**ImageNet Classification with Deep Convolutional Neural Networks**

**摘要**

我们训练了一个很大的深度卷积神经网络来将ImageNet LSVRC-2010 竞赛上的120万张高分辨率的图片分成了1000中不同的类型。在测试集上，我们取得了top1和top5的错误率分别为37.5%和17.0%，这与之前的最好技术相比要好得多。这个神经网络包含6千万个参数和65,000个神经元，这些参数和神经元组成了5个卷积层，其中一部分卷积层后面跟随着的是最大池化层，即三个全连接层以及一个1000路softmax最终输出层。为了使得训练更加快速，我们使用了非饱和的神经元以及高效的GPU卷积操作实现方式。我们使用了一种名为“dropout”的新的正则化方法来降低全连接层中的过拟合现象，在实际操作中表明这种方法非常有效。我们还使用这个模型的一个变种参加了ILSVRC-2012的竞赛，最终达到了top-5错误率为15.3%，而比赛的亚军的错误率是26.2%。

1. **简介**

现在的目标识别的手段主要是用到了机器学习的方法，为了提高性能，我们可以手机更大的数据集，学习更多强大的模型，以及使用更好的技术来防止过拟合。直到最近，带标签的图片数据集显得相对偏小了—处于万张图片的量级。简单的识别任务使用这种大小的数据集能够很容易的完成，特别是当增加了标签保留转换的时候。比如说，现在最好的基于MNIST的识别的错误率（<0.3%）已经接近于人类的表现了。但是现实环境中的目标会表现的更加多变，因此学习如何识别他们需要更大的数据集。实际上，小规模的图像数据集的缺点已经被大家广泛认识到，但直到最近才有可能收集包含数百万张图片的带标签的数据集。新出现的更大的数据集包括LabelMe（包含成千上万的完全分割的图像）以及ImageNet（包含1500万张超过22,000类带标签的高分辨率图像）。

要从数百万图像中学习数千个目标，我们需要一个具有强大学习能力的模型。然而，目标识别任务的巨大复杂性意味着即使是像ImageNet这样大的数据集也不能解决这个问题，所以我们的模型也应该有很多先验知识来弥补我们没有的所有数据。卷积神经网络（CNN）构成了这样一类模型。他们的能力可以通过改变他们的深度和广度来控制，而且他们也对图像的性质（即统计的平稳性和像素依赖性的局部性）做出强而且大部分正确的假设。因此，与具有相同大小的层的标准前馈神经网络相比，CNN具有更少的连接和参数，因此它们更容易训练，而理论上最好的性能仅略差一些。

尽管CNN具有诱人的特性，尽管本地架构的效率相对较高，但它们在大规模应用于高分辨率图像方面仍然过于昂贵。幸运的是，当前的GPU与高度优化的二维卷积实现配合使用，足以促进大型CNN的训练，最近的数据集（如ImageNet）包含足够多的标记示例来训练此类模型，而不会出现严重的过拟合问题。

本文的具体贡献如下：我们在ILSVRC-2010和ILSVRC-2012比赛中使用的ImageNet子集上训练了迄今为止最大的卷积神经网络之一，并取得了迄今为止这些数据集最好的结果。我们编写了一个高度优化的2D卷积GPU实现方法，以及卷积神经网络训练中固有的所有其他操作，并且我们已经将之公开了。我们的网络包含了许多新的和不同寻常的特性，这些特性可以提高性能并缩短训练时间，详见第3节。我们的网络规模使得我们面临了严重的过拟合问题，即使有120万个标记的训练样例，因此我们使用了各种方法来防止过拟合的出现，详见第4节。我们的最终网络包含5个卷积层和3个完全连接层，这个深度似乎很重要：我们发现去除任何卷积层（每个卷积层包含不超过1%的参数）会导致较差的性能。

最后，网络的规模主要受限于当前GPU上可用的内存以及我们愿意接受的训练时间。我们的网络需要使用片台GTX 580 3GB GPU花费五到六天的时间来训练。所有的实验都表明，只需等待更快的GPU和更大的可使用数据集，我们的结果就可以得到进一步提升。

