解读：Very Deep Convolutional Networks for Large-scale Image Recognition

王晓捷 (11521053)

May 23, 2016

1 引言

卷积神经网络（Convolutional Neural Network，简称CNN）是近年发展起来，并引起广泛重视的一种高效识别方法。现在，CNN已经成为众多科学领域的研究热点之一，特别是在计算机视觉方面的模式分类领域。由于该网络避免了对图像的复杂的前期处理，可以直接输入原始图像，因而得到了更为广泛的应用。CNN的广泛应用也得益于当今大规模的支持识别工作的图片数据库，如ImageNet，以及高性能的计算系统，例如GPU和各种大规模的分布式簇。

2012年，Krizhevsky 等人[1]在大规模图像识别中使用CNN 取得了不错的效果。他们训练了一个具有一定深度的CNN去分类高像素的图片。这个大型的网络包含有5 个卷积层，3个全连接层和最终的softmax输出层，包含6千万个参数和65万个神经元。在ILSCRV-2010 中，top-1 和top-5 的错误率分别为37.5% 和17.0%，在ILSCRV-2012 中top-5 的错误率仅为15.3%。这篇文章中涉及到的网络结构可以说是开启在ImageNet 数据集上进行更大、更深的CNN 的开山之作。在此之后，有很多研究人员从不同的角度尝试对[1]中描述的网络结构（简称为Alex-net）进行一定的调整从而提高CNN 的准确率。对Alex-net 的网络结构的调整目前主要包括两个方面，首先是在第一个卷积层中使用更小的感受野和步长，其次是在整个图片和不同尺度的图片上进行训练和测试[2]。

牛津大学的Karen Simonyan和Andrew Zisserman在于2014年发表的论文[3]中则主要探讨了深度对于卷积神经网络的重要性。本文也主要针对论文[3]中的内容进行一定的介绍说明。接下来本文的主要内容组织如下：第2部分简要介绍[1]中的Alex-net的结构及相关方法；第3部分介绍[3]中的网络结构及每层网络的配置；第4部分介绍详细说明了论文中用来图片识别的训练和测试策略，以及一些实现的细节；第5部分列出了论文中的实验结果；最后，在第6部分对本文进行一定的总结。

2 Alex-net概述

[1]中使用的数据集ILSVRC是ImageNet的一个子集，一共1000个类别，每个类别包含大约1000张图片。其中训练集有120 万张，验证集有5 万张，测试集有15万张。ImageNet中包含各种不同分辨率的图片，然而系统需要的图片要有着相同的分辨率。因此，对图片进行降采样使其分辨率为256×256。若原始图片为长方形，将其按比例调整至最小边大小为256的尺寸，然后围绕图片中心，裁剪出256*256分辨率的图片。对图片的预处理仅是对训练集中图片的每个像素减去平均的RGB值。

网络的总体结构如下图1所示。网络中第一个卷积层的输入数据是224×224 大小的RGB图片，维度为$224 \times 224 \times 3$，利用96种核，感受野为11×11，步长为4。第二层是256 种核，感受野为5×5。第三层为384种核，感受野为3×3。第四层为384 种核，感受野为3×3。第五层为256 种核，感受野为3×3。作者们经过实验得出结论移除任何一个卷积层都会使得整个系统的准确率下降，从而说明模型的深度的重要性。

