基于深度卷积神经网络的图像分类（AlexNet）

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton

**摘 要**

我们训练了一个大型深度卷积神经网络，将ImageNet LSVRC-2010竞赛数据集中的120万张高清图片分到1000个不同的类别中。在测试数据中，我们将Top-1错误（分配的第一个类错误）和 Top-5错误（分配的前五个类全错）分别降到了37.5%和 17.0%，这比之前的所有算法都要好得多。这个神经网络拥有6千万的参数和65万个神经元，包括五个卷积层，某些卷积层后面还有最大池化层以及3个带有1000维softmax的全连接层。为了让训练速度更快，我们使用非饱和神经元，并使用高效的 GPU来实现卷积操作。为了减少全连接层的过拟合，我们采用了一种最近研发出来的正则化方法——dropout，结果证明它十分有效。我们也在比赛中加入了这一模型的变体，将TOP-5错误率降到了15.3%，以远远优于第二名26.2%的成绩获胜。

**1 简介**

当前的目标识别方法主要是用机器学习方法，为了提高算法性能，我们可以收集更大的数据集，训练更强大的模型，并用更好的技术来避免过拟合。直到最近，有标注的图像数据集都是相对较小的，一般只有万张的数量级（例如，NORB[16]，Caltech-101/256 [8, 9]和CIFAR-10/100 [12]）。简单的识别任务在这个数量级的数据集上可以得到很好地解决，尤其是在通过标签保留变换进行数据增强的情况下。例如，目前在MNIST数字识别任务上（<0.3%）的最好准确率已经接近了人类水平。但真实环境中的对象表现出了相当大的可变性，因此为了学习识别它们，有必要使用更大的训练数据集。实际上，使用小数据集的缺陷已经被普遍认同了（例如，Pinto et al. [21]），但收集上百万图像的标注数据是在最近才变得的可能。这些新的大型数据集包括LabelMe [23]（包含大量被完全分割的图片），还有 ImageNet【6】（包含了2.2万个类别上的超过1500万张标注的高分辨率的图像）。

为了从百万张图片中学习到数千个物体，我们需要一个有强大学习能力的模型。然而，目标识别任务极高的复杂度意味着，即使拥有ImageNet这么大的数据集，这个问题也很难被具体化，因此，我们的模型也需要大量先验知识去补全所有缺失数据。卷积神经网络（CNNs）就是一种这样的模型[16, 11, 13, 18, 15, 22, 26]。它们的学习能力可以通过改变网络的深度和广度来调整，它们也可以对图片的本质（高层属性）做出强大而且基本准确的假设（统计的稳定性和像素依赖的局部性）。因此，与同样层数的标准前馈神经网相比，CNNs有更少的连接和参数，所以更易于训练，而且CNNs的理论最佳表现仅比前馈神经网络稍差一点。

虽然CNNs效果很好，而且对于局部架构非常高效，但将它们大规模的应用到高分辨率图像中仍然是极其昂贵的。幸运的是，目前的GPU搭配了高度优化的2D卷积实现，强大到足够促进有趣的大规模的CNN训练。而且最近的数据集比如ImageNet又包含了足量的有标注的样本，可以用来训练这些模型，又不会发生太严重的过拟合。

本文的主要贡献包括：我们在ILSVRC-2010和ILSVRC-2012[2]的ImageNet子集上训练了到目前为止最大的神经网络之一，并取得了迄今为止在这些数据集上报道过的最好结果。我们编写了一个高度优化的 2D 卷积的 GPU 实现，以及其他所有训练 CNNs 的训练卷积神经网络内部的操作，并将其公之于众。我们的网络包含一系列新的不同凡响的特征，这提高了网络的性能并减少了训练时间，具体情况在第三章介绍。即使使用了120万标注的训练样本，我们的网络尺寸仍然使过拟合成为一个明显的问题，因此我们使用了一系列技术手段来防止过拟合，详见第四节。我们的网络最终包含5个卷积层和 3个全连接层，这个深度也是很重要的：我们发现去掉任意一个卷积层都会导致更差的表现，即使每个卷积层仅包含不到 1%的模型参数。