1. **数据集**

ImageNet是一个超过1500万张22000个类别的带标签的高分辨率图像的数据集。这些图像是从网上收集的，并由人类使用亚马逊的Mechanical Turk群众采购工具进行标记。从2010年开始，作为Pascal视觉对象挑战赛的一部分，每年举办一次名为ImageNet大型视觉识别挑战赛（ILSVRC）的比赛。 ILSVRC使用ImageNet的一个子集，1000个类别中每个类别大约有1000个图像。总共有大约120万个训练图像，50,000个验证图像和15万个测试图像。

ILSVRC-2010是ILSVRC唯一测试集标签可用的一个版本，因此这也是我们进行大部分实验的版本。由于我们也在ILSVRC-2012比赛中使用了我们的模型，我们在第6节报告了这个版本的数据集的结果，而且这个测试集标签是不可用的。在ImageNet上，习惯上报告两个错误率：top-1和top-5，其中top-5错误率是指在模型认为的5个可能的标签中都不包含正确标签的百分数。

ImageNet由多种分辨率图像组成，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一个256×256的固定分辨率。给定一个矩形图像，我们首先重新缩放图像，使短边长度为256，然后从结果中裁剪出中央256×256的补丁图片。除了从每个像素中减去训练集上的平均activity之外，我们不以任何其他方式预处理图像。所以我们在像素的（中心）原始RGB值上训练了我们的网络。

1. **架构**

图2概括了我们网络的架构。它包含八个学习层 - 五个卷积层和三个完全连接层。下面，我们描述一些我们网络架构的新颖或不寻常的特征。 3.1-3.4节按照我们对其重要性的估计进行排序，其中第一个是最重要的。

* 1. **ReLU 的非线性**

神经元根据输入值x的输出函数f一般采用tanh作为激活函数，即或者。当使用梯度下降法寻找最优解时，这类饱和非线性激活函数，相比于非饱和非线性激活函数，会耗费更长的训练时间。借鉴Nair和Hinton [20]的研究，我们采用ReLUs非线性化神经元。采用ReLUs的Deep-CNN比采用tanh的CNN训练起来快很多，如图一所示。我们比较了采用不同激活函数的四层神经网络对CIFAR-10数据集训练，记录分别达到25%训练错误率所需要的循环数。此图表明，如果使用传统的饱和激活函数神经网络模型的话，我们无法完成如此大型的神经网络训练项目。

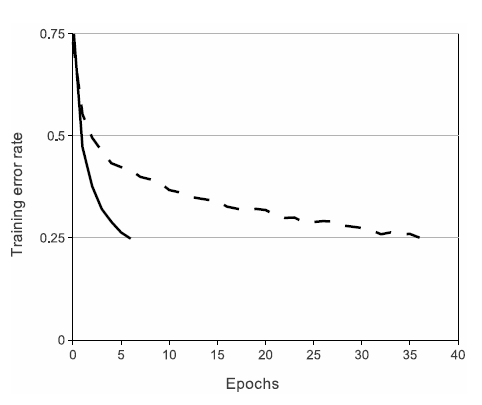


图1：采用ReLUs的四层CNN（实线）对CIFAR-10数据集达到25%训练错误率的速度是采用tanh的CNN（虚线）的六倍。每个网络的学习率（learning rate）都是独立选取以使其训练速度最大化，且都没有经过正则化处理。当然，两种CNN的训练时间上的差距依不同架构会有所不同，但采用ReLUs的CNN总是快过用饱和激活函数的CNN。

不过，我们也并不是最早想到使用非传统神经元模型的团队。Jarrett等人 [11] 曾提出将非线性激活函数运用于他们所构建的模型对Caltech-101数据集的训练取得了良好的效果，该模型包含了对比度归一化层（contrast normalization)及局部平均汇合层（local average pooling）。但训练Caltech-101数据集主要关注的是过拟合问题，所以他们所谓的“良好效果”与我们采用ReLUs提到的“训练速度提升效果” 是有所区别的。更快的训练速度对大型数据集的模型训练效果有很大的影响。

