使用深度卷积神经网络进行 ImageNet 分类

作者：Alex Krizhevsky，Ilya Sutskever，Geoffrey E. Hinton，多伦多大学
kriz@cs.utoronto.ca，ilya@cs.utoronto.ca，hinton@cs.utoronto.ca

摘要

我们训练了一个大型、深度的卷积神经网络，将 ImageNet LSVRC-2010 比赛中的 120 万张高分辨率图像分类为 1000 个不同的类别。在测试数据上，我们实现了 37.5%的 top-1 错误率和 17.0%的 top-5 错误率，这比以往的最先进技术要好得多。这个神经网络有 6000 万个参数和 65 万个神经元，由五个卷积层组成，其中一些后面跟着最大池化层，以及三个全连接层，最后是一个包含 1000 种类别的 softmax 层。为了加快训练速度，我们使用了非饱和神经元和一个非常高效的 GPU 实现卷积操作。为了减少全连接层的过拟合，我们采用了一种最近开发的正则化方法，称为“dropout”，证明非常有效。我们还将这个模型的一个变体参加了 ILSVRC-2012 比赛，并取得了 15.3%的获胜 top-5 测试错误率，而第二名的错误率为 26.2%。

1 介绍

目前的目标识别方法主要依赖于机器学习技术。为了提高它们的性能，我们可以收集更大的数据集，学习更强大的模型，并使用更好的技术来防止过拟合。直到最近，带标签的图像数据集相对较小，大约数以万计的图像（例如 NORB [16]，Caltech-101/256 [8, 9] 和 CIFAR-10/100 [12]）。对于这样大小的数据集，简单的识别任务可以得到很好的解决，尤其是如果它们通过保留标签的转换进行了增强。例如，MNIST 手写数字识别任务的当前最佳错误率（<0.3%）接近人类表现 [4]。但是，在真实环境中的物体表现出相当大的变化，因此为了学会识别它们，需要使用更大的训练集。事实上，小图像数据集的不足已经被广泛认识到（例如 Pinto 等人 [21]），但是直到最近才有可能收集到包含数百万图像的带标签数据集。新的更大数据集包括 LabelMe [23]，其中包含数十万个完全分割的图像，以及 ImageNet [6]，其中包含超过 1500 万张高分辨率图像，涵盖 2 万多个类别。

为了从数百万张图像中学习成千上万个物体，我们需要一个具有大学习能力的模型。然而，目标识别任务的巨大复杂性意味着即使是像 ImageNet 这样的数据集也无法完全描述这个问题，因此我们的模型还应该具备大量的先验知识，以弥补我们没有的所有数据。卷积神经网络（CNNs）是其中一类模型 [16, 11, 13, 18, 15, 22, 26]。它们的容量可以通过调整它们的深度和广度来控制，并且它们对图像的性质做出了强大且大多数正确的假设（即统计的稳定性和像素依赖的局部性）。因此，与具有类似大小的层的标准前馈神经网络相比，CNNs 的连接和参数要少得多，因此更容易训练，而它们在理论上的最佳性能可能只稍微差一些。

尽管 CNN 具有吸引人的特点，尽管它们的局部架构相对高效，但将它们大规模应用于高分辨率图像仍然是成本高昂的。幸运的是，当前的 GPU 配备了高度优化的 2D 卷积实现，足以支持训练规模较大的 CNN 模型，并且像 ImageNet 这样的最新数据集包含足够的带标签示例，可以在没有严重过拟合的情况下训练这样的模型。

本论文的具体贡献如下：我们在 ILSVRC-2010 和 ILSVRC-2012 竞赛中使用 ImageNet 的子集上训练了迄今为止最大的卷积神经网络，并在这些数据集上取得了迄今为止最好的结果。我们编写了高度优化的 GPU 实现的 2D 卷积和所有其他与训练卷积神经网络相关的

操作，并将其公开提供。我们的网络包含一些新颖且不寻常的特性，可以改善性能并减少训练时间，详细介绍在第 3 节中。由于网络规模较大，即使有 120 万个带标签的训练样例，过拟合仍然是一个显著的问题，因此我们采用了几种有效的技术来防止过拟合，这些技术在第 4 节中有详细描述。我们的最终网络包含五个卷积层和三个全连接层，而这个深度似乎很重要：我们发现移除任何一个卷积层（每个卷积层的参数不超过模型参数的 1%）都会导致性能下降。