最后，网络的尺寸主要受限于GPU的内存容量，以及我们可忍受的训练时间。我们的网络需要在两台GTX580 3GB GPUs上训练五至六天。我们所有的实验都表明，如果有更快的 GPU和更大的数据集出现，其结果能够被进一步提高。

**2 数据集**

ImageNet 是一个拥有超过 1500 万张带标注的高清图片的数据集，这些图片大约属于2.2万个类别。这些图片收集自网络并由亚马逊的Turk群智工具进行人工标记。从2010年开始，作为Pascal视觉对象挑战赛的一部分，每年都会举办ImageNet大规模视觉识别挑战赛（ILSVRC）。ILSVRC 使用ImageNet的一个子集，这个子集包含大约1000个类别，每个类别大概包涵1000张图。总共大概有120万张训练图片，5万张验证图片和15万张测试图片。

ILSVRC-2010是ILSVRC竞赛中唯一可以获得测试集标签的版本，因此我们大多数实验都是在这个版本上运行的。由于我们也使用我们的模型参加了ILSVRC-2012竞赛，因此在第六节我们也报告了模型在这个版本的数据集上的结果，但这个版本的测试标签是不可获得的。在ImageNet上，通常检验两类错误率：TOP‐1和TOP-5，TOP-5错误率表示测试图像的正确标签不在模型认为的五个最可能的便签之中。

ImageNet包含各种清晰度的图片，而我们的系统要求输入维度恒定，因此，我们对图片进行下采样，获得固定大小的256x256的分辨率。对于每张给定的长方形的图，我们首先缩放图像短边长度为256，然后取中心区域的256x256像素。除了在训练集上对像素减去平均活跃度外，我们不对图像做任何其它的预处理。因此我们在原始的RGB像素值（中心的）上训练我们的网络。

**3 模型体系结构**

网络的体系结构如图2，它包含8个学习层——5个卷积层和3个全连接层。接下来，我们讨论我们网络中创新的不寻常的结构。3.1~3.4节将按照我们心目中对它们重要性的评估进行排序，越重要越靠前。

3.1 ReLU 非线性

将神经元输出f建模为输入x的函数的标准方式是用f(x) = tanh(x)或f(x) = (1 + e−x)−1。考虑到梯度下降的训练时间，这些饱和的非线性比非饱和非线性f(x) = max(0,x)更慢。根据Nair和Hinton[20]的说法，我们将这种非线性神经元称为修正线性单元(ReLU)。采用ReLU的深度卷积神经网络训练时间比等价的tanh单元要快几倍。在图1中，对于一个特定的四层卷积网络，在CIFAR-10数据集上达到25%的训练误差所需要的迭代次数可以证实这一点。这张图显示，如果我们采用传统的饱和神经元， 我们将不可能为这项工作训练如此庞大的神经网络。

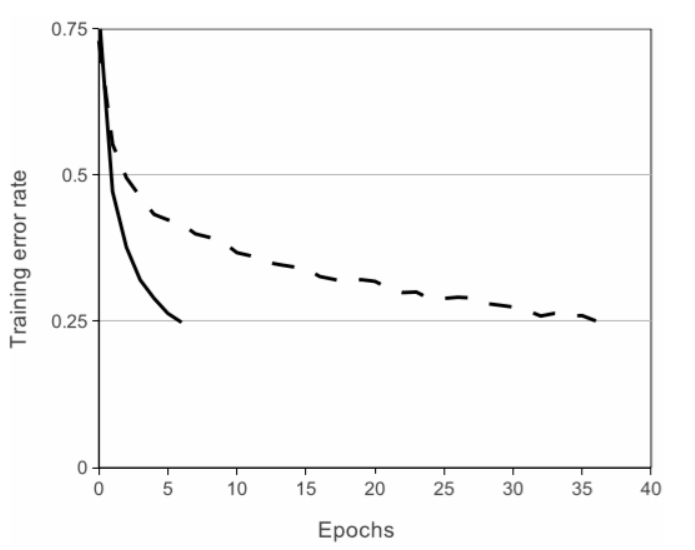


图1：使用ReLU的四层卷积神经网络在CIFAR-10数据集上达到25%的训练误差比使用tanh神经元的等价网络（虚线）快六倍。为了使训练尽可能快，每个网络的学习率是单独选择的。没有采用任何类型的正则化。影响的大小随着网络结构的变化而变化，这一点已得到证实，但使用ReLU的网络都比等价的饱和神经元快几倍。