* 1. **多GPU训练**

单个GTX580 GPU只有3GB显存，限制了其所能训练的神经网络的最大规模，而且实践证明120万张训练图像所需要的神经网络对于单个GPU来说过于庞大。因此，我们将神经网络搭建于两个GPU上。我们使用的GPU特别适合并行作业，因为他们能够直接互相读取和写入显存而不需要经过主机内存。通过并行作业，我们在每个GPU上各搭载了一半的神经元，并且设置了一个机制，即两个GPU只在特定的层级中互相“交流”。举例说明，就是某层级接收前一层级所有神经元的输出结果，而另一层级只接收搭载于同一GPU上的上一层级神经元的输出结果。具体连接模式可以通过交叉验证来调整，但这一机制使得我们可以精确地调整连接数，使得其占总计算量的比例达到我们可以接受的程度。

最后我们搭建的架构有一些类似Ciresan等人[5]提出的“柱状”CNN，不过我们的CNN网络的columns之间是非独立的（见图2）。使用两个GPU的神经网络通过运用这一机制与使用单个GPU且卷积层神经元数减半的神经网络相比，Top1错误率及Top5错误率分别低了1.7%和1.5%。而双GPU模型比单GPU模型所用的训练时间还稍短一些。（单GPU模型和双GPU模型的神经元数量其实差不多，因为神经网络大部分的参数集中在第一个全连接层，其接收的是最后一个卷积层的输出结果。所以为了使两种模型具有大致相同数量的参数，我们没有将最后一个卷积层的规模减半，其后的全连接层也不用。这样的处理导致两者的分类效果对比其实是有利于单GPU模型的，因为它的神经元数量比双GPU模型的“一半”多一些。）

* 1. **局部响应规范化**

ReLUs有一个良好的性质，就是它不需要对输入数据进行归一化以防止饱和。如果至少有一部分训练样本提供正的输入值给ReLU，那么训练就会在该神经元内执行。不过，我们还是发现下述局部归一化机制可以提高归纳能力。公式中表示神经元活动，即输入数据的局部(x,y)经过卷积核i处理然后经过ReLU处理。响应归一化活动的计算公式为：

其中，累加项汇总同一空间范围内n个“相邻”的卷积核，N表示该层中的卷积核总数。卷积核map的顺序是随机的且在训练开始前就已定好。这类响应归一化其实是受到真实神经元侧抑制现象的启发在进行类似的操作。其效果就是使不同卷积核计算的神经元输出值之间对计算值比较大的神经元活动(big activities)更为敏感。

式中常数以及都是超参数，用验证集调试得出；我们的模型使用。我们只对某些层（详见3.5小节）在经过ReLU处理后进行上述归一化处理。

这个方法和Jarrett [11]等人使用的局部对比度归一化有一些类似，但是我们的模型更符合"亮度归一化（brightness normalization）"这一范畴，因为我们没有减去均值。响应归一化使得我们模型的Top1错误率及Top5错误率分别降低了1.4%和1.2%。我们还用CIFAR-10数据集验证了响应归一化的效率：四层CNN没有归一化层时错误率为13%，而有归一化层时错误率为11%。

* 1. **重叠池化**

CNN里的汇合层对同一卷积核map内邻近的神经元组的输出值进行总结。传统方法里被汇合层单元处理后的邻近神经元是不重叠的（比如[17,11,4]）。汇合层，具体来说，可以表示为一个网格里相互间隔s个像素的一个个汇合单元，每一个单元处理邻近的以该单元为中心，大小为的神经元组。如果我们设置**s=z**,就得到传统的CNN。如果**s<z**，就得到我们模型使用的重叠汇合。我们的模型设置**s=2, z=3**.如此设置使得我们模型的Top1错误率及Top5错误率相比没有使用重叠汇合的模型（设置**s=2, z=2**）低了0.4%和0.3%，非重叠汇合模型得出的结果的维度和我们模型的一样。我们还在训练过程中发现使用重叠汇合的模型较不容易产生过拟合问题。

* 1. **全局架构**

现在，我们开始介绍CNN模型的整体架构。如图2所示，模型包含八个参数层；前五个是卷积层，后面三个是全连接层。最后一个全连接层的输出结果提供给1000-way softmax，并得出1000个分类标签的概率分布。我们的模型训练目标是最大化多元逻辑斯蒂函数，等价于最大化训练集预测分布下正确分类的log概率的均值。