最终，网络的规模主要受限于当前 GPU 上可用的内存量以及我们愿意容忍的训练时间。我们的网络在两块 GTX 580 3GB GPU 上训练需要五到六天的时间。我们所有的实验都表明，我们的结果可以通过等待更快的 GPU 和更大的数据集来提高。

2 数据集

ImageNet 是一个包含超过 1500 万张高分辨率图像的数据集，大约涵盖了 2.2 万个类别。这些图像是从网络上收集的，并使用亚马逊的 Mechanical Turk 众包工具由人类标注者进行标注。从 2010 年开始，作为 Pascal 视觉对象挑战的一部分，每年都会举办一个名为 ImageNet 大规模视觉识别挑战的竞赛。

ILSVRC 是指 ImageNet 大规模视觉识别挑战赛。ILSVRC 使用 ImageNet 的一个子集，大约每个类别有 1000 张图像。总共大约有 120 万张训练图像，5 万张验证图像和 15 万张测试图像。

ILSVRC-2010 是唯一具有测试集标签的 ILSVRC 版本，因此这是我们进行大部分实验的版本。由于我们还参加了 ILSVRC-2012 竞赛，因此在第 6 节中，我们还报告了我们在该数据集版本上的结果，该版本的测试集标签不可用。在 ImageNet 上，通常报告两种错误率：top-1 和 top-5，其中 top-5 错误率是指模型认为最有可能的五个标签中没有包含正确标签的测试图像的比例。

ImageNet 的图像分辨率各不相同，而我们的系统需要恒定的输入维度。因此，我们将图像下采样到固定分辨率为 256×256。对于矩形图像，我们首先重新调整图像，使得较短的一边长度为 256，然后从结果图像中裁剪出中心的 256×256 区域。除了从每个像素值中减去训练集上的均值活动之外，我们没有以任何其他方式对图像进行预处理。因此，我们是在（居中的）原始 RGB 像素值上训练我们的网络。

3 架构

我们网络的架构总结如图 2 所示。它包含了八个学习层，其中包括五个卷积层和三个全连接层。接下来，我们将描述我们网络架构中一些新颖或不寻常的特点。3.1-3.4 节按照我们对它们重要性的估计排序，最重要的部分排在最前面。

3.1 ReLU 非线性函数

模拟神经元输出 f 作为其输入 x 的函数的标准方式是使用 $f(x) = \tanh(x)$或 $f(x) = (1 + e^{-x})^{-1}$。在使用梯度下降训练时，这些饱和非线性要比非饱和非线性 $f(x) = \max(0,x)$慢得多。根据 Nair 和 Hinton [20]的研究，我们称具有这种非线性的神经元为修正线性单元（ReLU）。具有 ReLU 的深度卷积神经网络的训练速度是具有 tanh 单元的网络的几倍。这在图 1 中得到了证明，图中显示了在 CIFAR-10 数据集上达到 25%训练误差所需的迭代次数，对于一个特定的四层卷积网络。这个图表表明，如果使用传统的饱和神经元模型，我们将无法对这样大的神经网络进行实验。

我们并不是第一个考虑在 CNN 中使用传统神经元模型以外的替代方案的人。例如，Jarrett 等人[11]声称非线性 $f(x) = |\tanh(x)|$ 在他们的类型的对比度归一化后跟随局部平均池化

在 Caltech-101 数据集上表现特别好。然而，在该数据集上，主要关注的是防止过拟合，因此他们观察到的效果与我们使用 ReLU 时所报告的对训练集的加速拟合能力是不同的。更快的学习对在大型数据集上训练的大型模型的性能有很大影响。

