“看懂”卷积神经网

Matthew D. Zeiler

摘 要

近些年，大量基于卷积神经网的模型在 ImageNet 库的分类测试中取得了不俗的成绩，但很多人还没有搞懂为什么这些卷积模型会取得如此好的效果，为什么卷积网可以提高分类效果，本文将着力解决这些疑问。本文用反卷积技术可重构每层的输入特征并加以可视化，通过可视化的展示来分析如何建立更好的网络结构，该网络结构在 ImageNet 分类测试上取得的成绩超过了 Krizhevsky 等人提出的模型。本文还详细分析了该网络结构中，每层对整体分类性能的贡献。最后本文对比了该模型在其他数据集上取得的不俗成绩：仅仅对 softmax 分类器重新训练，该模型击败了 Caltech-101 和 Caltech-256 测试集的历史最好成绩。

1. 简介

自发明以来(LeCun et al., 1989)，卷积网在手写数字识别和人脸检测方面取得了惊人成绩。在过去一年中，许多文章都表示利用卷积神经网，在各种视觉分类任务中取得了不错成绩。例如：(Ciresan et al.,2012)展示了他们的卷积网在 NORB 和 CIFAR-10 数据库上取得的不错效果；最令人印象深刻的是(Krizhevsky et al.,2012)在 ImageNet 2012 分类评测中，取得了错误率仅为 16.4%的好成绩，远远超出第二名错误率为 26.1%的成绩。

卷积网的复兴有以下原因：1. 拥有数以百万带标签训练集的出现；2. 基于 GPU 训练算法的出现，使得训练复杂卷积网模型不再是奢望；3.更好模型调优策略，如 Dropout 策略(Hinton et al., 2012)。

尽管卷积网取得了如此辉煌的成绩，但许多人对卷积网深层次的认识仍十分有限。从科研角度来讲，靠运气来构造和调节卷积网参数，这种方法是非常不可取的。本文利用反卷积网重构每层的输入信息，再将重构信息投影到像素空间中，从而实现了可视化。通过可视化技术来分析“输入的色彩如何映射成不同层上的特征”，“特征如何随着训练过程而发生变化”等问题，甚至利用可视化技术来诊断和改进当前网络结果可能存在的问题。本文还进行了模型敏感性测试，通过遮挡输入图片的局部来分析场景中哪部分信息对分类效果影响最大。反卷积网络(deconvnet)在(Zeiler et al., 2011)中有详细论述。

本文主要参考了(Krizhevsky et al.,2012)的模型，通过改变该模型的一些参数（核大小、跨度等），选出在 ImageNet 上分类效果最好的结构作为最终模型，然后仅仅重新训练模型末端的 softmax 分类器，评估该模型在其他数据集上的分类效果。

1.1 相关工作

通过肉眼观察来分析网络模型的相关特征是常用的分析方法，缺点是该方法只能用于第 1 层网络的分析，因为输入信息是人们可理解的图像，更高层就不能直接用肉眼观察了。现有可用来分析高层特性的方法也十分有限（Erhan et al., 2009）。神经网络常用的训练方法是梯度下降法，该方法不但易受初始值影响，也无法反映神经元的不变性能力。针对第二个缺点，(Leet al., 2010)(最早的 idea 来自(Berkes & Wiskott,2006))通过分析神经元的赫森矩阵与最优响应间的数学关系，揭示了不变性的一些特点，该方法的缺点是：对于高层神经元，不变性特征已经变得非常复杂，难以用一个二次函数拟合。相比而言，我们提供了一个非参数化观察网络不变性的方法，展示了训练集中的哪些图案能够刺激网络形成特征。(Donahue et al., 2013) 展示了有关联的图形能够强烈刺激高层网络产生特征，与他们有所不同，本文不仅仅分析输入图片，而且通过由高到低，由输出到输入的重构，逐步分析每个输入信号中的特殊图形和某个特定特征之间的映射关系。

2. 实现方法

本文采用了由(LeCun etal.,1989)以及(Krizhevsky etal., 2012)提出的标准的有监督学习的卷积网模型，该模型通过一系列隐含层，将输入的二维彩色图像映射成长度为 C 的一维概率向量，向量中的每个概率分别对应 C 个不同分类。每层包含以下部分：1. 卷积层，每个卷积图都由前面一层网络的输出结果（对于第一层来说，上层输出结果就是输入图片），与学习获得的特定核进行卷积运算产生。2.矫正层，对每个卷积结果都进行矫正运算 $relu(x)=max(x,0)$；3.[可选] max pooling 层，对矫正运算结果进行一定邻域内的 max pooling 操作，获得降采样图；4.[可选] 对降采样图进行对比度归一化操作，使得输出特征平稳。更多操作细节，请参考（Krizhevsky et al., 2012）以及 （Jarrett et al.,2009）。最后几层是全连接网络，输出层是一个 softmax 分类器。图 3 上部展示了这个模型。

我们使用 N 张标签图片{x, y}构成的数据集来训练模型，其中标签y_i是一个离散变量，用来表示图片的类别。用交叉熵误差函数来评估输出标签$\hat{y}_i$和真实标签y_i的差异。整个网络参数（包括卷积层的卷积核，全连接层的权值矩阵和偏置值）通过反向传播算法进行训练，选择随机梯度下降法更新权值。具体细节参见章节 3.

2.1 通过反卷积网（Deconvnet）实现可视化

要想深入了解卷积网，就需要了解中间层特征的作用。本文将中间层特征反向映射到像素空间，观察出什么输入会导致特定的输出，可视化过程基于（Zeiler et al., 2011）提出的反卷积网络实现。一层反卷积网可以看成是一层卷积网的逆过程，它们拥有相同的卷积核和 pooling 函数（准确来讲，应该是逆函数），因此反卷积网是将输出特征逆映射成输入信号。在（Zeiler et al., 2011）中，反卷积网络被用作无监督学习，本文则用来进行可视化演示。

在本文的模型中，卷积网的每一层都附加了一个反卷积层，参见图 1，提供了一条由输出特征到输入图像的反通路。首先，输入图像通过卷积网模型，每层都会产生出特定特征；而后，我们将反卷积网中观测层的其他连接权值全部置零，将卷积网观测层产生的特征当作输入，送给对应的反卷积层，依次进行以下操作：1. unpool；2. 矫正；3. 反卷积。

unpooling：严格来讲，max pooling 操作是不可逆的，本文用了一种近似方法来计算 max pooling 的逆过程：在 max pooling 过程中，用 Max Locations “Switches”表格记录下每个最大值的位置，在 unpooling 过程中　我们将最大值标注回记录所在位置，其余位置填 0。图 1 底部显示了这一过程。