我们并不是最早考虑替换传统 CNN 神经元模型的人。例如，Jarrett等人声称非线性函数f (x) = |tanh(x)|在Caltech-101数据集上上做对比度归一化和局部均值池化工作表现很好。然而，关于这个数据集的主要问题是要防止过拟合，因此他们观测到的影响不同于我们使用ReLU拟合数据集时的加速能力。更快的学习对于在大型数据集上训练大型模型的表现有重大影响。

3.2 多 GPU 并行训练

单个GTX 580 GPU只有3G内存，这会限制能够在其上进行训练的网络的最大尺寸。事实证明，120万张训练样本图足够用来训练网络，但这个网络过于庞大，以致于一个GPU难以实现。因此我们将网络部署在两个GPU上。现在的 GPU 非常适合做跨 GPU 并行运算，因为它们可以不通过主机内存而直接向彼此的内存做读写操作。我们采用的这种并行模式主要是每个GPU放置一半的核（或神经元），除此之外还有一个小技巧：只在某些特定的层上进行GPU通信。这意味着，比如，第3层的核会将第2层的所有核映射作为输入，而第4层的核只将位于同一GPU上的第3层的核映射作为输入。连接模式的选择是一个交叉验证问题，但这允许我们精确调整连接的数量，直到计算量落入一个可接受的范围内。

除了我们的列不是独立的之外（看图2），由此产生的架构有点类似于Ciresan等人开发的“columnar” CNN。与每个卷积层一半的核在单GPU上训练的网络相比，这个方案将我们的TOP-1错误和TOP‐5错误分别降低1.7%和1.2%。双GPU网络比单GPU网络减少了训练时间。

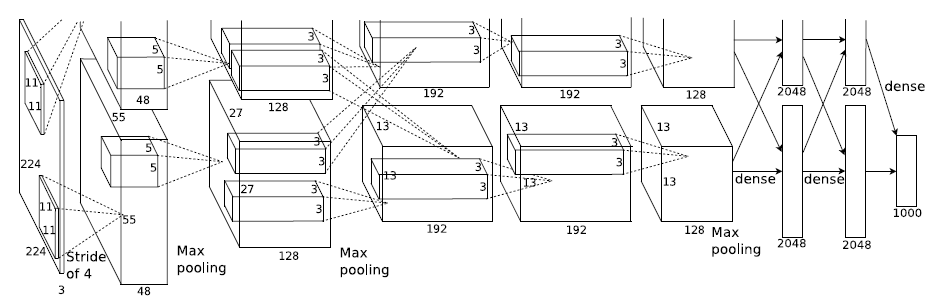
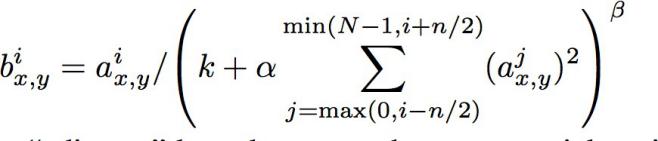


图 2：CNN架构图解，明确地描述了两个GPU之间的对应关系。一个GPU运行某一个层（图的上部分），同时，另一个GPU运行另一些层（图的下部分）。两个 GPU 只在特定的层进行通讯。网络的输入是150,528维的，而除输入层外，余下五个卷积层和三个全连接层的神经元数目分别是253,440 - 186,624 - 64,896 - 64,896 - 43,264 - 4096 - 4096 - 1000。

3.3 局部响应归一化

ReLUs有一个很赞的属性，它们无需对输入数据进行归一化来避免饱和。如果至少有一些训练样例为ReLU产生了正输入，那么这个神经元就会进行学习。然而，我们还是发现下面这种局部响应归一化的模式能够更好地泛化。设由第i个卷积核计算(x, y)位置的ReLU非线性的活动为1540263225(1)，响应归一化结果的计算公式如下：