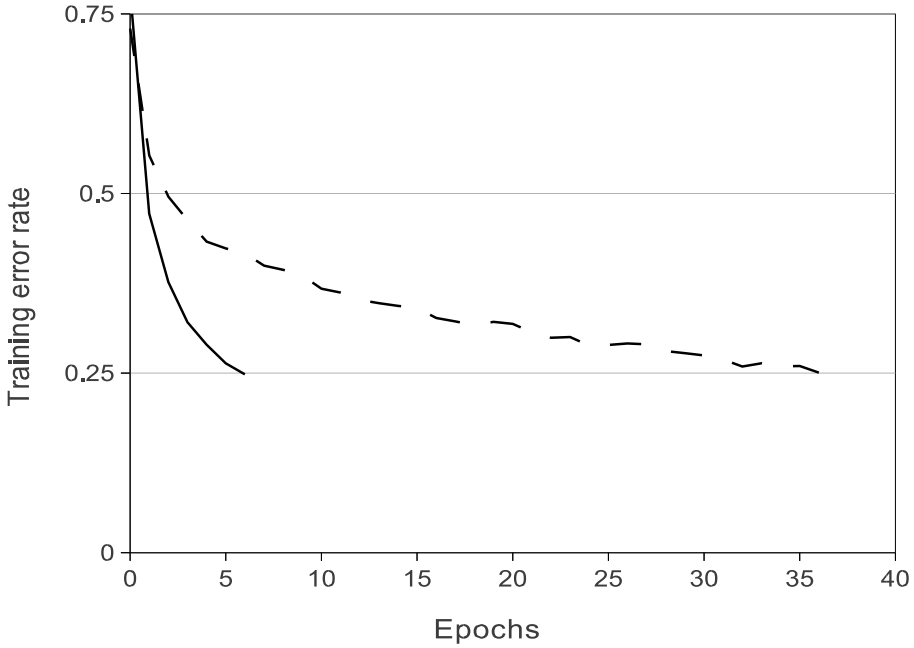

图 1：具有 ReLU 的四层卷积神经网络（实线）在 CIFAR-10 上达到 25%的训练误差率比具有 tanh 神经元的等效网络（虚线）快六倍。每个网络的学习率都是独立选择的，以使训练尽可能快。没有采用任何形式的正则化。这里展示的效果大小随网络架构而异，但具有 ReLU 的网络始终比具有饱和神经元的等效网络学习速度快几倍。

3.2 在多个 gpu 上训练

一块单独的 GTX 580 GPU 只有 3GB 的内存，这限制了可以在其上进行训练的网络的最大尺寸。事实证明，120 万个训练样本足以训练超过一块 GPU 内存容量的大型网络。因此，我们将网络分布在两个 GPU 上进行训练。当前的 GPU 非常适合跨 GPU 并行化，因为它们能够直接读取和写入彼此的内存，而无需经过主机内存。我们采用的并行化方案基本上是将一半的卷积核（或神经元）放在每个 GPU 上，并采用了一个额外的技巧：GPU 之间仅在某些层进行通信。这意味着，例如，第 3 层的卷积核从第 2 层的所有卷积核图中获取输入。然而，第 4 层的卷积核仅从与其所在 GPU 相同的第 3 层的卷积核图中获取输入。选择连接模式是一个交叉验证的问题，但这使我们能够精确调整通信量，使其成为计算量的可接受比例。

我们的网络架构与 Cires¸an 等人[5]使用的“columnar” CNN 的架构有些相似，只是我们的列不是独立的（见图 2）。与在一块 GPU 上训练的每个卷积层具有一半卷积核的网络相比，这种方案将我们的 top-1 和 top-5 错误率分别降低了 1.7%和 1.2%。与一块 GPU 上的网络相比，双 GPU 网络的训练时间略短。

3.3 局部响应归一化

ReLUs 具有一个理想的特性，即它们不需要输入规范化来防止它们饱和。如果至少一些训练样本产生了正的输入给 ReLU，那么该神经元就会发生学习。然而，我们仍然发现以

下的局部归一化方案有助于泛化。用 表示通过在位置(x,y)应用内核 i 计算的神经元的活性，然后应用 ReLU 非线性函数，响应归一化后的活性 bix,y 由以下表达式给出：

$$b_{x,y}^{i} = a_{x,y}^{i} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^{j})^2 \right)^{\beta}$$

其中求和是在相同空间位置的 n 个“相邻”卷积核图之间进行的，N 是该层中卷积核的总数。卷积核图的顺序当然是任意的，并且在训练开始之前确定。这种响应归一化实现了一种受真实神经元启发的侧抑制形式，通过在使用不同卷积核计算的神经元输出之间创建对大活性的竞争。常数 k、n、α 和 β 是超参数，其值是使用验证集确定的；我们使用了 k = 2，n = 5，α = 10^-4 和 β = 0.75。我们在某些层中应用了 ReLU 非线性函数后应用了这种归一化（参见第 3.5 节）。