矫正：在卷积网中，为保证特征有效性，我们通过 relu 非线性函数来保证所有输出都为非负数，这个约束对反卷积过程依然成立，因此将重构信号送入 relu 函数中。

反卷积：卷积网使用学习得到的卷积核与上层输出做卷积，得到特征。为了实现逆过程，反卷积网使用相同卷积核的转置作为核，与矫正后的特征进行卷积运算。

在 unpooling 过程中，由于"Switches"只记录了极大值的位置信息，其余位置均用 0 填充，因此重构出的图片看起来会不连续，很像原始图片中的某个碎片，这些碎片就是训练出高性能卷积网的关键。由于这些重构图像不是从模型中采样生成，因此中间不存在生成式过程。

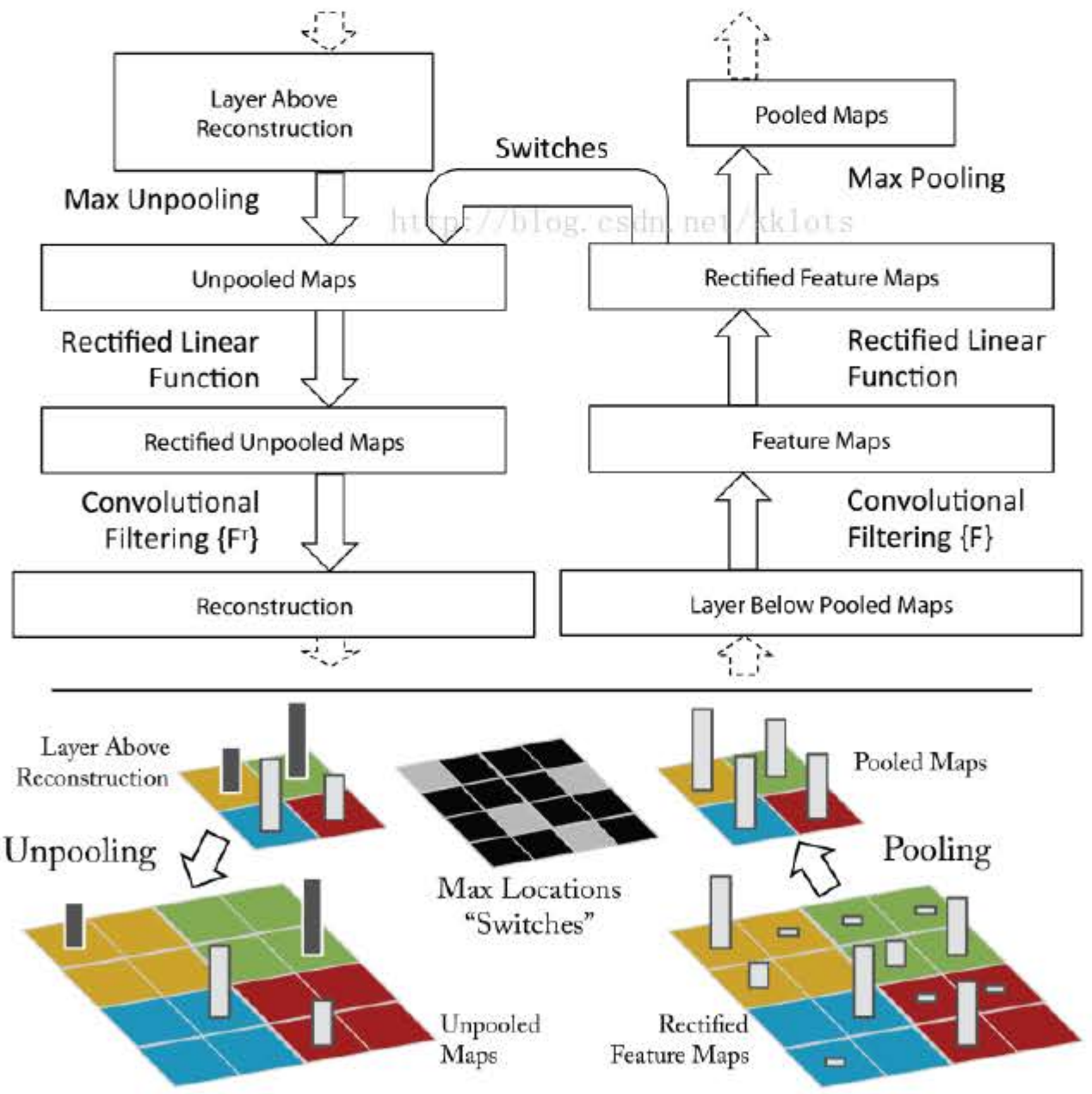


图 1. Top：一层反卷积网（左）附加在一层卷积网（右）上。反卷积网层会近似重构出下卷积网层产生出的特征。Bottom：反卷积网 unpooling 过程的演示，使用 switches 表格记录极大值点的位置，从而近似还原出 pooling 操作前的特征。

3. 训练细节

图 3 中的网络模型与(Krizhevsky et al., 2012)使用的卷积模型很相似，不同点在于：1. Krizhevsky 在 3，4，5 层使用的是稀疏连接（由于该模型被分配到了两个 GPU 上），而本文用了稠密连接。2. 另一个重要的不同将在章节 4.1 和图 6 中详细阐述。

本文选择了 ImageNet 2012 作为训练集（130 万张图片，超过 1000 个不同类别），首先截取每张 RGB 图片最中心的 256×256 区域，然后减去整张图片颜色均值，再截出 10 个不同的 224×224 窗口（可对原图进行水平翻转，窗口可在区域中滑动）。采用随机梯度下降法学习，batchsize 选择 128，学习率选择 0.01，动量系数选择 0.9；当误差趋于收敛时，手动停止训练过程；Dropout 策略(Hinton et al.,2012)运用在全连接层中，系数设为 0.5，所有权值初始值设为 0.01，偏置值设为 0.

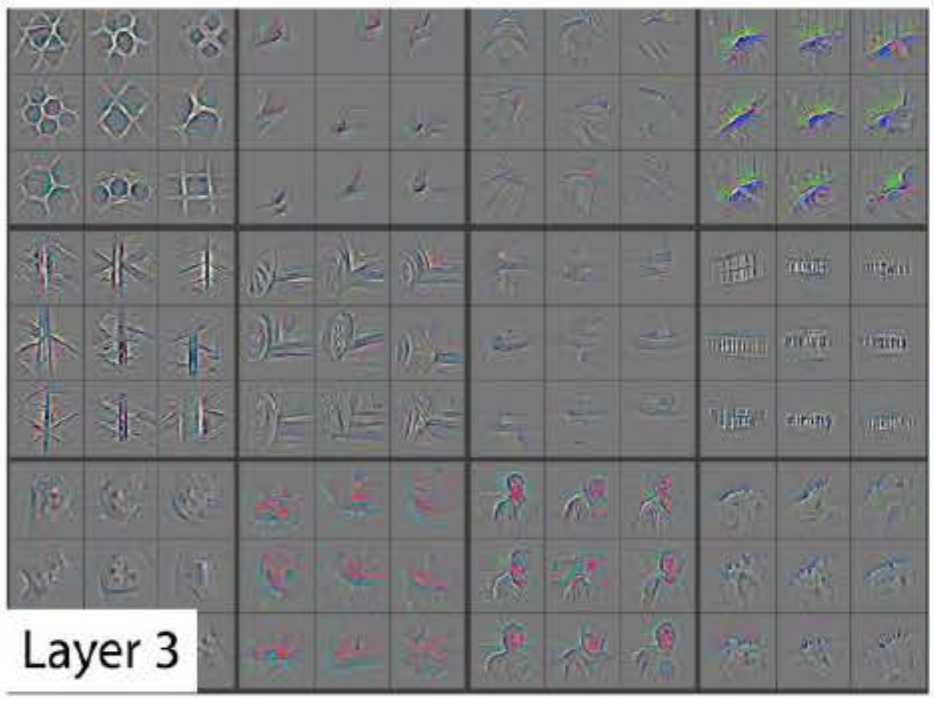
图 6(a)展示了部分训练得到的第 1 层卷积核，其中有一部分核数值过大，为了避免这种情况，我们采取了如下策略：均方根超过 0.1 的核将重新进行归一化，使其均方根为 0.1。该步骤非常关键，因为第 1 层的输入变化范围在[-128,128]之间。前面提到了，我们通过滑动窗口截取和对原始图像的水平翻转来提高训练集的大小，这一点和(Krizhevsky et al.,2012)相同。整个训练过程基于(Krizhevsky et al.,2012)的代码实现，在单块 GTX580 GPU 上进行，总共进行了 70 次全库迭代，运行了 12 天。