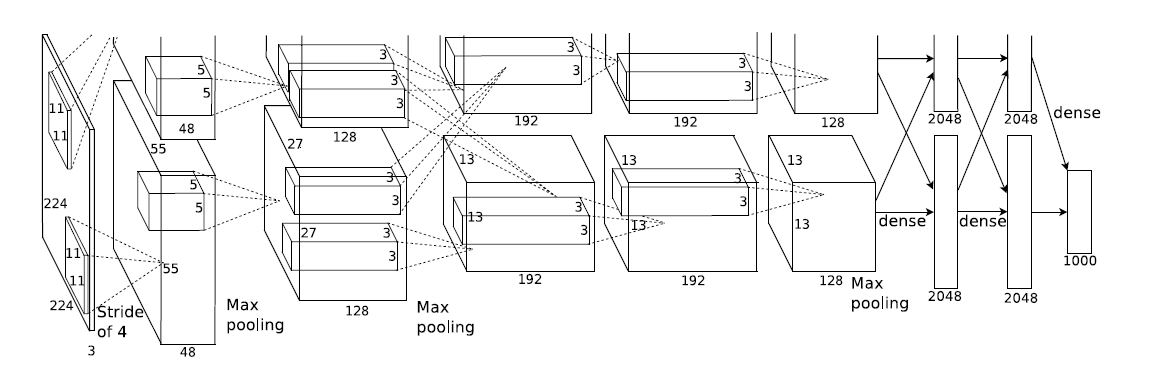


图2：我们的CNN模型的一个图像简述，直观地展示了两个GPU分别负责的任务。图中上下两个部分的卷积层各自搭载于一块GPU上。两个GPU的网络只在部分层次中有交互。神经网络的输入层维数是150,528，网络中剩下的层级的神经元数量分别是253,440、186,624、64,896、64,896、43264、4096、4096、1000.

第一个卷积层用96个大小为的卷积核过滤维度为的输入图像，步长为4像素（步长是指卷积核接收的相邻神经元组感受野中心之间的距离，也就是卷积核每次操作移动的像素数，用于决定输出结果的尺寸）。第一个卷积层的输出结果经过响应归一以及汇合后的结果作为第二卷积层的输入值，第二卷积层使用256个卷积核，大小为5\times 5\times 48.第三、四和五卷积层之间没有其他归一层或汇合层。第三卷积层使用384个卷积核，大小为，连接第二卷积层的输出结果（归一+汇合处理后）。第四卷积层使用384个卷积核，大小为，第五卷积层使用256个卷积核，大小为。每一个全连接层都有4096个神经元。

1. **降低过拟合**

我们的神经网络总共有6000万个参数。虽然ILSVRC的数据集仅有1000个标签类别使得每一个样本对其到所属标签的映射施加了10bits的限制，但即便如此，训练含有如此多参数的模型时，还是会出现较严重的过拟合现象。

* 1. **数据增量**

最简单也最常见的减少过拟合的方法就是通过保留标签转换人为地扩大数据集（例如[25,4,5]）。我们运用两种数据增量方式，计算量都很小，所以转换得到的新图像不用存在硬盘中。我们的转换操作是在CPU上用python实现的，而GPU专门用于训练模型。所以实际训练中，数据增量操作对我们的CNN训练的总计算量没有影响。

第一种数据增量方式是图像变换和水平翻转。具体操作是从原本大小为的图象中提取所有大小为的子图像（以及他们的水平翻转图像），然后将这些子图象作为我们CNN的输入图像。（这解释了为什么图二中我们模型的输入层的大小是）。经过如此操作，我们的训练数据集变为了原来的2048倍。虽然扩大后的数据之间的相关性非常大，但如果不这样操作，我们的网络会出现严重的过拟合现象，可能会迫使我们使用规模更小的网络。在测试的时候，模型对每个输入图像提取五个子图像（四个角落和中心）以及他们分别的水平翻转图像（总共10个），通过softmax层进行预测，并将10个预测值平均。