这种方案在某种程度上类似于 Jarrett 等人[11]的局部对比度归一化方案，但我们的更正确地称为“亮度归一化”，因为我们没有减去平均活性。响应归一化分别将我们的 top-1 和 top-5 错误率降低了 1.4%和 1.2%。我们还在 CIFAR-10 数据集上验证了这种方案的有效性：一个四层 CNN 在没有归一化的情况下达到了 13%的测试错误率，而在进行归一化后降至 11%。

3.4 重叠池化

在 CNN 中，池化层会汇总同一卷积核图中相邻神经元组的输出。传统上，相邻池化单元汇总的邻域不重叠（例如，[17, 11, 4]）。更准确地说，可以将池化层看作由一组间隔为 s 像素的池化单元组成的网格，每个池化单元汇总一个大小为 z×z 的邻域，该邻域位于池化单元的位置中心。如果将 s 设置为 z，则得到常用于 CNN 中的传统局部池化。如果将 s < z，则得到重叠池化。我们在整个网络中使用这种方案，其中 s = 2，z = 3。与产生等效维度输出的非重叠方案 s = 2，z = 2 相比，这种方案将 top-1 和 top-5 错误率分别降低了 0.4%和 0.3%。在训练过程中，我们通常观察到使用重叠池化的模型稍微更难过拟合。

3.5 整体架构

现在我们准备描述我们的 CNN 的整体架构。如图 2 所示，该网络包含八个带权重的层；前五层是卷积层，剩下的三层是全连接层。最后一个全连接层的输出被馈送到一个具有 1000 种类别的 softmax 层，产生了对这 1000 个类别标签的分布。我们的网络最大化多项式逻辑回归目标，这等同于最大化训练案例中正确标签在预测分布下的对数概率的平均值。

整体架构指的是整个神经网络的结构，包括各层的连接方式、层之间的关系以及激活函数的选择等。在描述我们的 CNN 的整体架构时，我们有八个带权重的层，前五层是卷积层，剩下的三层是全连接层。最后一个全连接层的输出被馈送到一个具有 1000 个类别的 softmax 层，产生了对这 1000 个类别标签的分布。我们的网络最大化多项式逻辑回归目标，即最大化训练案例中正确标签在预测分布下的对数概率的平均值。第二、第四和第五个卷积层的卷积核只连接到前一层中与其在同一 GPU 上的卷积核图。第三个卷积层的卷积核连接到第二层的所有卷积核图。全连接层中的神经元连接到前一层中的所有神经元。响应归一化层跟随第一个和第二个卷积层。在第五个卷积层后，以及响应归一化层后，都会接着池化层。ReLU 非线性函数被应用于每个卷积层和全连接层的输出。

第一个卷积层使用 96 个大小为 11×11×3 的卷积核对 224×224×3 的输入图像进行滤波，步长为 4 像素（这是相邻感受野中心之间的距离）。

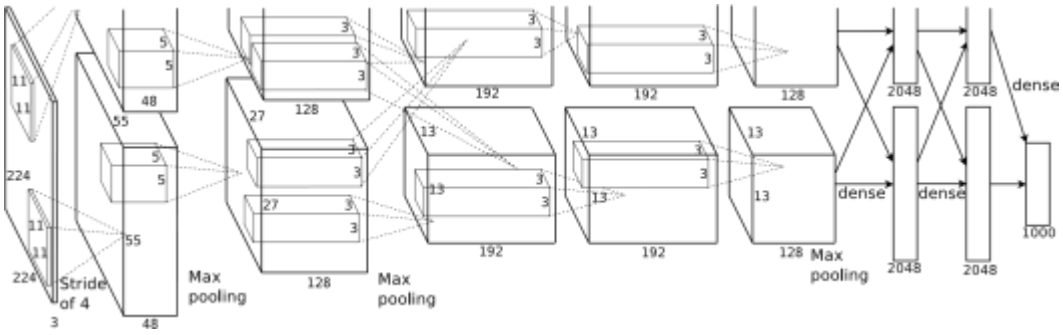

图 2：我们 CNN 的架构示意图，明确显示了两个 GPU 之间责任的划分。一个 GPU 运行图上方的层部分，而另一个 GPU 运行图下方的层部分。GPU 仅在某些层次上进行通信。网络的输入是 150,528 维，网络其余层的神经元数量为 253,440-186,624-64,896-64,896-43,264-4096-4096-1000。