4. 卷积网可视化

通过章节 3 描述的结构框架，我们开始使用反卷积网来展示反向生成的刺激。

特征可视化：图 2 展示了训练结束后，模型各个隐含层提取的特征，图 2 显示了在给定输出特征的情况下，反卷积层产生的最强的 9 个输入特征。将这些计算所得的特征，用像素空间表示后，可以清晰地看出：一组特定的输入特征（通过重构获得），将刺激卷积网产生一个固定的输出特征。这一点解释了为什么当输入存在一定畸变时，网络的输出结果保持不变。在可视化结果的右边是对应的输入图片，和重构特征相比，输入图片之间的差异性很大，而重构特征只包含那些具有判别能力纹理结构。举例说明：层 5 第 1 行第 2 列的 9 张输入图片各不相同，差异很大，而对应的重构输入特征则都显示了背景中的草地，没有显示五花八门的前景。

每层的可视化结果都展示了网络的层次化特点。层 2 展示了物体的边缘和轮廓，以及与颜色的组合；层 3 拥有了更复杂的不变性，主要展示了相似的纹理（例如：第 1 行第 1 列的网格模型；第 2 行第 4 列的花纹）；层 4 不同组重构特征存在着重大差异性，开始体现类与类之间的差异：狗狗的脸（第 1 行第 1 列），鸟的腿（第 4 行第 2 列）。层 5 每组图片都展示了存在重大差异的一类物体，例如：键盘（第 1 行第 1 列），狗（第 4 行）。



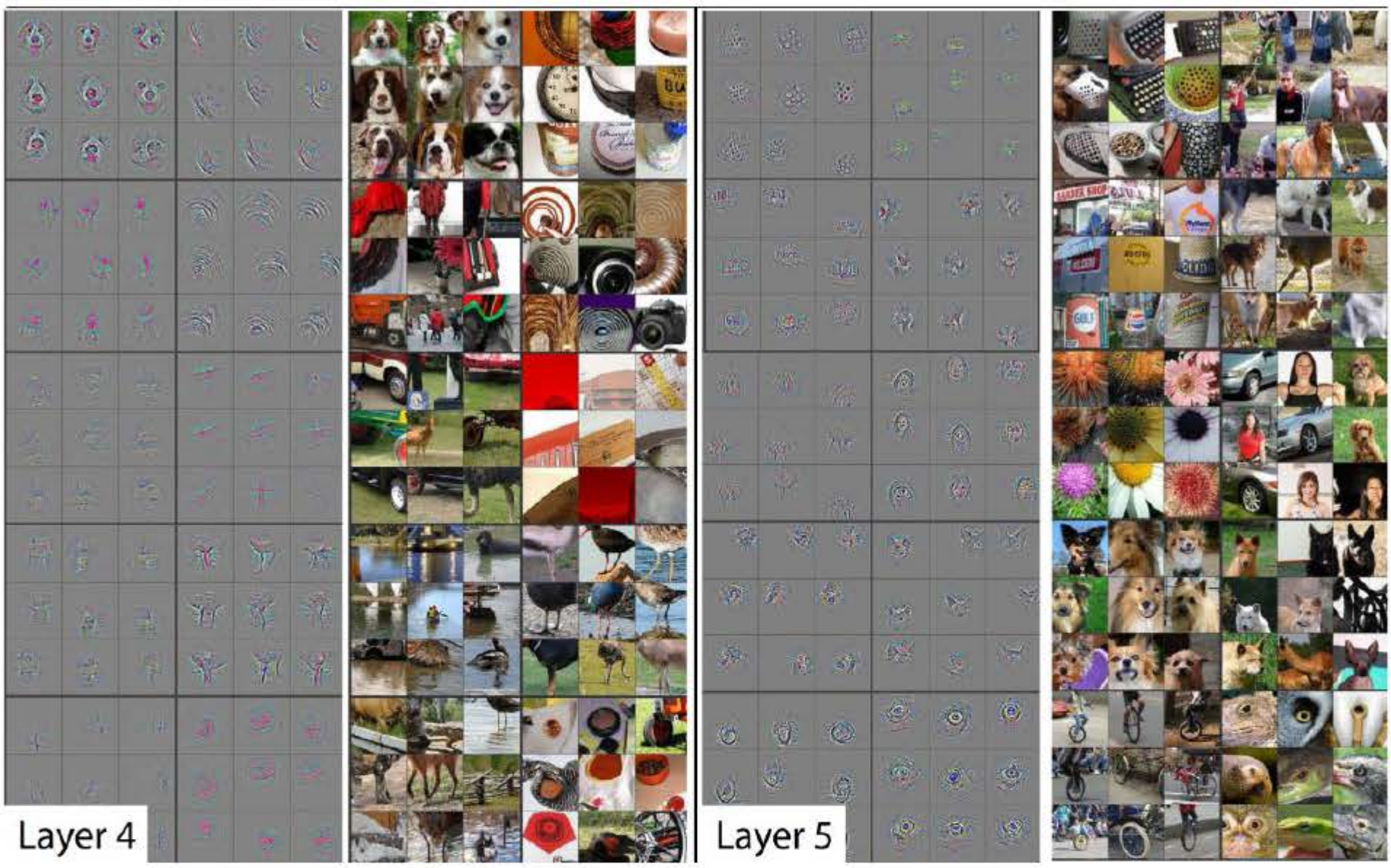


图 2.训练好的卷积网，显示了层 2 到层 5 通过反卷积层计算，得到的 9 个最强输入特征，并将输入特征映射到了像素空间。本文的重构输入特征不是采样生成的：它们是固定的，由特定的输出特征反卷积计算产生。每一个重构输入特征都对应地显示了它的输入图像。三点启示：1.每组重构特征（9 个）都有强关联性；2.层次越高，不变性越强；3.都是原始输入图片具有强辨识度部分的夸张展现。例如：狗的眼睛、鼻子（层 4，第 1 行，第 1 列）。