第二种方式是调整训练图像的RGB各颜色通道强度。具体操作是，对训练数据集所有图像的每个像素RGB值分别进行主成分分析（PCA）。然后将原本的图像加上1）主成分特征向量与2）特征值和3）一个随机量的乘积。也就是对于某图像的每一个像素加上以下算式的结果：

其中和是图像RGB值计算的协方差矩阵的第i个特征向量和特征值，而\alpha\_{i}就是前面提到的随机量，服从均值为0，标准差为0.1的正态分布。随机产生的一组将用于某张图的所有像素，直到该图再次被训练时才会重新产生新的。这一调整是为了突出自然图像的一个重要性质，就是对物体图像的识别不应该受到其表面色彩的强度和颜色的影响。通过该操作，我们CNN的Top1错误率降低了1个百分点。

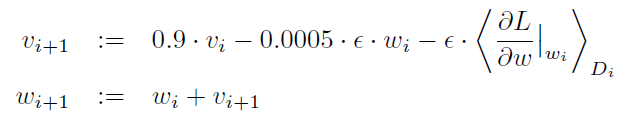
* 1. **随机失活**

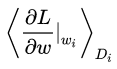
结合多个不同模型的预测结果可以降低测试错误率 [1,3]，但对于本身就需要数天时间训练的大型神经网络而言，这是很奢侈的。然而，还是有很高效的方法能够结合模型的预测结果，而且只耗费大约两倍的训练时间。其中一种最新的方法叫“随机失活”[10]，将隐含层神经元的输出结果依0.5概率设置为0。被随机失活“脱离”的神经元因此不提供有效输出予前向信息传递也不参与后向传播。因此，每一次训练一个图像时，神经网络就会随机生成一个新的架构，但这些架构中使用的权重是一样的。通过随机失活减少了神经元之间复杂的互相适应性（co-adaptation），因为通过随机失活，神经元无法过分依赖于某个有输出结果的前一神经元。在随机神经元组的配合下，这个神经元也因此被迫去学习更加鲁棒且有用的特征。在测试时，我们使用所有的神经元，将他们的输出结果乘以0.5，这其实是由极多的经过随机失活的神经网络产生的平均分类结果的一个合理近似值。

我们在图二中的前两个全连接层理运用随机失活。否则，神经网络训练就会出现很严重的过拟合。但随机失活几乎使得模型收敛所需要的循环翻倍。

1. **学习过程中的细节**

我们用随机梯度下降来训练模型，每一个批量有128个样本，动量为0.9，权值衰减为0.0005。我们发现小权值衰减对模型的训练是很重要的。也就是说，权值衰减在模型中不单单起到正则化作用；它还协助降低模型的训练错误率。权重的更新方法如下：



是循环序数，是动量参数， 是学习率，是第 i个批量样本（128个）上所有目标函数在处对权重的偏导数的均值。

我们将每一个层级的权重初始化为均值0，标准差0.01的正态随机量。第二、四核五卷积层以及全连接层的偏差系数（bias）设置为1。这样可以在训练初期给ReLU单元提供正的输入值，从而加快训练速度。其他层级的偏差系数初始设为0。

所有的层级我们都使用相同的学习率，具体数值是我们在训练过程中不断调整得出的。主要调整方法是每当模型在当前的学习率下验证错误率不再降低时，我们就把学习率除以10。初始学习率是0.01，在完成训练过程中总共减少了三次。我们对120万张图像训练了约90个周期(cycle，也可称为epoch），使用两块NVIDIA GTX 580 3GB GPU，总共花费5-6天。

1. **结果**

模型对ILSVRC-2010的训练结果总结在表1。Top1和Top5错误率分别为37.5%和17%（若没有按4.1介绍的对10个预测值取平均的话，错误率是39%和18.3%）。而当时比赛最佳结果是47.1%和28.2%，是由六个稀疏编码(sparse coding)模型的预测值取平均得到，且所用的训练特征都不同[2]。比赛后至今最佳结果是45.7%和25.7%，是两个分类器的预测值的平均值，分类器的训练基于由不同类型的密集取样特征计算出的[Fisher Vector](http://link.zhihu.com/?target=https%3A//hal.inria.fr/hal-00779493/file/RR-8209.pdf)（FV）[24]。