其中，累加公式中的n表示n个“毗邻的”核映射，N是本层的卷积核数目。核映射的顺序是任意的，并且是在训练之前就定好了的。在训练开始前确定。响应归一化的顺序实现了一种侧抑制形式，灵感来自于真实神经元中发现的类型，模仿生物神经元的横向抑制，为让神经元在利用不同卷积核进行计算的大规模活动中产生竞争。常数 k，n，α和β是超参数，它们的值由验证集决定：我们取k = 2， n=5，α=0.0001，β=0.75。我们在特定层使用 ReLU非线性之后应用这种归一化（详见 3.5）。

这种方案与Jarrett等人提出的“局部对比度归一化”有点类似，但我们的方法更准确的描述应该叫“亮度归一化”，因为我们并不减去均值。通过响应归一化将我们的TOP-1和TOP-5错误分别降低了1.4%和1.2%。我们还在CIFAR‐10数据集上验证了该方案的效果：四层的CNN不用归一化的错误率为13%，用了之后降到了11%。

3.4 重叠池化

CNN中的池化层归纳了同一核映射上相邻组神经元的输出。一般地，相邻池化单元归纳的区域是不重叠的（例如[17, 11, 4]）。更精确地说，一个池化层可以看做是由相隔s个像素占据的池化单元组成的网格所构成，每个单元负责对相邻的z\*z范围的中心区域求和。若设s=z，我们就能够获得用于大多数 CNN 的传统的局部池化方法。若设 s<z，我们就得到了有重叠的池化。我们在自己的网络中使用的方法是设置s=2，z=3。与无重叠的 s=z=2 相比，这一方案在产生相同维度的输出时分别将 TOP-1和TOP-5降低了0.4%和0.3%。而且我们在训练过程中观察采用重叠池化的模型，发现它更难过拟合。

3.5 整体结构

现在描述我们CNN的整体架构。如图2，这个网络包含8个加权的层：前五个是卷积层，后三个是全连接层。最后一层全连接层的输出是1000维softmax的输入，softmax会产生1000类标签的分布。我们的网络最大化多项逻辑回归的目标，这等价于最大化预测分布下训练样本正确标签的对数概率的均值。

第2，4，5卷积层的核只与位于同一GPU上的前一层的核映射相连接（看图2）。第3卷积层的核与第2层的所有核映射相连。全连接层的神经元与前一层的所有神经元相连。第1，2卷积层之后是响应归一化层。3.4节描述的这种最大池化层在响应归一化层和第5卷积层之后。ReLU非线性应用在每个卷积层和全连接层的输出上。

第一个卷积层的输入是224×224×3（3表示RGB）的图像，然后用96个11×11×3的步长为4像素的卷积核进行滤波（步长是相邻神经元感知区域中心之间的距离）；第二个卷积层将第一个卷积层的输出作为输入（反应归一化并池化），然后用256个5×5×48的卷积核进行滤波；第三、四、五层卷积层前后相连，之间没有池化层和归一化层。第三个卷积层有384个3×3×256的卷积核，连接着第二个卷积层的输出（归一化+池化）。第四个卷积层有384个3×3×192的卷积核，第五个卷积层有256个3×3×192的卷积核。每个全连接层各有4096个神经元。

**4减少过拟合**

我们的神经网络拥有6000万的参数，虽然ILSVRC的1000个类别将从图片到标签的映射限制在10个bits，这依然不足以训练这么多的参数而不造成过拟合。下面，我们将介绍两种对付过拟合的基本方法。

4.1数据集放大

最简单最常用的减少过拟合的方法是使用标签保留变换（例如[25,4,5]）来人工增大数据集。我们采取了两种不同形式的数据放大，它们都允许在仅对原图做少量计算的情况下产生变形的新图，所以变形后的新图无需存储在硬盘中。在我们的实现中，变换的新图由Python在CPU上计算产生，与此同时，GPU仍在训练前一批图像。所以这种放大数据集的方式是免费、高效且节省计算资源的。

