明这种优化困难是一个基本问题。

图 6（中）显示了 ResNets 的行为。与 ImageNet 案例（图 4 右）类似，我们的 ResNets 也能克服优化困难，并在深度增加时显示出准确率的提高。

我们进一步探讨了 n = 18，即 110 层的 ResNet。在这种情况下，我们发现 0.1 的初始学习率稍大，无法开始收敛。因此，我们使用 0.01 来预热训练，直到训练误差低于 80%（约 400 次迭代），然后回到 0.1 继续训练。其余的学习计划如前所述。这个 110 层的网络收敛良好（图 6，中间）。与其他深层和薄层网络相比，它的参数更少，但也跻身最先进结果的队伍中。（6.43%，表 6）

层响应分析：图 7 显示了各层响应的标准差(std)。响应是每个 3×3 层的输出，在 BN 之后和在其他非线性(ReLU/加法)之前。对于 ResNets，该分析揭示了残差函数的响应强度。图 7 显示，resnet 的响应通常比 plain 的响应小。这些结果支持我们的基本动机(第 3.1 节)，即残差函数通常比非残差函数更接近于零。我们还注意到更深的 ResNet 具有较小的响应幅度，如图 7 中 ResNet-20, 56 和 110 的比较所证明。当有更多的层时，单个层的 ResNets 倾向于较少地修改信号。

探索超过 1000 层：我们探索了一个超过 1000 层的深度模型。我们设置 n = 200，得到一个 1202 层的网络，按照上面的描述进行训练。我们的方法没有优化难度，这个 103 层的网络能够实现训练误差<0.1%(图 6，右)。它的测试误差仍然相当好(7.93%，表 6)。

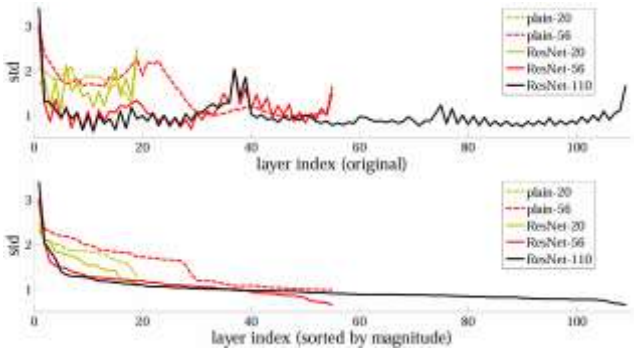

图 7. CIFAR10 数据集上层响应的标准差(std)。这些响应是每个 3×3 层在 BN 之后、非线性激活函数之前的输出。上图：层按其原始顺序显示。下图：响应按降序排列。

但是，这种深度模型还存在一些问题。这个 1202 层网络的测试结果比我们的 110 层网络差，尽管两者的训练误差相似。我们认为这是由于过度拟合造成的。对于这个小数据集来说，1202 层网络可能过于庞大（19.4M）。为了在该数据集上获得最佳结果（[10, 25, 24, 35]），我们采用了诸如 maxout [10] 或 dropout [14] 等强正则化方法。在本文中，我们没有使用 maxout/dropout，只是简单地通过深层和薄层架构设计施加正则化，没有分散对优化难点的关注。但结合更强的正则化可能会改善结果，我们将在未来对此进行研究。

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**

表 7 所示。使用基准 Faster R-CNN 的 PASCAL VOC 2007/2012 测试集上的目标检测 mAP(%)。请参见表 10 和表 11 以获得更好的结果。

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

表 8. 在 COCO 验证集上使用基准 Faster R-CNN 进行目标检测的 mAP（%）。另请参阅表 9 以获取更好的结果。

4.3. PASCAL 和 MS COCO 上的目标检测

我们的方法在其他识别任务上表现出良好的泛化性能。表 7 和 8 显示了在 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，我们对用 ResNet-101 替换 VGG-16 [41]的改进感兴趣。使用这两种模型的检测实现（见附录）是相同的，因此增益只能归因于更好的网络。特别值得注意的是，在具有挑战性的 COCO 数据集上，我们获得了 COCO 标准度量（mAP@[.5, .95]）的 6.0%增加，相对改进为 28%。这一收益完全归因于学习到的表示。