特征在训练过程中的演化：图 4 展示了在训练过程中，由特定输出特征反向卷积，所获得的最强重构输入特征（从所有训练样本中选出）是如何演化的，当输入图片中的最强刺激源发生变化时，对应的输出特征轮廓发生跳变。经过一定次数的迭代后，底层特征趋于稳定，但更高层的特征则需要更多的迭代才能收敛（约 40～50 个周期），这表明：只有所有层都收敛时，分类模型才堪用。

特征不变性：图 5 展示了 5 个不同的例子，它们分别被平移、旋转和缩放。图 5 右边显示了不同层特征向量所具有的不变性能力。在第 1 层，很小的微变都会导致输出特征变化明显，但越往高层走，平移和尺度变化对最终结果的影响越小。总体来讲：卷积网无法对旋转操作产生不变性，除非物体具有很强的对称性。

4.1 结构选取

观察 Krizhevsky 的网络模型可以帮助我们在一开始就选择一个好的模型。反卷积网可视化技术显示了 Krizhevsky 卷积网的一些问题。如图 6(a)以及 6(d)所示，第 1 层卷积核混杂了大量的高频和低频信息，缺少中频信息；第 2 层由于卷积过程选择 4 作为跨度，产生了混乱无用的特征。为了解决这些问题，我们做了以下工作：(i)将第 1 层的卷积核大小由 11×11 调整为 7×7；(ii)将卷积跨度由 4 调整为 2；新的模型不但保留了 1、2 层绝大部分的有用特征，如图 6(c),6(e)所示，还提高了最终分类性能，我们将在章节 5.1 中看到具体结果。

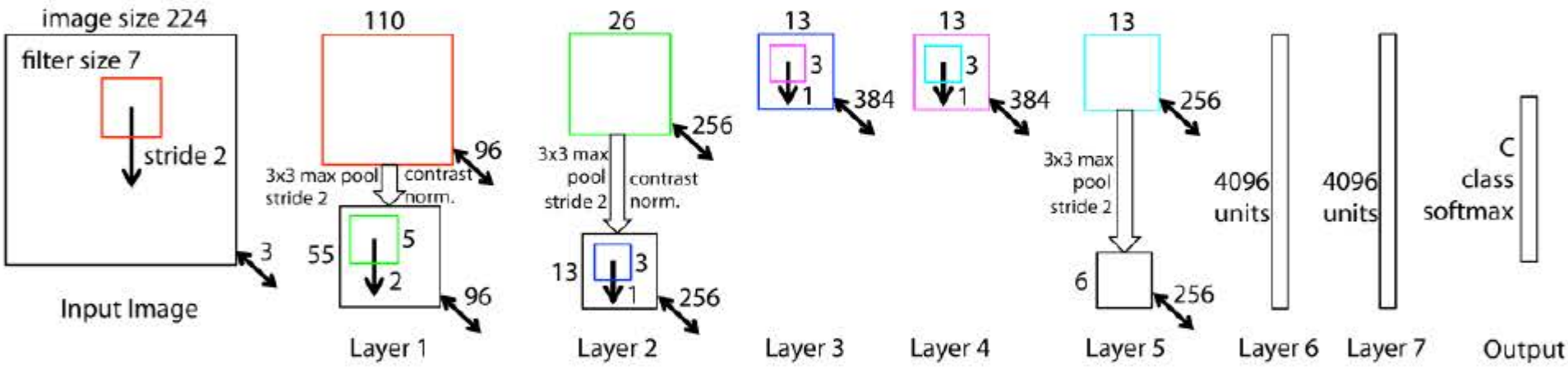


图 3.本文使用 8 层卷积网模型。输入层为 224×224 的 3 通道 RGB 图像，从原始图像裁剪产生。层 1 包含了 96 个卷积核（红色表示），每个核大小为 7×7，x 和 y 方向的跨度均为 2。获得的卷积图进行如下操作：1.通过矫正函数 relu(x)=max(x,0)，使所有卷积值均不小于 0(图中未显示)； 2.进行 max pooling 操作(3×3 区域，跨度为 2)；3.对比度归一化操作。最终产生 96 个不同的特征模板，大小为 55×55。层 2、3、4、5 都是类似操作，层 5 输出 256 个 6×6 的特征图。最后两层网络为全连接，最终层是一个 C 类 softmax 函数，C 为类别个数。所有的卷积核与特征图均为正方形。

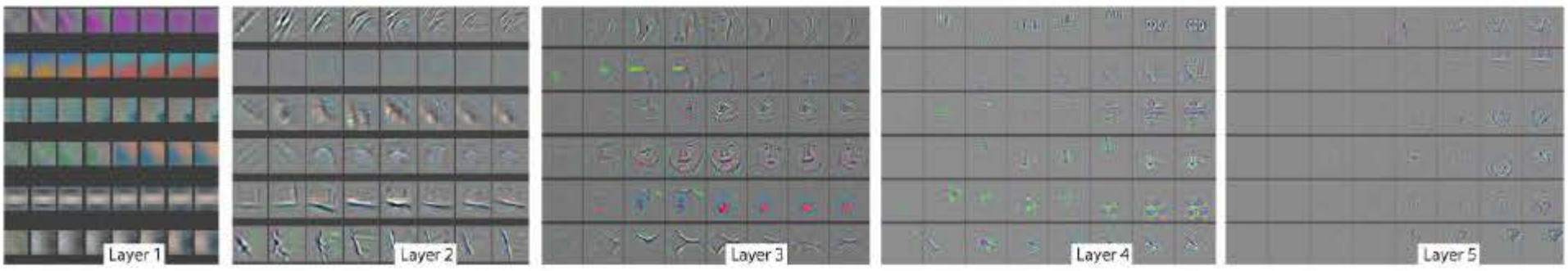


图 4.模型特征逐层演化过程。从左至右的块，依次为层 1 到层 5 的重构特征。块展示在随机选定一个具体输出特征时，计算所得的重构输入特征在第 1，2，5，10，20，30，40，64 次迭代时（训练集所有图片跑 1 遍为 1 次迭代），是什么样子（1 列为 1 组）。显示效果经过了人工色彩增强。

4.2 遮挡敏感性

当模型达到期望的分类性能时，一个自然而然的想法是：分类器究竟使用了什么信息实现分类？是图像中具体位置的像素值，还是图像中的上下文。我们试图回答这个问题，图 7 中使用了一个灰色矩形对输入图像的每个部分进行遮挡，并测试在不同遮挡情况下，分类器的输出结果，可以清楚地看到：当关键区域发生遮挡时，分类器性能急剧下降。图 7 还展示了最上层卷积网的最强响应特征，展示了遮挡位置和响应强度之间的关系：当遮挡发生在关键物体出现的位置时，响应强度急剧下降。该图真实地反映了输入什么样的刺激，会促使系统产生某个特定的输出特征，用这种方法可以一一查找出图 2 和图 4 中特定特征的最佳刺激是什么。