第一种放大数据集（产生新图）的方式包括图像变换和水平镜像。我们从256x256的图片中随机抽取224x224的区块（及其水平镜像）来实现这种方法，并在这些抽取后得到的区块上训练我们的神经网络。这种方法为我们的训练集增加了2048个因子，尽管最终的训练样本是高度相关的。没有这个方案，我们的网络会有大量的过拟合，这会迫使我们使用更小的网络。在测试时，网络会提取5个224×224的图像块（四个角上的图像块和中心的图像块）和它们的水平翻转（因此总共10个图像块）进行预测，然后对网络在10个图像块上的softmax层进行平均。

第二种放大数据集的方法是改变训练图像的RGB通道的强度。具体来说，我们在整个ImageNet训练集上对RGB像素进行主成分分析（PCA），对于每张训练图像，我们通过均值为0，方差为0.1的高斯分布产生一个随机值（设为a），然后通过向图像中加入更大比例的（相应的本征值的a倍），把其主成分翻倍。因此，对于每个RGB像素，我们加入值：

，

其中，pi和λi分别是第i个特征向量和第i个3x3RGB协方差矩阵的本征值。而αi是前面所述的随机变量。对于一张特定的训练图片的所有像素，每个αi仅被抽取一次，直到这张图再次被用于训练才会再次提取随机变量。这一方案能够近似地捕捉原始图像的一些重要特征，即那些不随光线强度与颜色变化的物体特质。该方案把Top-1错误率降低了1%。

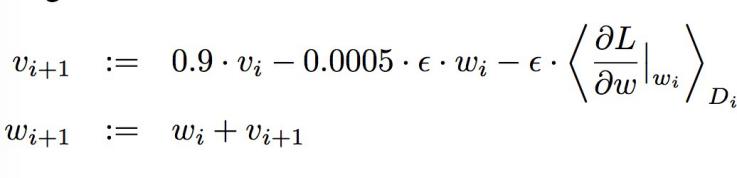
4.2丢弃失活

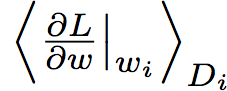
将许多不同模型的预测结合起来是降低测试误差[1,3]的一个非常成功的方法，但这种方法对于需要好几天去训练的大型神经网络来说似乎太昂贵了。然而，有一种非常高效的模型联立方法，只需要在训练过程中消耗一到两个因子。这种新近研究出来的技术叫做“DROPOUT”，它会以50%的概率将每个隐藏层神经元置零。以这种方法被置零的神经元不再参与前馈和BP过程。因此每次输入时，神经网络会采样一个不同的架构，但所有架构共享权重。这种技术降低了神经元间相互适应的复杂性，因为每个神经元都不可能依赖其他特定某个神经元的表现。因此，模型学习的特征更鲁棒，使之能够被许多不同的随机神经元子集使用。在测试中，我们使用所有的神经元，把它们的输出乘以0.5，这是一种对大量dropout网络产生的预测分布的几何均值的合理近似。

我们在图2中的前两个全连接层使用dropout。否则，我们的网络会表现出严重的过拟合。dropout失活大致上使要求收敛的迭代次数翻了一倍。

**5学习过程的细节**

我们使用随机梯度下降来训练我们的模型，样本的batchsize为128，动量为0.9，权重衰减为0.0005。我们发现小的权重衰减对于模型学习很重要，换句话说，权重衰减不仅仅是一个正则项：它减少了模型的训练误差。权值w的更新规则是：



其中，i是迭代次数，v是动量变量，ε是学习速率，是第i批次的目标函数关于w的导数（wi的偏导数）Di的平均值。

我们将每一层的权值利用均值为0，方差为0.01的高斯分布随机初始化，我们用常数1初始化第2、4、5卷积层和全连接隐藏层的偏置神经元（常数单元）。这种初始化通过向ReLUs提供正输入，加速了学习的早期过程。我们将其它层的偏置神经元初始化为0。

在整个学习过程中，我们在所有层都使用人工调整的相等的学习速率。我们采用的启发式方法是当验证误差不在降低时，就把当前的学习速率除以10。学习速率初始化为0.01，并在结束前减小3次。（做三次除以10）我们大概用120万张图片把我们的网络训练了约90个循环，在两个NVIDIAGTX5803GBGPU上花费了5到6天。

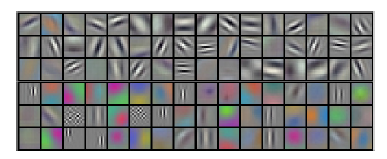


图3：96个通过第一个卷积层学习224x224x3的图片得到的11x11x3的卷积卷积核。上面48个和下面48个分别由两个GPU学习得到，详见6.1.