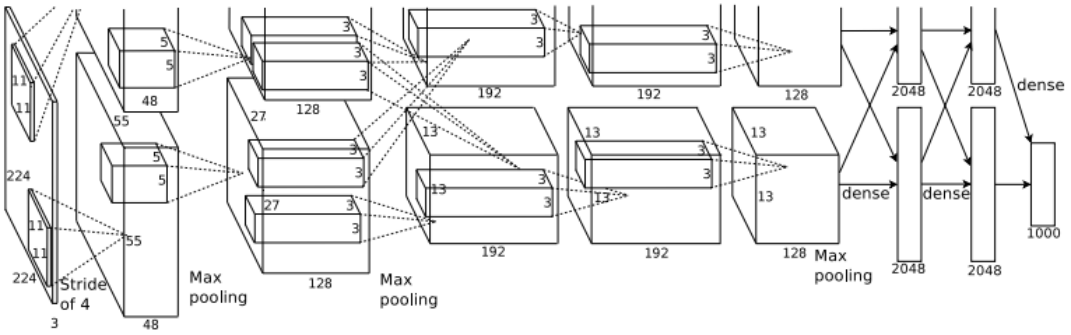

图 1: Alex-net网络结构，图片上下部分由不同GPU负责计算。GPU 只在固定层进行通信。网络的输入维度为150,528，每层神经元的个数为253,440-186,624-64,896-64,896-43,264-4096-4096-1000。

在Alex-net 中，作者提出了一些新奇的结构特征，接下来将按照其重要性递减顺序进行描述。

2.1 ReLU非线性

模拟神经元的激活函数一般形式为：$\tanh(x)$或者$sigmoid(x)$函数。这两类函数属于饱和非线性函数。所谓饱和就是当函数自变量x很大的时候，其函数值的变动很小。在使用梯度下降方法时，饱和非线性函数的收敛速度比不饱和非线性函数要慢很多。而且在每一层都需要对输入的数据做归一化处理，否则当逐层累计后输入的数据可能会变大，导致输出值的变动不大，非线性的性质被削弱。为了提高训练速度，作者们提出使用ReLU(Rectified Linear Units), 其函数形式为$f(x) = \max(0, x)$。应用在每一个卷积层和最后的全连接层。

2.2 利用多个GPU进行训练

一个GTX580的内存只有3GB，限制了可以利用其训练的网络的规模。作者们为了训练更大规模的网络，将网络一分为2，分配到2个GPU上，通过并行计算来解决，不用通过主机的缓存，GPU相互之间可以很好地进行读写操作。作者们提出一个“技巧”，只在网络的某些特定卷积层才进行GPU 之间的通信。在Alex-net中只有第三个卷积层连接第二个卷积层的提取出的所有特征，需要进行GPU 之间的通信，其他层则不用进行通信，只连接同一个GPU 中的特征。

2.3 局部归一化

LRN（Local Response Normalization）层是用来做归一化的。作者发现，虽然ReLU 层对于很大的输入x，仍然可以有效的学习，但是他们发现即使这样，对数据进行归一化对于学习来说还是有帮助的。令a_x, y^i表示在(x, y)这个位置上，以核i 来计算后，经过的激励后的输出。经过LRN 层后，作为下一层输入的数据b_x, y^i 变为：

$$b^i_{x,y} = a^i_{x,y}/(k + \alpha \sum_{j=\max(i,i-n/2)}^{\min(N-1,i+n/2)} (a^i_{x,y})^2)^\beta \tag{2.1}$$

其中N是该层的feature map总数，n表示取该feature map为中间的左右各n/2个feature map来求均值。之后的研究者发现，其实LRN并没有对准确率起到明显的提升效果，之后的很多结构中都没有使用这一层。

2.4 重叠的池化

池化层一般用于降维，将一个$k \times k$的区域内取平均或取最大值，作为这一个小区域内的特征，传递到下一层。传统的池化层是不重叠的，而作者们提出重叠的池化层可以降低错误率，而且对防止过拟合有轻微的效果，网络中采用的池化的区域大小为3×3，步长为2，取最大值。