第二个卷积层以第一个卷积层的（响应归一化和池化后的）输出作为输入，并使用 256 个大小为 5×5×48 的卷积核进行滤波。第三、第四和第五个卷积层彼此相连，没有任何中间的池化或归一化层。第三个卷积层有 384 个大小为 3×3×256 的卷积核，连接到第二个卷积层的（归一化、池化后的）输出。第四个卷积层有 384 个大小为 3×3×192 的卷积核，第五个卷积层有 256 个大小为 3×3×192 的卷积核。全连接层每层有 4096 个神经元。

4 减少过拟合

我们的神经网络架构具有 6000 万个参数。虽然 ILSVRC 的 1000 个类别使得每个训练样本对从图像到标签的映射施加了 10 位的约束，但这事实证明对如此多的参数进行学习而不出现过拟合是不够的。接下来，我们将描述我们应对过拟合的两种主要方法。

4.1 数据增强

在图像数据上减少过拟合的最简单和最常见的方法是使用保持标签的转换人为扩大数据集（例如，[25, 4, 5]）。我们采用两种不同形式的数据增强，两种增强方式都可以通过很少的计算从原始图像生成转换后的图像，因此转换后的图像不需要存储在磁盘上。在我们的实现中，转换后的图像是通过 Python 代码在 CPU 上生成的，而 GPU 则在训练之前的图像批次上进行训练。因此，这些数据增强方案在计算上是几乎免费的。

第一种数据增强形式包括生成图像平移和水平反射。我们通过从 256×256 的图像中提取随机的 224×224 补丁（以及它们的水平反射）来实现这一点，并在这些提取的补丁上训练我们的网络。尽管由此产生的训练样本之间存在高度的相互依赖，但这将训练集的大小增加了 2048 倍。如果没有这个方案，我们的网络将会出现严重的过拟合，这将迫使我们使用规模小得多的网络。在测试时，网络通过提取五个 224×224 的补丁（四个角落补丁和中心补丁）以及它们的水平反射（因此总共有十个补丁），并对网络 softmax 层在这十个补丁上的预测进行平均，来进行预测。

第二种数据增强形式是在训练图像中改变 RGB 通道的强度。具体而言，我们对整个 ImageNet 训练集的 RGB 像素值进行 PCA 处理。对于每个训练图像，我们添加一定倍数的主成分，其大小与相应的特征值乘以从均值为零、标准差为 0.1 的高斯分布中抽取的随机变量成比例。因此，对于每个 RGB 图像像素 $I_{xy} = [I_{xy}^R, I_{xy}^G, I_{xy}^B]^T$ 我们添加以下数量：

$$[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$$

其中，pi 和 λi 分别是 RGB 像素值的 3×3 协方差矩阵的第 i 个特征向量和特征值，αi 是前面提到的随机变量。对于特定训练图像的所有像素，每个 αi 只抽取一次，直到该图像再

次用于训练时，才重新抽取。这个方案大致捕捉了自然图像的一个重要特性，即物体的身份对光照强度和颜色的变化是不变的。这个方案将 top-1 错误率降低了超过 1%。

4.2 随机失活正则化

将许多不同模型的预测组合起来是减少测试错误的一种非常成功的方法[1, 3]，但对于已经需要数天才能训练的大型神经网络来说，这种方法似乎太昂贵了。然而，有一种非常高效的模型组合方法，它在训练期间只增加了大约一倍的成本。这种最近引入的技术称为“dropout”[10]，它的作用是以 0.5 的概率将每个隐藏神经元的输出设为零。以这种方式“丢弃”的神经元不参与前向传播，也不参与反向传播。因此，每次输入被呈现时，神经网络都会对不同的架构进行采样，但所有这些架构共享权重。这种技术减少了神经元之间的复杂相互适应，因为一个神经元不能依赖于特定其他神经元的存在。因此，它被迫学习更加健壮的特征，这些特征在与其他神经元的许多不同随机子集一起使用时是有用的。在测试时，我们使用所有神经元，但将它们的输出乘以 0.5，这是对由成千上万个 dropout 网络产生的预测分布取几何平均的合理近似。

我们在图 2 的前两个全连接层中使用了 dropout。如果没有使用 dropout，我们的网络会出现严重的过拟合。使用 dropout 会大致使收敛所需的迭代次数加倍。.