基于深度残差网络，我们在 ILSVRC & COCO 2015 比赛的多个赛道中获得了第一名：ImageNet 检测，ImageNet 定位，COCO 检测和 COCO 分割。详细信息请参见附录。

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 coadaptation 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. Muller. 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. Deeplysupervised 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. Dollar, 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. Montufar, 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

In this section we introduce our detection method based on the baseline Faster R-CNN [32] system. The models are initialized by the ImageNet classification models, and then fine-tuned on the object detection data. We have experimented with ResNet-50/101 at the time of the ILSVRC & COCO 2015 detection competitions.

Unlike VGG-16 used in [32], our ResNet has no hidden fc layers. We adopt the idea of "Networks on Conv feature maps" (NoC) [33] to address this issue. We compute the full-image shared conv feature maps using those layers whose strides on the image are no greater than 16 pixels (*i.e.*, conv1, conv2 - x, conv3 x, and conv4 x, totally 91 conv layers in ResNet-101; Table 1). We consider these layers as analogous to the 13 conv layers in VGG-16, and by doing so, both ResNet and VGG-16 have conv feature maps of the same total stride (16 pixels). These layers are shared by a region proposal network (RPN, generating 300 proposals) [32] and a Fast R-CNN detection network [7]. RoI pooling [7] is performed before conv5 1. On this RoI-pooled feature, all layers of conv5 _x and up are adopted for each region, playing the roles of VGG-16's fc layers. The final classification layer is replaced by two sibling layers (classification and box regression [7]).

For the usage of BN layers, after pre-training, we compute the BN statistics (means and variances) for each layer on the ImageNet training set. Then the BN layers are fixed during fine-tuning for object detection. As such, the BN layers become linear activations with constant offsets and scales, and BN statistics are not updated by fine-tuning. We fix the BN layers mainly for reducing memory consumption in Faster R-CNN training.

PASCAL VOC

Following [7, 32], for the PASCAL VOC 2007 *test* set, we use the 5k *trainval* images in VOC 2007 and 16k *trainval* images in VOC 2012 for training ("07+12"). For the PASCAL VOC 2012 *test* set, we use the 10k *trainval+test* images in VOC 2007 and 16k *trainval* images in VOC 2012 for training ("07++12"). The hyper-parameters for training Faster R-CNN are the same as in [32]. Table 7 shows the results. ResNet-101 improves the mAP by >3% over VGG-16. This gain is solely because of the improved features learned by ResNet.

MS COCO

The MS COCO dataset [26] involves 80 object categories. We evaluate the PASCAL VOC metric (mAP @ IoU = 0.5) and the standard COCO metric (mAP @ IoU = .5:.05:.95). We use the 80k images on the train set for training and the 40k images on the val set for evaluation. Our detection system for COCO is similar to that for PASCAL VOC. We train the COCO models with an 8-GPU implementation, and thus the RPN step has a mini-batch size of 8 images (*i.e.*, 1 per GPU) and the Fast R-CNN step has a mini-batch size of 16 images. The RPN step and Fast RCNN step are both trained for 240k iterations with a learning rate of 0.001 and then for 80k iterations with 0.0001.

Table 8 shows the results on the MS COCO validation set. ResNet-101 has a 6% increase of mAP@[.5, .95] over VGG-16, which is a 28% relative improvement, solely contributed by the features learned by the better network. Remarkably, the mAP@[.5, .95]'s absolute increase (6.0%) is nearly as big as mAP@.5's (6.9%). This suggests that a deeper network can improve both recognition and localization.

B. Object Detection Improvements

For completeness, we report the improvements made for the competitions. These improvements are based on deep features and thus should benefit from residual learning.

MS COCO

Box refinement. Our box refinement partially follows the iterative localization in [6]. In Faster R-CNN, the final output is a regressed box that is different from its proposal box. So for inference, we pool a new feature from the regressed box and obtain a new classification score and a new regressed box. We combine these 300 new predictions with the original 300 predictions. Non-maximum suppression (NMS) is applied on the union set of predicted boxes using an IoU threshold of 0.3 [8], followed by box voting [6]. Box refinement improves mAP by about 2 points (Table 9).