**6实验结果**

我们在ILSVRC‐2010数据集上的实验结果归纳在表1里。我们的网络top‐1和top‐5测试误差分别是37.5%和17.0%。在ILSVRC-2010竞赛中最佳结果是top‐1和top‐5测试误差分别是47.1%和28.2%，使用的方法是对6个在不同特征上训练的稀疏编码模型生成的预测进行平均，从那时起已公布的最好结果是top‐1和top‐5测试误差分别是45.7%和25.7%，使用的方法是平均在Fisher向量（FV）上训练的两个分类器的预测结果，Fisher向量是通过两种密集采样特征计算得到的。

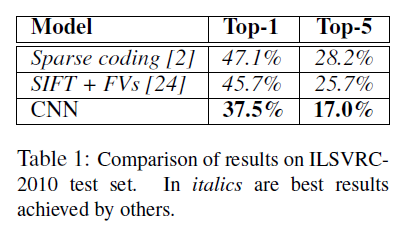


表1：ILSVRC-2010测试集上的结果对比。斜体是其它人取得的最好结果。

我们也让我们的模型参加了ILSVRC-­‐2012的比赛，并在表2中展示了我们的结果。因为ILSVRC-­‐2012测试集的标签并未公开，所以我们不能报告我们所有试过的模型的测试错误率。在这一段的余下部分，我们使用验证误差代替测试误差，因为根据我们的经验，它们的差距不会大于0.1%（见表2）。本文介绍的卷积神经网络达到了Top‐5错误18.2%的水平。5个相同CNN平均TOP‐5错误为16.4%。训练一个比之前说的五个卷积层还多一个卷积层的CNN去分类整个ImageNetFall2011数据集（1500万张图，22000个类别），然后对其进行调整，在ILSVRC‐2012上可以达到16.6%的TOP‐5错误。两个在ImageNetFall2011数据集上预训练的CNN，加上前面提到的五个CNN，平均TOP‐5为15.3%。比赛的第二名达到了26.2%的TOP‐5，他们用的是对几个在特征取样密度不同的Fisher向量（FV）上训练的分类器的预测结果取平均的方法。

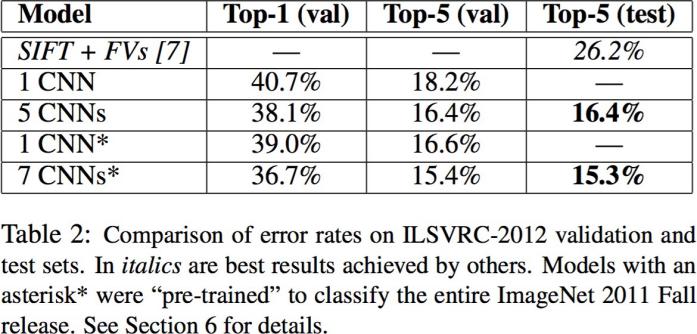


表2：ILSVRC-2012验证集和测试集的误差对比。斜线部分是其它人取得的最好的结果。带星号的是“预训练的”对ImageNet2011秋季数据集进行分类的模型。更多细节请看第六节。

最后，我们也报告了我们在ImageNet2009秋季数据集上的误差率，ImageNet2009秋季数据集有10184个类，890万图像。在这个数据集上我们按照惯例用一半的图像来训练，一半的图像来测试。由于没有建立测试集，我们的数据集分割有必要不同于以前作者的数据集分割，但这对结果没有明显的影响。我们在这个数据集上的的top-1和top-5错误率是67.4%和40.9%，使用的是上面描述的在最后的池化层之后有一个额外的第6卷积层网络。这个数据集上公开可获得的最好结果是78.1%和60.9%。