2.5 减少过拟合

由于网络中的参数有6千万个，而训练数据只有120万张图片，很容易出现过拟合的现象。为了减少过拟合的现象，作者们采用了两种方式。

- 数据增益：对现有数据进行一定的变换，从而使得总数据量得到提升。可以对图片进行几何变换，如平移、水平翻转等。Alex-net采用了两种方法：
  - 图片裁剪与水平翻转。在训练时，从以处理后的256×256 的图片中随机抽取224×224的片段，共有1024种结果，并对抽取后的片段进行水平翻转，得到另一个图片。使用这种方法，使得训练数据扩大了2048倍。在测试的时候，从分辨率为256×256的图片的四个角及中心部位抽取出分辨率为224×224的片段，并对它们进行水平翻转，可以得到10个图片。然后将对这10个图片的预测值的平均值作为最终的预测结果。
  - 对图片的RGB通道进行强度改变。对图片进行主成分分析，然后对每个像素的RGB的修改如下公式2.2 所示：

    $$I_{xy} = [I^R_{xy}, I^G_{xy}, I^B_{xy}]^T + [p_1, p_2, p_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T \tag{2.2}$$

    λ和p分别是对像素的RGB组成的3×3相关系数矩阵的特征值和特征向量，α是服从$\mathcal{N}(0, 0.1)$的随机变量。
- Dropout：网络中的每个神经元在对每个实例进行训练的过程中以一定的概率不参与到前向传播与后向传播。在测试的过程中使用所有的神经元，但是对他们的输出乘以0.5。Alex-net中只在全连接层中的前两层使用了dropout，迭代时间增加了将近2倍，但是若不适用的话，会出现很严重的过拟合现象。

3 网络配置

[3]是在[1]提出的Alex-net的基础上，进行了一定的调整，从而探究了深度的重要性。所以，网络架构中有许多与[1] 中的配置是一样的。包括，网络输入的是固定大小为224×224 的RGB图片。唯一的预处理操作仅是对训练集中图片的每个像素减去平均的RGB 值。在一系列的卷积层后是3 个全连接层，依次有4096、4096、1000 （因为ILSVRC 中的图片有1000个类别）个神经元。最后一层是softmax输出层。与[1]相同，所有的隐藏层都使用了ReLU非线性函数。

与[1]不同的是。每个卷积层中的感受野都设置为3×3，这是能够捕获上下左右和中心概念的最小的尺寸，步长设为1。[1]中前两层个卷积层的感受野则依次为11×11和5×5。在一种网络结构中，也使用1×1 的感受野，其作用是在不影响输入输出维度的情况下，对输入进行线性变换，然后通过ReLU进行非线性处理，增加了网络的非线性表达能力，从而比较增加非线性表达和深度对准确率的影响。零填充的数量为1。池化层没有采用重叠的方式，窗口为2×2，步幅为2。共有5个池化层，配置在某些卷积层后。没有使用LRN （局部归一化）层，因为该层对准确率并没有很好的提高，还会增加内存的消耗和计算时间。

[3]中设计的所有的卷积神经网络架构如表1所示，其中每一列为一个网络结构，命名如第一行所示，为A-E。这些网络中的所有设计细节都如上面所描述的，唯一的区别便是网络的深度，从A的11层（包括8个卷积层和3个全连接层）到E的19层（包含16个卷积层和3个全连接层）。从表中可以看出，网络中有5个最大池化层，是5阶段卷积特征提取。相同池化层之间的不同网络的卷积层的channel数是相同，唯一不同的是卷积层的层数。对其他条件的统一控制，可以增加深度对实验结果的影响的可信程度。卷积层的宽度很小，首阶段为64，然后每个阶段增长一倍，直至达到最大值512，然后保持不变。表2描述了每种网络中需要训练的参数个数。

表 1: 卷积神经网络配置（按列）。网络深度从A至E逐步增加，越来越多的卷积层加入到网络中，如表格中黑体加粗部分表示。卷积层的参数表位为“conv⟨接收野⟩-channel数”。

卷积神经网络配置					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
输入(224 × 224 RGB 图片)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
最大池化层					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
最大池化层					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
最大池化层					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 (conv1-512)	conv3-512 conv3-512 textbfconv3-512	conv3-512 conv3-512 conv3-512 **conv3-512**
最大池化层					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**