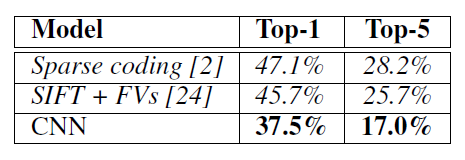


表1：ILSVRC-2010结果对比，斜体数字是其他模型取得的最佳结果。

我们还将模型应用在ILSVRC-2012比赛上，并将训练结果列于表2。由于ILSVRC-2012没有提供测试集的标签，我们无法计算测试错误率。在本段落接下来的篇幅内，验证错误率和测试错误率代表同一个东西，因为表2显示它们的差距不超过0.1%。我们的CNN的Top5错误率是18.2%。五个结构类似的CNN（5 CNNs）的预测结果取均值得到16.4%的错误率。另外图二第四个CNN模型，事先训练过整个ImageNet 2011秋季 数据集（1500万张图像共22000种类别），其内部位于最后一个汇合层之后还有第六个卷积层。该模型经过“微调”后，用于训练ILSVRC-2012数据集得到的错误率是16.6%。将事先训练2011秋季 数据集的两个CNN和前述的5 CNNs模型的预测值取平均，得到错误率为15.3%。比赛中亚军预测结果错误率是26.2%，其模型是数个分类器的预测值的平均值，分类器的训练基于由不同类型的密集取样特征计算出的Fisher Vector（FV）。

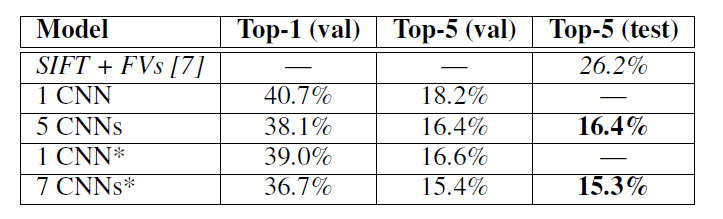


表2：ILSVRC-2012验证集和测试集错误率对比。斜体数据是其他模型获得的最佳结果。带\*的模型都是预先训练过ImageNet 2011秋季 数据集的模型。

最后，我们还对ImageNet 2009秋季 数据集进行训练，该数据集包含890万张图像共10184个标签类别。对该数据集，我们依照以往研究的规矩，将其对半分，一半是训练集，一半是测试集。因为没有给定的测试集，所以我们的分法与其他作者的分法肯定有区别，但还好这并不影响结果。我们最终得到Top1和Top5错误率分别为67.4%和40.9%，但使用的模型是前述CNN的基础上再最后一个汇合层后增加了第六卷积层。目前为止，对该数据集的最佳预测错误率是78.1%和60.9%[19]。

* 1. **量化评估**

图3展示了神经网络两个数据连接层（输入层和第一卷积层）训练的卷积核的输出结果。神经网络训练出了丰富的卷积核 - 频率选择型，方向选择型（frequency- and orientation-selective kernels）以及多种色块（colored blobs）。另外，也可以看出两块GPU的明确分工，这是通过3.5节中介绍的部分连接型实现的。显卡一上的卷积核基本上是不识别颜色的，而显卡二上的卷积核大部分是能识别颜色的。这样的分工效果在每一次训练中都会出现，而且与任一权重的随机初始化是不相关的。

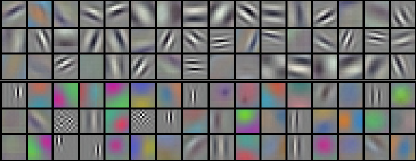


图3：第一个卷积层的96个卷积核对输入的图像训练后的输出结果。前48个和后48个运算分别在两个GPU上进行。

下图4中左半部分是我们模型对八幅测试图像计算的前五个分类，作为我们评估模型训练效果的定性分析基础。值得一提的是，部分不在图像中心的物体，比如左上角的小虫子都可以被模型准确辨识。而且大部分预测都是比较靠谱的。比如，对猎豹的前五个分类中，除了第一个正确分类外，其他分类结果都至少将其归类为猫科动物。不过有些情况下（比如汽车和樱桃），图像内容和真实标签的对应本身就有很大的歧义。