Global context. We combine global context in the Fast R-CNN step. Given the full-image conv feature map, we pool a feature by global Spatial Pyramid Pooling [12] (with a "single-level" pyramid) which can be implemented as "RoI" pooling using the entire image's bounding box as the RoI. This pooled feature is fed into the post-RoI layers to obtain a global context feature. This global feature is concatenated with the original per-region feature, followed by the sibling classification and box regression layers. This new structure is trained end-to-end. Global context improves mAP@.5 by about 1 point (Table 9).

Multi-scale testing. In the above, all results are obtained by single-scale training/testing as in [32], where the image's shorter side is $s = 600$ pixels. Multi-scale training/testing has been developed in [12, 7] by selecting a scale from a feature pyramid, and in [33] by using maxout layers. In our current implementation, we have performed multi-scale *testing* following [33]; we have not performed multi-scale training because of limited time. In addition, we have performed multi-scale testing only for the Fast R-CNN step (but not yet for the RPN step). With a trained model, we compute conv feature maps on an image pyramid, where the image's shorter sides are $s \in \{200, 400, 600, 800, 1000\}$.

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. Object detection improvements on MS COCO using Faster R-CNN and ResNet-101.

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. Detection results on the PASCAL VOC 2007 test set. The baseline is the Faster R-CNN system. The system "baseline+++" include box refinement, context, and multi-scale testing in Table 9.

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. Detection results on the PASCAL VOC 2012 test set (http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=4). The baseline is the Faster R-CNN system. The system "baseline+++" include box refinement, context, and multi-scale testing in Table 9.

We select two adjacent scales from the pyramid following [33]. RoI pooling and subsequent layers are performed on the feature maps of these two scales [33], which are merged by maxout as in [33]. Multi-scale testing improves the mAP by over 2 points (Table 9).

Using validation data. Next we use the 80k+40k trainval set for training and the 20k test-dev set for evaluation. The testdev set has no publicly available ground truth and the result is reported by the evaluation server. Under this setting, the results are an mAP@.5 of 55.7% and an mAP@[.5, .95] of 34.9% (Table 9). This is our single-model result.

Ensemble. In Faster R-CNN, the system is designed to learn region proposals and also object classifiers, so an ensemble can be used to boost both tasks. We use an ensemble for proposing regions, and the union set of proposals are processed by an ensemble of per-region classifiers. Table 9 shows our result based on an ensemble of 3 networks. The mAP is 59.0% and 37.4% on the test-dev set. *This result won the 1st place in the detection task in COCO 2015.*

PASCAL VOC

We revisit the PASCAL VOC dataset based on the above model. With the single model on the COCO dataset (55.7% mAP@.5 in Table 9), we fine-tune this model on the PASCAL VOC sets. The improvements of box refinement, context, and multi-scale testing are also adopted. By doing so

	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. Our results (mAP, %) on the ImageNet detection dataset. Our detection system is Faster R-CNN [32] with the improvements in Table 9, using ResNet-101.

we achieve 85.6% mAP on PASCAL VOC 2007 (Table 10) and 83.8% on PASCAL VOC 2012 (Table 11)[1]. The result on PASCAL VOC 2012 is 10 points higher than the previous state-of-the-art result [6].

[1] http://host.robots.ox.ac.uk:8080/anonymous/3OJ4OJ.html, submitted on 2015-11-26.

ImageNet Detection

The ImageNet Detection (DET) task involves 200 object categories. The accuracy is evaluated by mAP@.5. Our object detection algorithm for ImageNet DET is the same as that for MS COCO in Table 9. The networks are pretrained on the 1000-class ImageNet classification set, and are fine-tuned on the DET data. We split the validation set into two parts (val1/val2) following [8]. We fine-tune the detection models using the DET training set and the val1 set. The val2 set is used for validation. We do not use other ILSVRC 2015 data. Our single model with ResNet-101 has

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. Localization error (%) on the ImageNet validation. In the column of "LOC error on GT class" ([41]), the ground truth class is used. In the "testing" column, "1-crop" denotes testing on a center crop of 224×224 pixels, "dense" denotes dense (fully convolutional) and multi-scale testing.