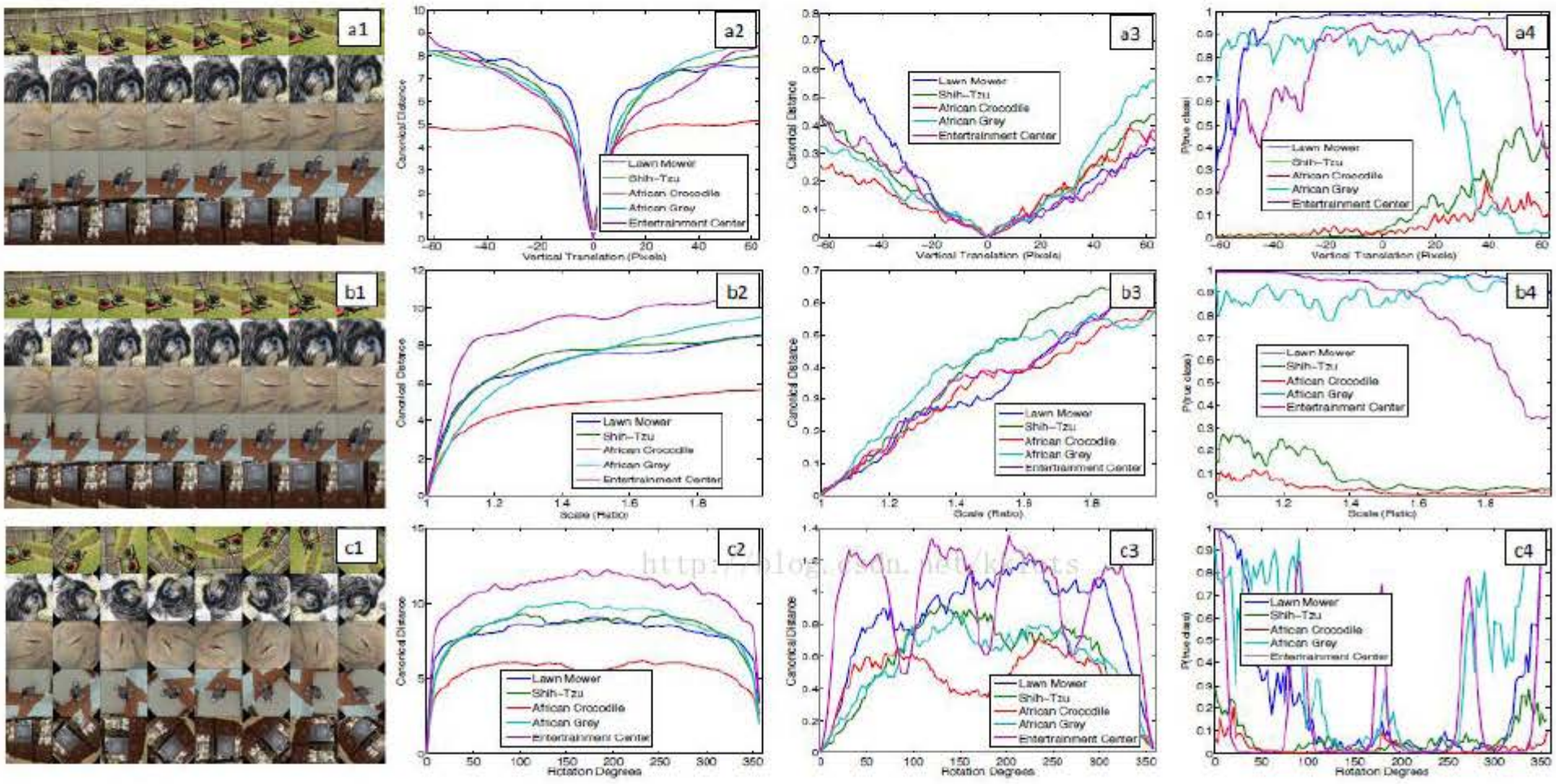


图 5.图像的垂直移动(a)、旋转(b)、尺度变化(c)以及卷积网模型中相应的特征不变性。列 1：对图像进行各种变形；列 2 和列 3：原始图片和变形图片分别在层 1～层 7 所产生特征间的欧式距离。列 4：真实类别在输出中的概率。

4.3 图片相关性分析

与其他许多已知的识别模型不同，深度神经网络没有一套有效理论来分析特定物体部件之间的关系（例如：如何解释人脸眼睛和鼻子在空间位置上的关系），但深度网络很可能非显式地计算了这些特征。为了验证这些假设，本文随机选择了 5 张狗狗的正面图片，并系统性地挡住狗狗所有照片的一部分（例如：所有的左眼，参见图 8）。对于每张图 I, 计算$\epsilon_i^l = x_i^l - \hat{x}_i^l$, 其中 x_i^l和$\hat{x}_i^l$分别表示原始图片和被遮挡图片所产生的特征，然后测量所有图片对(i, j)的误差向量ϵ的一致性：$\Delta_l = \sum_{i.i=1.i\neq j}^{5} \mathcal{H}(sign(\epsilon_i^l), sign(\epsilon_j^l))$, 其中，$\mathcal{H}$是 Hamming distance，$\Delta_l$值越小，对应操作对狗狗分类的影响越一致，就表明这些不同图片上被遮挡的部件越存在紧密联系。表 1 中我们对比了遮挡左眼、右眼、鼻子以及随机遮挡 4 种情况，选择了第 5 层和第 7 层的重构特征，可以看出遮挡左眼、右眼和鼻子的Δ比随机遮挡的Δ更低，说明眼睛图片和鼻子图片内部存在相关性。第 5 层鼻子和眼睛的得分差异明显 ，说明第 5 层卷积网对部件级（鼻子、眼睛等等）的相关性更为关注；第 7 层各个部分得分差异不大，说明第 7 层卷积网开始关注更高层的信息（狗狗的品种等等）。