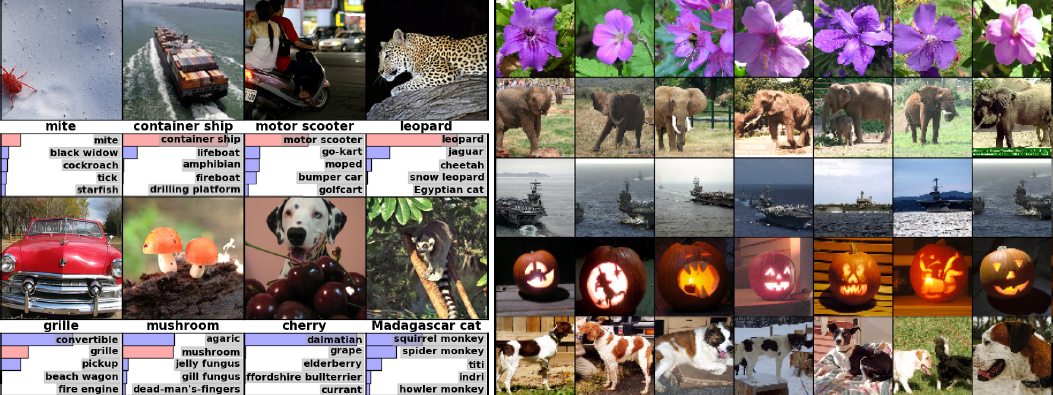


图4：（左边）八幅ILSVRC-2010测试图像以及我们的模型对其做出的前五个分类。图片的正确标签列于图片下方，而正确分类的概率也用红色条带标识（如果前五个分类结果包含正确标签）（右边）第一列是五张ILSVRC-2010测试图像。剩下的六列图像是训练集图像中通过模型最后的隐含层（也就是输出层的前一层）后得出的特征向量与第一列测试图像的相应的特征向量距离（欧式距离）最近的图像。

另一个探索神经网络视觉识别能力的方法是研究图像在最后一个层级，即维度为4096的隐含层上产生的特征激活状态（feature activation）。如果两个图像的特征激活状态向量之间的欧式距离比较小，那么就代表神经网络内部较高层次认为这两张图是类似的。图4右半部分展示了五张ILSVRC-2010测试图像以及分别计算出六张最为接近的训练集图像。可以观察到，在像素层面上，返回的几张训练图像其实与第一列的测试图像的L2距离不是特别接近。例如，第二行和第五行用大象和狗的测试图像返回的训练图像里姿势多种多样。我们在补充材料里提供了更多的测试图像与训练图像匹配结果。

计算4096维实向量之间的欧式距离是很低效的，但可以通过训练一个自动编码器将向量压缩成较短的二进制码，从而提高效率。对这些向量进行计算返回的训练集图像比直接将自动编码器运用在输入图形上返回的结果要好得多[14]，因为直接计算像素而不使用其标签会使得计算偏向于在图像边缘寻找模式的相似性，不管它们实际图片内容上是否相似。

1. **讨论**

我们的结果显示大型深度CNN可以在比较困难的数据集上单纯使用监督学习方法取得突破性进展。值得一提的是若移除我们模型中的任何一个卷积层，训练效果都会大打折扣。例如，去除任何一个中间层都会导致Top1分类结果损失增大约2%。所以我们设置的深度对于取得优良的结果是非常关键的。

为了简化我们的实验，我们没有进行非监督的预训练即使我们觉得这会有所帮助，尤其是当我们获得足够的计算能力得以增大神经网络的规模，却没有相应的更大的带标签图像集。目前为止，我们的训练结果通过不断扩大神经网络以及不断增加训练时间而有所优化。但是，如果想要接近人类视觉系统的infero-temporal路径还是有很长的路要走。我们的最终目标是用非常大且深的神经网络对视频序列进行训练，因为视频文件的瞬时结构提供了很有用的信息，而这些信息是静态图片所不具备或表达的不明显的。