6.1定量分析

图3显示了网络的两个数据连接层学习到的卷积核。网络学习到了大量的频率核、方向选择核，也学到了各种颜色点。注意两个GPU表现出的专业化，3.5小节中描述的受限连接的结果。GPU1上的核主要是没有颜色的，而GPU2上的核主要是针对颜色的。这种专业化在每次运行时都会发生，并且是与任何特别的随机权重初始化（以GPU的重新编号为模）无关的。

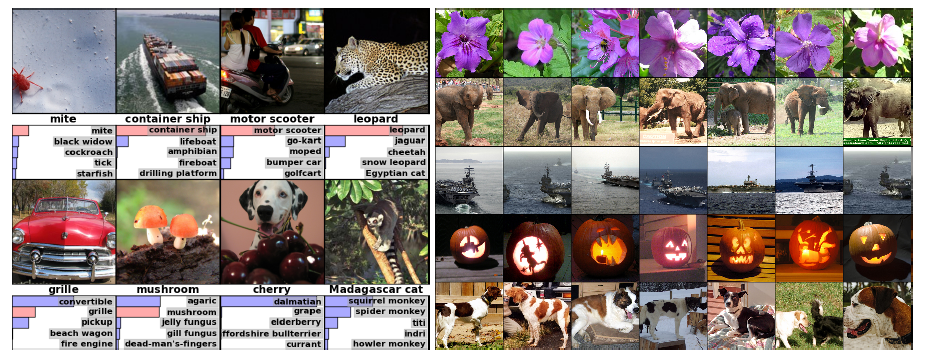


图4：（左）8张ILSVRC-2010测试图像和我们的模型认为最可能的5个标签。每张图像的下面是它的正确标签，正确标签的概率用红条表示（如果正确标签在top5中）。（右）第一列是5张ILSVRC-2010测试图像。剩下的列展示了6张训练图像，这些图像在最后的隐藏层的特征向量与测试图像的特征向量有最小的欧氏距离。

在图4的左边部分，我们定量地展示了对于8张图片网络所学习到的前五个预测。注意即使是不在图像中心的目标也能被网络识别，例如左上角的小虫。大多数的top-5标签似乎是合理的。例如，对于美洲豹来说，只有其它类型的猫被认为是看似合理的标签。在某些案例（格栅，樱桃）中，网络究竟该关注哪个物体确实存在歧义。

另一个研究可视化网络所学知识的方法是考虑最后一个4096维隐层所激活的特征向量。如果两张图的向量欧氏距离很小，我们可以说很大程度上神经网络认为它们是相似的。图4展示了五张测试集中的图片，以及按照上述方法找出的分别与这五张图最相似的6张训练集图片。注意在像素尺度上，检索到的训练图像与第一列的查询图像在L2上通常是不接近的。比如，检索的狗和大象似乎有很多姿态。我们用更多的测试集图片支持证明了这一观点。

通过两个4096维的实数向量之间的欧氏距离来计算相似度显然效率很低，但可以通过训练一个自编码器去把这些向量压缩为二进制编码来提高效率。这应该能够产生一种比对原始像素进行自编码更好的图像检索方法，因为（对原始像素进行自编码）用不到标签，因此它倾向于找出具有同样边缘模式的图片，而不是语义上相似的图。

**7 讨论**

我们的结果显示一个大型深度卷积神经网络能够在一个极具挑战的数据集上进行纯有监督学习可以取得破纪录的结果。值得注意的是，如果把我们的网络去掉一层卷积层，表现就会变差。比如，去掉任意隐藏层会让top‐1错误增加2%，所以深度对于我们的成功真的很重要。

为了简化我们的实验，我们并未使用任何非监督的预训练，即使我们希望他会有帮助，特别是如果我们能够获得足够的计算力去大幅提升网络规模却不相应地增加标注数据的数量。到目前为止，我们的结果已经提高了，因为我们的网络更大、训练时间更长，但为了匹配人类视觉系统的下颞线（视觉专业术语）我们仍然有许多数量级要达到。最后我们想在视频序列上使用非常大的深度卷积网络，视频序列的时序结构会提供非常有帮助的信息，这些信息在静态图像上是缺失的或远不那么明显。