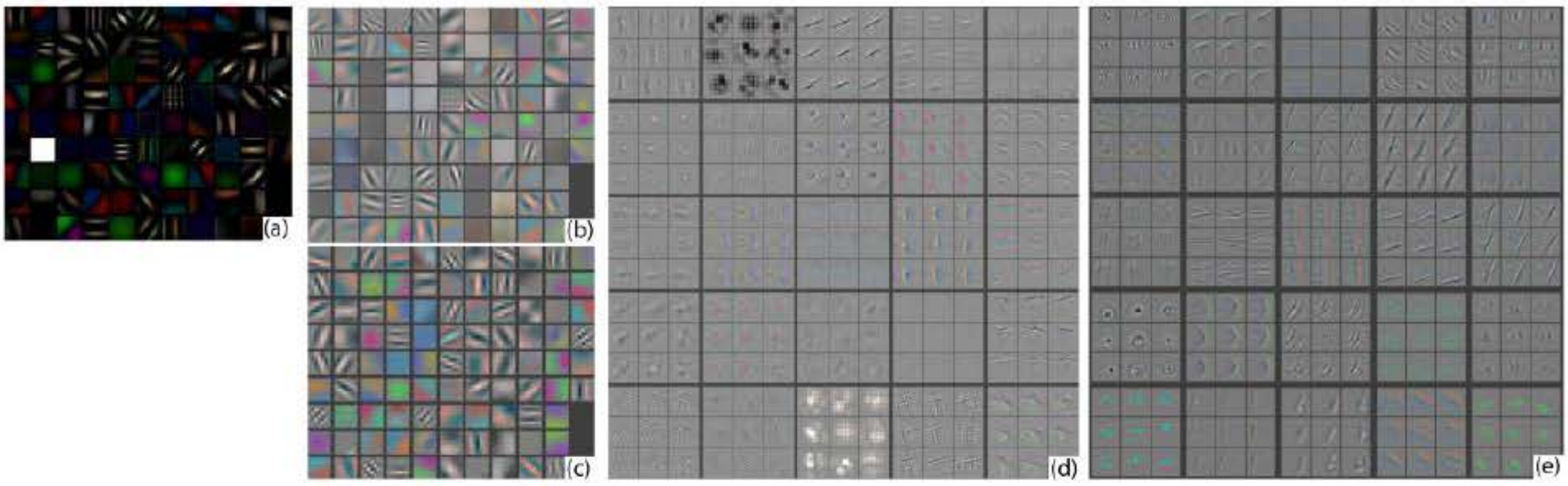


图 6.(a):层 1 输出的特征，还未经过尺度约束操作，可以看到有一个特征十分巨大；(b): (Krizhevsky et al., 2012)第 1 层产生的特征；(c):本文模型第 1 层产生的特征。更小的跨度(2 vs 4)，更小的核尺寸(7×7 vs 11×11)从而产生了更具辨识度的特征和更少的“无用特征”；(d): (Krizhevsky et al., 2012)第 2 层产生的特征；(e):本文模型第 2 层产生的特征。很明显，没有(d)中的模糊特征。

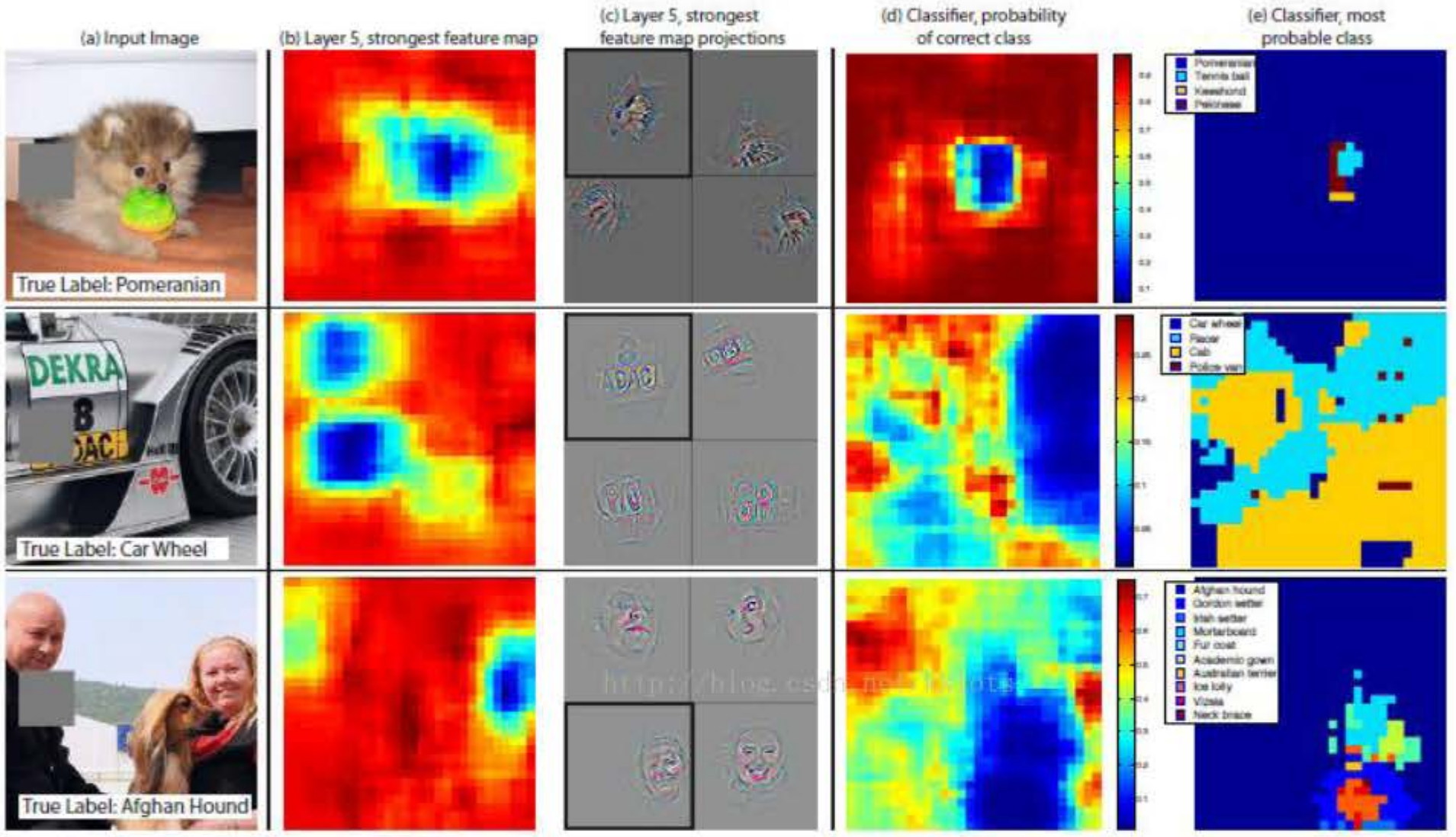


图 7.输入图片被遮挡时的情况。灰色方块遮挡了不同区域(第 1 列)，会对第 5 层的输出强度产生影响 (b 和 c)，分类结果也发生改变(d 和 e)。(b)：图像遮挡位置对第 5 层特定输出强度的影响。(c):将第 5 层特定输出特征投影到像素空间的情形（带黑框的），第 1 行展示了狗狗图片产生的最强特征。当存在遮挡时，对应输入图片对特征产生的刺激强度降低（蓝色区域表示降低）。(d):正确分类对应的概率，是关于遮挡位置的函数，当小狗面部发生遮挡时，波西米亚小狗的概率急剧降低。(e):最可能类的分布图，也是一个关于遮挡位置的函数。在第 1 行中，只要遮挡区域不在狗狗面部，输出结果都是波西米亚小狗，当遮挡区域发生在狗狗面部，但又没有遮挡网球时，输出结果是“网球”。在第 2 行中，车上的纹理是第 5 层卷积网的最强输出特征，但也很容易被误判为“车轮”。第 3 行包含了多个物体，第 5 层卷积网对应的最强输出特征是人脸，但分类器对“狗狗”十分敏感（(d)中的蓝色区域），原因在于 softmax 分类器使用了多组特征（既有人的特征，又有狗的特征）。