5 学习过程中的细节

我们使用批量大小为 128 的随机梯度下降法训练我们的模型，动量为 0.9，权重衰减为 0.0005。我们发现这个小量的权重衰减对模型的学习非常重要。换句话说，在这里，权重衰减不仅仅是一种正则化方法：它还减少了模型的训练误差。权重 w 的更新规则为

$$v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$$

$$w_{i+1} := w_i + v_{i+1}$$

其中，i 是迭代索引，v 是动量变量，ε 是学习率，$\left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$ 是对于第 i 批次 Di 的目标函数关于 w 的导数在 wi 处的平均值

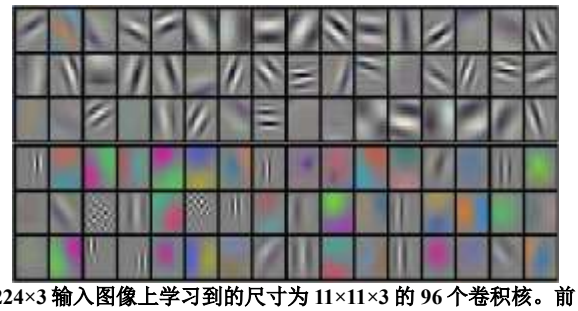

图 3：第一卷积层在 224×224×3 输入图像上学习到的尺寸为 11×11×3 的 96 个卷积核。前 48 个卷积核是在 GPU 1 上学习的，而后 48 个卷积核是在 GPU 2 上学习的。详细信息请参见第 6.1 节。

我们从均值为零、标准差为 0.01 的高斯分布中初始化了每一层的权重。我们将第二、第四、第五个卷积层以及全连接隐藏层中的神经元偏置初始化为常数 1。这种初始化通过为 ReLU 提供正输入加速了学习的早期阶段。我们将其余层中的神经元偏置初始化为常数 0。

我们对所有层使用相同的学习率，在整个训练过程中手动进行调整。我们遵循的启发式方法是，当验证错误率在当前学习率下不再改善时，将学习率减小 10 倍。学习率初始值

为 0.01，在终止之前减小了三次。我们通过 120 万张图像的训练集大致训练了 90 个周期，这在两块 NVIDIA GTX 580 3GB GPU 上花费了五到六天的时间。

6 结果

我们在ILSVRC-2010上的结果总结在表1中。我们的网络实现了37.5%的top-1和17.0%的 top-5 测试集错误率。ILSVRC2010 竞赛期间取得的最佳性能为 47.1%和 28.2%，采用了对六个在不同特征上训练的稀疏编码模型产生的预测进行平均的方法[2]。自那时以来，最佳的已发布结果为45.7%和25.7%，采用了对从两种密集采样特征计算得到的Fisher Vectors（FVs）训练的两个分类器产生的预测进行平均的方法[24]。

我们还参加了 ILSVRC-2012 竞赛，并在表 2 中报告了我们的结果。由于 ILSVRC-2012 测试集标签并未公开，我们无法报告我们尝试的所有模型的测试错误率。在本段剩余部分中，我们将验证和测试错误率互换使用，因为根据我们的经验，它们的差异不超过 0.1%（请参阅表 2）。本文描述的 CNN 实现了 18.2%的 top-5 错误率。对五个类似的 CNN 的预测进行平均，得到了 16.4%的错误率。训练一个 CNN，在最后一个池化层之上增加一个额外的第六个卷积层，对整个 ImageNet 2011 年秋季发布（1500 万张图像，22,000 个类别）进行分类，然后在 ILSVRC-2012 上进行“微调”，得到了 16.6%的错误率。将两个在整个 2011 年秋季发布上进行了预训练的 CNN 的预测与前述的五个 CNN 的预测进行平均，得到了 15.3%的错误率。第二好的竞赛参赛作品采用了对从不同类型的密集采样特征计算得到的 Fisher Vectors（FVs）训练的多个分类器的预测进行平均的方法，取得了 26.2%的错误率[7]。