表 2: 参数个数（百万）

网络	A,A-LRN	B	C	D	E
参数个数	122	133	134	138	144

易知，2个3×3的卷积层的有效感受野是5×5，3个3×3的感受野是7×7，可以替代更大的感受野。但是为什么要使用多层较小的感受野卷积层来代替一个卷积层呢？首先，多层卷积网络可以增加非线性表达的能力。另外，多个3×3的卷积层比一个大尺寸的感受野有更少的参数。假设卷积层的输入和输出的特征图的大小相同为C，那么3个3×3 的卷积层参数个数为$3 \times (3 \times 3 \times C \times C) = 27C^2$,而一个$7 \times 7$的卷积层参数为$49C^2$，多了$81\%$。

4 训练与测试

4.1 训练

训练其实就是使用动量利用批梯度下降方法优化多项式逻辑回归目标函数的过程。训练过程与[1]类似，批梯度的规模设为256，动量设为0.9，利用权重衰减进行训练的调整。在全连接层的前两层使用dropout方法，选择概率为0.5。学习速率初始化为10^{-2}，之后当验证集停止对准确率有提升时，手动将学习速率减少10倍。总共，学习率下降了3次，大概在37万次迭代（74个周期）后停止学习。虽然网络中的参数更多、深度也更深，但达到收敛的时间却更少了，作者们对此进行了一定的猜测，他们提出两个可能原因：首先是更深更小的卷积感受野起到了隐式的规则化的作用；其次是某些层的预初始化的处理。接下来针对第二个原因，进行一定的说明。

网络权重的初始化十分关键，因为糟糕的初始化可能会由于网络中梯度的不稳定性而使学习停滞。为了解决这个问题，作者们先训练了较为浅层的网络，也就是A，可以使用随机初始化的方式进行训练。利用A中训练后得到的参数对其他深层网络的前4层卷积层和后面3层全连接层的参数进行初始化，对于这些预初始化的层，不减少其学习率以使得其可以在学习过程中继续改变。网络中部的卷积层的参数则使用随机初始化的方法，权值是服从$(0, 10^{-2})$的正态分

布的随机变量。偏置设为0。作者们在提交论文后发现也可以不使用预初始化的方法，而是使用Glorot和Bengio在2010 年提出的随机初始化方法来对权值进行初始化。

从缩放后的训练集图片中随机取大小为224×224的碎片输入到ConvNet中，然后进行随机梯度下降的迭代。为了让训练数据量变大，采用了与[1]中一样的数据增益的方法（见2.5）。接下来主要讨论如何对训练集图片进行缩放。

设S是各向同性缩放后的训练集图片的最小边的大小，为了能够从中得到224×224的图片，原则上$S \geq 224$。若$S \gg 224$，则碎片将只对应原始图片的一小部分，可能只是要识别物体的一部分。作者提出2种设置S的方法。

- 固定S值，也就是单一尺度的训练集数据，在采样得到的224×224的碎片中的图片内容仍然具有多尺度统计特性的。实验中，设置了两个固定的尺度，$S = 256$（与[1]相同）和$S = 384$。先用$S = 256$的训练集数据训练网络。为了加快训练，在利用$S = 384$的训练集数据训练网络时，其权重的初始化使用$S = 256$得到的参数结果，学习速率初始化为10^{-3}。

- 多尺度训练。$S \in [S_{min}, S_{max}]$，针对每个训练集图片都对S进行随机抽样，根据得到的S值确定图片的缩放比例，对其进行缩放。论文中作者设置$S_{min} = 256$，$S_{max} = 512$。这种方式也可以看做是在图片尺度上的数据增益。为了保证训练速度，同样在训练过程中也使用预初始化的方法，使用固定尺度中的$S = 384$得到的参数值来进行初始化。