图 8.其他用于遮挡实验的图片。第 1 列：原始图片；第 2，3，4 列：遮挡分别发生在右眼、左眼和鼻子部位；其余列显示了随机遮挡。

Occlusion Location	Mean Feature Sign Change Layer 5	Mean Feature Sign Change Layer 7
Right Eye	0.067 ± 0.007	0.069 ± 0.015
Left Eye	0.069 ± 0.007	0.068 ± 0.013
Nose	0.079 ± 0.017	0.069 ± 0.011
Random	0.107 ± 0.017	0.073 ± 0.014

表 1.测试当狗狗不同部位的相关性。在第 5 层中，眼睛和鼻子的得分更低，表明网络已经开始产生相关性；7 层得分差异不大，说明高层网络开始关注分类特征（狗狗的品种），而不是局部特征。

5. 实验内容

5.1 ImageNet 2012

该图像库共包含了（130 万/5 万/10 万）张（训练/确认/测试）样例，种类数超过 1000。表 2 显示了本文模型的测试结果。

首先，本文重构了（Krizhevsky et al.，2012）的模型，重构模型的错误率与作者给出的错误率十分一致，误差在 0.1%以内，以此作为参考标准。

而后，本文将第 1 层的卷积核大小调整为 7×7，将第 1 层和第 2 层卷积运算的跨度改为 2，获得了相当不错的结果，与（Krizhevsky et al.，2012）相比，我们的错误率为 14.8%，比（Krizhevsky et al.，2012）的 15.3%提高了 0.5 个百分点。

Error %	Val Top-1	Val Top-5	Test Top-5
(Gunji et al., 2012)	-	-	26.2
(Krizhevsky et al., 2012), 1 convnet	40.7	18.2	--
(Krizhevsky et al., 2012), 5 convnets	38.1	16.4	16.4
(Krizhevsky et al., 2012)*, 1 convnets	39.0	16.6	--
(Krizhevsky et al., 2012)*, 7 convnets	36.7	15.4	15.3
Our replication of (Krizhevsky et al., 2012), 1 convnet	40.5	18.1	--
1 convnet as per Fig. 3	38.4	16.5	--
5 convnets as per Fig. 3 – (a)	36.7	15.3	15.3
1 convnet as per Fig. 3 but with layers 3,4,5: 512,1024,512 maps – (b)	37.5	16.0	16.1
6 convnets, (a) & (b) combined	**36.0**	**14.7**	**14.8**

表 2.ImageNet2012 分类错误率。星号*表示使用了 ImageNet2011 和 ImageNet2012 两个训练集

改变卷积网结构：如表 3 所示，本文测试了改变（Krizhevsky et al.，2012）模型的结构会对最终分类造成什么样的影响，例如：调节隐藏层节点个数，或者将某隐含层直接删除等等。每种情况下，都将改变后的结构从头训练。当层 6、7 被完全删除后，错误率只有轻微上升；删除掉两个隐含卷积层，错误率也只有轻微上升。然而当所有的中间卷积层都被删除后，仅仅只有 4 层的模型分类能力急剧下降。这个现象或许说明了模型的深度与分类效果密切相关，深度越大，效果越好。改变全连接层的节点个数对分类性能影响不大；扩大中间卷积层的节点数对训练效果有提高，但也同时加大了全连接出现过拟合的可能。

Error %	Train Top-1	Val Top-1	Val Top-5
Our replication of (Krizhevsky et al., 2012), 1 convnet	35.1	40.5	18.1
Removed layers 3,4	41.8	45.4	22.1
Removed layer 7	27.4	40.0	18.4
Removed layers 6,7	27.4	44.8	22.4
Removed layer 3,4,6,7	71.1	71.3	50.1
Adjust layers 6,7: 2048 units	40.3	41.7	18.8
Adjust layers 6,7: 8192 units	26.8	40.0	18.1
Our Model (as per Fig. 3)	33.1	38.4	16.5
Adjust layers 6,7: 2048 units	38.2	40.2	17.6
Adjust layers 6,7: 8192 units	22.0	38.8	17.0
Adjust layers 3,4,5: 512,1024,512 maps	18.8	**37.5**	**16.0**
Adjust layers 6,7: 8192 units and Layers 3,4,5: 512,1024,512 maps	**10.0**	38.3	16.9

表 3.不同结构在 ImageNet2012 上的分类错误率，上表为(Krizhevsky et al., 2012)模型，下表为本文的模型

5.2 特征泛化能力

为了测试模型泛化能力，本文又测试了 Caltch-101，Caltech-256 和 PASCAL VOC 2012 共 3 个库。具体方法：不改变模型 1～7 层训练结果，仅仅对最高层的 softmax 分类器重新训练。

由于本文模型中的分类器（softmax）与其他方法（例如：SVM）在复杂度上很相似，因此也对比了本文学习到的特征是否可用到其他分类器上。

Caltech-101库:测试方法遵循(Fei-fei et al., 2006)提出的方法。可以看到，本文学习的模型以 2.2%的优势击败历史最好成绩(Bo et al., 2013)。在另一个实验中：我们基于Caltech-101库重新训练卷

积网模型，获得的分类效果十分惨淡，正确率仅有46.5%，说明基于ImageNet学到的特征更有效。

# Train	Acc % 15/class	Acc % 30/class
(Bo et al., 2013)	–	81.4 ± 0.33
(Jianchao et al., 2009)	73.2	84.3
Non-pretrained convnet	22.8 ± 1.5	46.5 ± 1.7
ImageNet-pretrained convnet	$\mathbf{83.8 \pm 0.5}$	$\mathbf{86.5 \pm 0.5}$