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

表 1：ILSVRC 2010 测试集结果比较。斜体字表示其他人取得的最佳结果。

最后，我们还报告了我们在 2009 年秋季版本的 ImageNet 上的错误率，该数据集包含 10,184 个类别和 8.9 百万张图像。在这个数据集上，我们遵循文献中使用一半图像进行训练，一半进行测试的惯例。由于没有建立的测试集，我们的拆分必然与先前作者使用的拆分不同，但这并不会对结果产生明显影响。我们在这个数据集上的 top-1 和 top-5 错误率分别为 67.4%和 40.9%，使用上述描述的网络，但在最后的池化层之上增加了第六个卷积层。这个数据集上的最佳发布结果是 78.1%和 60.9%。

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

表 2：ILSVRC-2012 验证集和测试集上错误率的比较。斜体表示其他人取得的最佳结果。带星号*的模型是“预先训练”用于对整个 ImageNet 2011 秋季版本进行分类。详细信息请参见第 6 节。

6.1 定性评估

图 3 显示了网络的两个数据连接层学习到的卷积核。网络已经学习到了各种频率和方向选择性的卷积核，以及各种色块。请注意两个 GPU 所展示的专业化特性，这是受限连接性在第 3.5 节中描述的结果。GPU 1 上的卷积核主要与颜色无关，而 GPU 2 上的卷积核主要与颜色相关。这种专业化现象在每次运行中都会发生，并且与任何特定的随机权重初始化无关（除了重新编号 GPU）。

在不进行如第 4.1 节所述的对十个补丁的预测进行平均的情况下，错误率分别为 39.0% 和 18.3%。

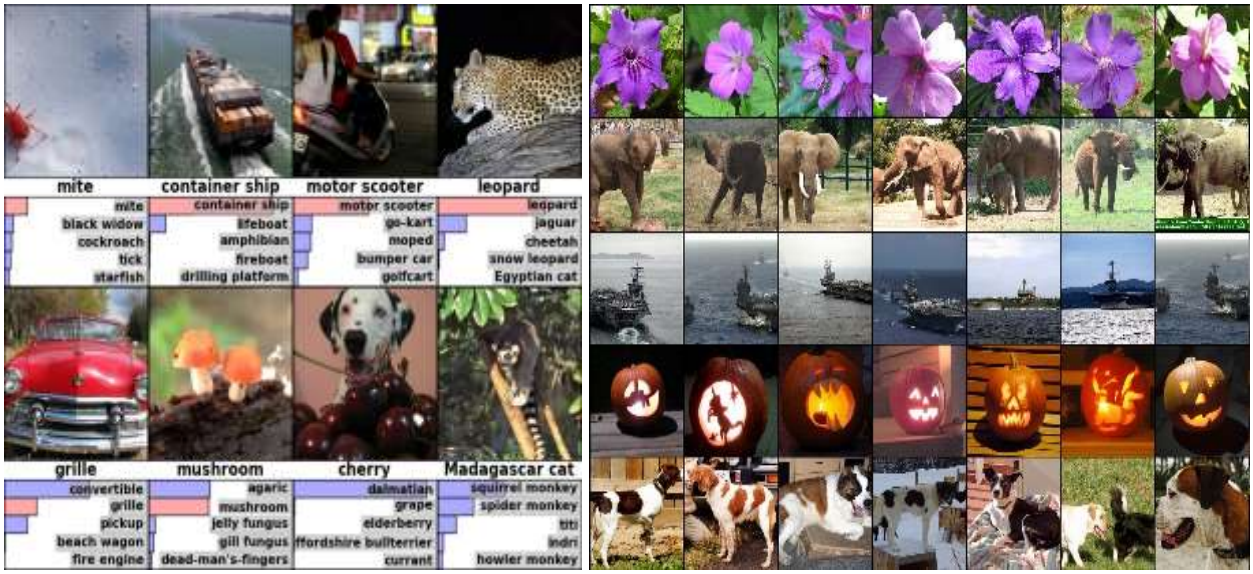

图 4：（左）八个 ILSVRC-2010 测试图像和我们的模型认为最有可能的五个标签。正确的标签写在每张图像下面，并且赋予正确标签的概率也用红色条形图显示（如果它恰好在前 5 名中）。（右）第一列是五个 ILSVRC-2010 测试图像。其余列显示了产生特征向量的六个训练图像，这些特征向量在最后一个隐藏层中与测试图像的特征向量具有最小欧几里得距离。