58.8% mAP and our ensemble of 3 models has 62.1% mAP on the DET test set (Table 12). *This result won the 1st place in the ImageNet detection task in ILSVRC 2015*, surpassing the second place by 8.5 points (absolute).

C. ImageNet Localization

The ImageNet Localization (LOC) task [36] requires to classify and localize the objects. Following [40, 41], we assume that the image-level classifiers are first adopted for predicting the class labels of an image, and the localization algorithm only accounts for predicting bounding boxes based on the predicted classes. We adopt the "per-class regression" (PCR) strategy [40, 41], learning a bounding box regressor for each class. We pre-train the networks for ImageNet classification and then fine-tune them for localization. We train networks on the provided 1000-class ImageNet training set.

Our localization algorithm is based on the RPN framework of [32] with a few modifications. Unlike the way in [32] that is category-agnostic, our RPN for localization is designed in a *per-class* form. This RPN ends with two sibling 1×1 convolutional layers for binary classification (*cls*) and box regression (*reg*), as in [32]. The *cls* and *reg* layers are both in a *per-class* from, in contrast to [32]. Specifically, the *cls* layer has a 1000-d output, and each dimension is *binary logistic regression* for predicting being or not being an object class; the *reg* layer has a 1000×4-d output consisting of box regressors for 1000 classes. As in [32], our bounding box regression is with reference to multiple translation-invariant "anchor" boxes at each position.

As in our ImageNet classification training (Sec. 3.4), we randomly sample 224×224 crops for data augmentation. We use a mini-batch size of 256 images for fine-tuning. To avoid negative samples being dominate, 8 anchors are randomly sampled for each image, where the sampled positive and negative anchors have a ratio of 1:1 [32]. For testing, the network is applied on the image fully-convolutionally.

Table 13 compares the localization results. Following [41], we first perform "oracle" testing using the ground truth class as the classification prediction. VGG's paper [41] re-

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. Comparisons of localization error (%) on the ImageNet dataset with state-of-the-art methods.

ports a center-crop error of 33.1% (Table 13) using ground truth classes. Under the same setting, our RPN method using ResNet-101 net significantly reduces the center-crop error to 13.3%. This comparison demonstrates the excellent performance of our framework. With dense (fully convolutional) and multi-scale testing, our ResNet-101 has an error of 11.7% using ground truth classes. Using ResNet-101 for predicting classes (4.6% top-5 classification error, Table 4), the top-5 localization error is 14.4%.

The above results are only based on the *proposal network* (RPN) in Faster R-CNN [32]. One may use the *detection network* (Fast R-CNN [7]) in Faster R-CNN to improve the results. But we notice that on this dataset, one image usually contains a single dominate object, and the proposal regions highly overlap with each other and

thus have very similar RoI-pooled features. As a result, the image-centric training of Fast R-CNN [7] generates samples of small variations, which may not be desired for stochastic training. Motivated by this, in our current experiment we use the original RCNN [8] that is RoI-centric, in place of Fast R-CNN.

Our R-CNN implementation is as follows. We apply the per-class RPN trained as above on the training images to predict bounding boxes for the ground truth class. These predicted boxes play a role of class-dependent proposals. For each training image, the highest scored 200 proposals are extracted as training samples to train an R-CNN classifier. The image region is cropped from a proposal, warped to 224×224 pixels, and fed into the classification network as in R-CNN [8]. The outputs of this network consist of two sibling fc layers for *cls* and *reg*, also in a per-class form. This R-CNN network is fine-tuned on the training set using a mini-batch size of 256 in the RoI-centric fashion. For testing, the RPN generates the highest scored 200 proposals for each predicted class, and the R-CNN network is used to update these proposals' scores and box positions.

This method reduces the top-5 localization error to 10.6% (Table 13). This is our single-model result on the validation set. Using an ensemble of networks for both classification and localization, we achieve a top-5 localization error of 9.0% on the test set. This number significantly outperforms the ILSVRC 14 results (Table 14), showing a 64% relative reduction of error. *This result won the 1st place in the ImageNet localization task in ILSVRC 2015.*