表 4. Caltech-101 历史最好 2 个成绩与本文模型成绩的对比

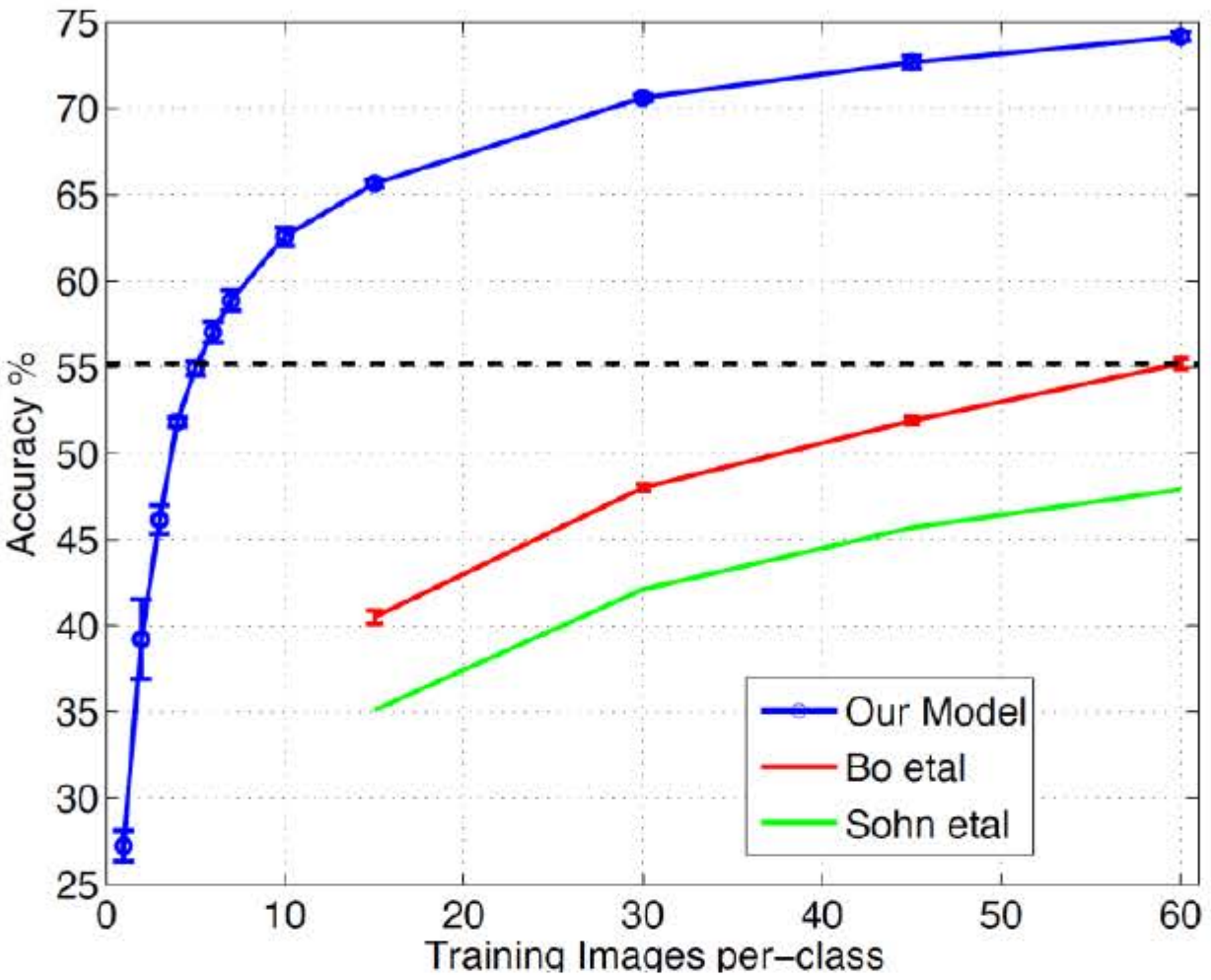


图 9.每类训练样本数量和识别率的曲线图

Caltech-256库:遵循(Griffinet al., 2006)的测试方法进行测试，结果如表5所示。基于ImageNet预先学习的模型以准确率高出19%的巨大优势，击败了历史最好成绩。图9从另一个角度描述了基于ImageNet预先学习模型的成功。值得注意的是：当基于Caltech-256库重新训练整个模型后，分类精度仅有38.8%。

# Train	Acc % 15/class	Acc % 30/class	Acc % 45/class	Acc % 60/class
(Sohn et al., 2011)	35.1	42.1	45.7	47.9
(Bo et al., 2013)	40.5 ± 0.4	48.0 ± 0.2	51.9 ± 0.2	55.2 ± 0.3
Non-pretr.	9.0 ± 1.4	22.5 ± 0.7	31.2 ± 0.5	38.8 ± 1.4
ImageNet-pretr.	$\mathbf{65.7 \pm 0.2}$	$\mathbf{70.6 \pm 0.2}$	$\mathbf{72.7 \pm 0.4}$	$\mathbf{74.2 \pm 0.3}$

表 5. Caltech-256 历史最好 2 个成绩与本文模型成绩的对比

PASCAL 2012库:本文使用标准的训练方法来训练softmax分类器。由于PASCAL库中的测试图片有可能一张包含多个物体，而我们的模型一张图片只给出一个预测，因此没能超越历史最好记录(Yan et al.,2012)，大约落后了3.2%，不过本文模型仍然在5个类别上超过了他们，有些类别优势还很明显。

Acc %	[A]	[B]	Ours	Acc %	[A]	[B]	Ours
Airplane	92.0	**97.3**	96.0	Dining tab	63.2	**77.8**	67.7
Bicycle	74.2	**84.2**	77.1	Dog	68.9	83.0	**87.8**
Bird	73.0	80.8	**88.4**	Horse	78.2	**87.5**	86.0
Boat	77.5	85.3	**85.5**	Motorbike	81.0	**90.1**	85.1
Bottle	54.3	**60.8**	55.8	Person	91.6	**95.0**	90.9
Bus	85.2	**89.9**	85.8	Potted pl	55.9	**57.8**	52.2
Car	81.9	**86.8**	78.6	Sheep	69.4	79.2	**83.6**
Cat	76.4	89.3	**91.2**	Sofa	65.4	**73.4**	61.1
Chair	65.2	**75.4**	65.0	Train	86.7	**94.5**	91.8
Cow	63.2	**77.8**	74.4	Tv	77.4	**80.7**	76.1
Mean	74.3	**82.2**	79.0	# won	0	**15**	5

表6.PASCAL 2012分类结果，历史最好2个成绩与我们的分类器成绩的对比([A]=(Sande et al., 2012) , [B] = (Yan et al.,2012)).

5.3 特征分析

本文通过如下方法来测试模型在 ImageNet 训练库中所学习到的特征：保留训练后模型的前 n 层，后端连接线性 SVM 或 softmax 分类器。表 7 显示了基于 Caltech101 与 Caltech256 两个库的分类结果。可以看出：模型学习到的特征同样适用于 SVM 进行分类。另外，随着保留层的增多，分类能力稳步上升，当保留全部层时，分类结果达到最好。该结果证明了：当深度增加时，网络可学到更好的特征。

	Cal-101 (30/class)	Cal-256 (60/class)
SVM (1)	44.8 ± 0.7	24.6 ± 0.4
SVM (2)	66.2 ± 0.5	39.6 ± 0.3
SVM (3)	72.3 ± 0.4	46.0 ± 0.3
SVM (4)	76.6 ± 0.4	51.3 ± 0.1
SVM (5)	$\mathbf{86.2 \pm 0.8}$	65.6 ± 0.3
SVM (7)	$\mathbf{85.5 \pm 0.4}$	$\mathbf{71.7 \pm 0.2}$
Softmax (5)	82.9 ± 0.4	65.7 ± 0.5
Softmax (7)	$\mathbf{85.4 \pm 0.4}$	$\mathbf{72.6 \pm 0.1}$

表7.SVM与Softmax分别连接不同层的分类能力

6. 讨论（略）

参考文献（略）