4.2 测试

首先，将测试图片各向同性缩放至最小边大小为预先定义好的测试尺度Q，Q不必与S相等。与[1]中在缩放后为256×256的图片中抽取4个角和中心部位的224×224大小的图片碎片及其水平翻转的图片，分别进行测试然后预测结果求均值的方法不同。论文采用的是[2]中提到的测试方法，即多尺度分类。在测试时，作者将后三层的全连接层转换为卷积层，可看做为分类层，第一个全连接层变为感受野为7×7 的卷积层，后两个全连接层则变为感受野为1×1的卷积层，使得整个网络变为一个全部由卷积层组成的网络，包括特征抽取层和分类层。因为网络输入的图片大小为224×224，整个网络经过5个步幅为2池化层，其他卷积层步幅均为1，所以相当于降采样频率为$2^5 = 32$，经过最后一个卷积阶段池化后的feature map的每条边大小为$224/32 = 7$，feature map的尺寸为7×7，按照[2]的想法，故将第一个变为卷积层的全连接层的感受野设为7×7。

测试过程中直接在整个图片上进行测试，将整张图片作为输入，在特征抽取层的最后一个卷积阶段产生feature map。由于图片大小不同，所以feature map 的大小也不同，在分类层的第一个卷积处针对这些尺度不同的feature map，采用密集滑动窗口的方式，经过后面2 层1×1 的卷积层，最终可以得到针对该窗口的图片分类。假设图片经过最后一个卷积阶段的feature map大小为8×8，因为分类层的第一层卷积层的感受野为7×7，所以滑动窗口的大小为7×7，可知该feature map上的滑动窗口的个数为2×2，最终经过后面两个1×1 的卷积层的输出结果也是2×2，即每个值对应其窗口的预测。最后将这2×2个结果进行融合，求均值，作为整个图片的预测结果。这种方法相比[1]中的测试方法要更加高效，因为只要进行一次卷积过程就可以。而且，ImageNet图片中待分类识别的物体通常出现在图片的中心部位，且所占比例较大，但是现实生活中的图片并不都是这么完美，明显仅能识别中心部分的物体类别是不能满足现实需要的。而这种多尺度测试方法中输入照片的尺度不同，会使得滑动窗口得到的窗口的物体位置、大小等更符合实际情况，而不仅仅是物体分布在图片的正中央，这使得整个系统具有更好的鲁棒性。另外，作者们也会对原始图片进行水平翻转，从而实现测试数据的增益。

虽然密集采样的方式比[1]中使用的测试方法，先将图片分为多块满足要求的大小再输入到网络中的方法要高效一些，但是[4]中使用这种多块采样的方法提高了系统的准确率。一方面是因为采样的输入图片更好；另一方面可能是由于两种方式的卷积填充状况不同。密集采样方式中，卷积填充的是图片周围区域的结果，在一定程度上增加了整个网络的感受野，所以更多的内容被捕获。而多块采样的方式则只是填充零值。为了对比量两种方式的差异，作者也进行了多块采样的实验。每个尺度的测试图片设置了50个抽样块（实际上是5×5，并对这些抽样块进行水平翻转），在3个尺度上一共有150个抽样块。

4.3 实现细节

利用C++ Caffe toolbox进行项目的开发，在4个NVIDIA Titan Black GPU上并行计算，比单独的GPU快3.75倍，每个网络大约要训练2 3周的时间。

5 实验结果

数据集：使用ILSVRC-2012的数据集，该数据集中包括130万训练集，5万验证集，10万测试集。大多数的实验中，作者使用验证集作为测试集。某些实验也在测试集上进行测试，并将结果提交了官方ILSVRC服务器，作为"VGG"团队的参赛模型。分类性能由$top-1$和$top-5$错误率来进行衡量。

5.1 单一尺度测试

该部分实验使用单一的测试尺度Q。针对固定的S，令$Q=S$；针对多尺度S，令$Q=0.5(S_{min}+S_{max})$。实验结果如表3所示。

表 3: 单一测试尺度CNN性能

网络结构	图片最小边尺寸		$top-1$错误率	$top-5$错误率
	训练(S)	测试(Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