在图 4 的左侧面板中，我们通过计算网络对八个测试图像的前五个预测来定性评估网络学到了什么。请注意，即使是偏离中心的物体，比如左上角的螨虫，网络也能够识别出来。大多数的前五个标签看起来都是合理的。例如，只有其他类型的猫被认为是豹子的可能标签。在某些情况下（如 grille、cherry），对照片的拍摄焦点存在真正的歧义。

另一种探究网络视觉知识的方法是考虑由图像在最后的 4096 维隐藏层引发的特征激活。如果两个图像产生的特征激活向量具有较小的欧几里得距离，我们可以说神经网络的较高层级认为它们相似。图 4 显示了来自测试集的五个图像以及根据这个度量方式，与每个测试图像最相似的六个来自训练集的图像。请注意，在像素级别上，检索到的训练图像通常与第一列的查询图像在 L2 范数上并不接近。例如，检索到的狗和大象出现在各种姿势中。我们在补充材料中还提供了更多测试图像的结果。

通过计算两个 4096 维实值向量之间的欧几里得距离来计算相似性是低效的，但可以通过训练自动编码器将这些向量压缩为短的二进制编码来实现高效计算。这应该会产生比将自动编码器应用于原始像素[14]更好的图像检索方法，后者不利用图像标签，因此倾向于检索具有相似边缘模式的图像，无论它们是否在语义上相似。

7 讨论

我们的结果表明，一个大型、深层的卷积神经网络能够在一个极具挑战性的数据集上，仅通过监督学习实现创纪录的成绩。值得注意的是，如果去掉一个卷积层，我们网络的性能会下降。例如，去掉任何一个中间层都会导致网络的 top-1 性能损失约 2%。因此，深度对于实现我们的结果是非常重要的。

为了简化我们的实验，我们没有使用任何无监督的预训练，尽管我们预计这将有所帮助，特别是如果我们获得足够的计算能力，可以显著增加网络的规模，而不需要相应增加标记数据的数量。到目前为止，随着我们使网络变得更大并进行更长时间的训练，我们的结果已经有所改善，但为了与人类视觉系统的 infero-temporal 通路相匹敌，我们仍然需要进行大幅度的改进。最终，我们希望在视频序列上使用非常大型和深层的卷积网络，其中时间结构提供了非常有用的信息，在静态图像中缺失或不太明显。

8 参考文献

[1] M. Bell and Y. Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[2] A. Berg, J. Deng, and L. Fei-Fei. Large scale visual recognition challenge 2010. www.imagenet.org/challenges. 2010.

[3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] D. Cires¸an, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. *Arxiv preprint arXiv:1202.2745*, 2012.

[5] D.C. Cires¸an, U. Meier, J. Masci, L.M. Gambardella, and J. Schmidhuber. High-performance neural networks for visual object classification. *Arxiv preprint arXiv:1102.0183*, 2011.

[6] 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.

[7] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. *ILSVRC-2012*, 2012. URL http://www.image-net.org/challenges/LSVRC/2012/.

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[9] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. URL http://authors.library.caltech.edu/7694.

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

[11] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *International Conference on Computer Vision*, pages 2146–2153. IEEE, 2009.

[12] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, 2009.

[13] A. Krizhevsky. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 2010.

[14] A. Krizhevsky and G.E. Hinton. Using very deep autoencoders for content-based image retrieval. In *ESANN*, 2011.

[15] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, et al. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, 1990.

[16] Y. LeCun, F.J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–97. IEEE, 2004.

[17] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256. IEEE, 2010.

[18] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.

[19] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Metric Learning for Large Scale Image Classification: Generalizing to New Classes at Near-Zero Cost. In *ECCV - European Conference on Computer Vision*, Florence, Italy, October 2012.

[20] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. 27th International Conference on Machine Learning*, 2010.

[21] N. Pinto, D.D. Cox, and J.J. DiCarlo. Why is real-world visual object recognition hard? *PLoS computational biology*, 4(1):e27, 2008.

[22] N. Pinto, D. Doukhan, J.J. DiCarlo, and D.D. Cox. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS computational biology*, 5(11):e1000579, 2009.

[23] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008.

[24] J. Sánchez and F. Perronnin. High-dimensional signature compression for large-scale image classification. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1665–1672. IEEE, 2011.

[25] P.Y. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, volume 2, pages 958–962, 2003.

[26] S.C. Turaga, J.F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H.S. Seung. Convolutional networks can learn to generate affinity graphs for image segmentation. *Neural Computation*, 22(2):511–538, 2010.