观察上表可以得出以下结论：

- A vs A-LRN : A-LRN的结果并没有比A好，说明LRN作用不大，因此作者没有在更深的网络B-E 中使用LRN。
- A,B,C,D,E之间结果对比: 网络结构越深，效果越好。
- B vs C ：C的效果更好。C比B中增加了3个1×1的卷积层，说明增加额外的非线性表示能够提高准确率。
- C vs D ：D的效果明显好于C。D与C的不同是将3个1×1的卷积层变为3×3的卷积层。这说明增加具有适当的感受野的卷积层比增加非线性表示更加重要。
- 当网络结构为19层时，准确率接近饱和，然而更深的网络可以应用于容量更大的数据库。
- 单一S vs 多尺度S：在尺度区间$[256;512]$通过随机尺度S增益来训练网络，比固定两个$S=256$和$S=512$，虽然，测试时使用的都是单一尺度图片，但是结果仍然有明显提升。说明多尺度的训练确实对捕获图片的不变形有很大作用。

另外，作者也使用了一个浅层的只有5个感受野均为5×5的卷积神经网络进行了比较，这个浅层网络是用一个感受野为5×5的卷积层来替换2个连续的无池化的3×3的卷积层得到的。结果显示浅层网络的$top-1$错误率要比网络B高出7%。结果证实了使用多个较小感受野的卷积层深层网络比使用大感受野的浅层网络效果要好。

5.2 多尺度测试

实验中，选取3个Q值进行多尺度的测试分类，将图片分别缩放至三个尺度，然后按照4.2中所讲的测试方法进行测试，最后将后验概率求均值即可。为了防止训练和测试尺度之间的差距过大而导致效果不好，作者对Q 进行了一定的规定。针对固定的训练尺度S，相应的测试尺度$Q=S-32,S,S+32$；针对随机训练尺度$S\in[S_{min},S_{max}]$，$Q=S_{min},0.5(S_{min}+S_{max}),S_{max}$。实验结果如表4所示。

表 4: 多测试尺度CNN性能

网络结构	图片最小边尺寸		$top-1$错误率	$top-5$错误率
	训练(S)	测试(Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256;512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256;512]	256,384,512	**24.8**	**7.5**
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256;512]	256,384,512	**24.8**	**7.5**

观察上表可以得出以下结论：

- 表3 vs 表4：表4结果要更好，说明多尺度测试性能更好。
- 单一S vs 多尺度S：多尺度训练效果比单一尺度训练效果要好。说明多尺度训练更有助于获得图片的不变性，网络在单一尺度上的提取能力有限。
- B,C,D,E之间对比：深度越深，错误率越低。与单一尺度测试结果类似，当网络为19层时，错误率基本达到了饱和。

5.3 评估方法比较

本部分实验主要针对4.2部分所提到的针对测试图片的密集采样评估方法以及多块采样评估的方法进行比较分析。作者们也将两种方式结合在一起进行了实验，将两种方式得到的后验概率求均值作为最终结果。实验结果如表5 所示。

表 5: 卷积网络测试评估方法比较。所有实验中训练尺度$S \in [256; 512]$，测试尺度$Q = 256, 384, 512$。

网络结构	评估方法	$top-1$错误率	$top-5$错误率
D	密集采样	24.8	7.5
	多块采样	24.6	7.5
	多块& 密集	**24.4**	**7.2**
E	密集采样	24.8	7.5
	多块采样	24.6	7.4
	多块& 密集	**24.4**	**7.1**

可以看到使用单独使用多块采样评估的方式错误率较密集采样评估的方式错误率低了0.1%，二者并没有太大的区别。但是结合两种评估方法的结果是最好的。作者们猜测这可能是与4.2中提到的二者之间卷积边界状态的不同导致的。

5.4 模型融合

之前的实验都是针对单一的网络，接下来的实验中作者将多个不同的网络进行融合，然后对每个网络在softmax输出层的后验概率求均值，作为融合后模型的输出结果。提交到ILSVRC参赛的模型是融合了7个网络，包括3个D网络、2个C网络和2个E网络，每个网络的训练和测试策略如表6所示。在提交参赛模型后，作者们又考虑到只融合分类效果最好的2个网络($D/[256; 512]/256, 384, 512$)和($E/[256; 512]/256, 384, 512$)。但是采用不同的测试评估方法，结果如表6所示。有意思的是，两个顶尖模型融合的结果比融合7个模型的结果还要好。

表 6: 模型融合结果。其中，“$top-5$ test”是在ILSVRC测试集上进行测试的结果，“$top-1$ val.”与“$top-5$ val.”列是在ILSVRC验证集上进行测试的结果。

融合模型	错误率		
	$top-1$ val.	$top-5$ val.	$top-5$ test
ILSVRC submission			
(D/256/224,256,288),(D/384/352,384,416),(D/[256;512]/256,384,512) (C/256/224,256,288),(C/384/352,384,416) (E/256/224,256,288),(E/384/352,384,416)	24.7	7.5	7.3
post-submission			
(D/[256;512]/256,384,512),(E/[256,512]/256,384,512),dense eval	24.0	7.1	7.0
(D/[256;512]/256,384,512),(E/[256,512]/256,384,512),multi-crop	23.9	7.2	-
(D/[256;512]/256,384,512),(E/[256,512]/256,384,512),multi-crop &dense eval	**23.7**	**6.8**	**6.8**

5.5 与其他网络的比较

在ILSVRC-2014挑战中的分类任务中，作者提出的”VGG”模型获得了第二名的成绩，融合了7个模型，在测试集上的错误率为7.3%。提交模型后，作者们又通过融合2个模型将错误率降至”6.8%”。下表7显示了“VGG”模型与当今顶级模型结果的比较，可以看出“VGG”模型的性能还是不错的。

表 7: 与其他分类网络的比较

Method	$top-1$ val. error	$top-5$ val. error	$top-5$ test error
VGG(2 nets, multi-crop & dense)	**23.7**	**6.8**	**6.8**
VGG(1 net, multi-crop & dense)	24.4	7.1	7.3
VGG(ILSVRC submission, 7 nets, dense)	24.7	7.5	7.3
GoogLeNet(Szegedy et al.,2014)(1 net)	-	7.9	
GoogLeNet(Szegedy et al.,2014)(1 net)	-	**6.7**	
MSRA(He et al.,2014)(11 nets)	-	-	8.1
MSRA(He et al.,2014)(1 net)	27.9	9.1	9.1
Clarifai(Fussakovsky et al.,2014)(multiple nets)	-	-	11.7
Clarifai(Fussakovsky et al.,2014)(1 net)	-	-	12.5
Zeiler&Fergus(Zeiler&Fergus,2013)(6 nets)	36.0	14.7	14.8
Zeiler&Fergus(Zeiler&Fergus,2013)(1 net)	37.5	16.0	16.1
OverFeat(Sermanet et al.,2014)(7 nets)	34.0	13.2	13.6
OverFeat(Sermanet et al.,2014)(1 net)	35.7	14.2	-
Krizhevsky et al.(Krizhevsky et al.,2012)(5 nets)	38.1	16.4	16.4
Krizhevsky et al.(Krizhevsky et al.,2012)(1 net)	40.7	18.2	-

6 小结与讨论

这篇论文通过构造具有一定深度的卷积神经网络（最深至19层），每一个卷积层都使用较小的3×3的感受野，采用[1]和[2]中提到的训练和测试策略，通过大量实验证明相比浅层的具有较大感受野的卷积神经网络，深层的具有较小感受野的卷积神经网络分类效果更佳。这可能是因为深层网络中非线性表达的能力更强。作者的工作也再一次证明了卷积神经网络中深度的重要性，回答了“卷积神经网络真的有必要是深度的吗”这一问题。

参考文献

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[2] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 2013.

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[4] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.