# 挤压激励网络

Jie Hu[1]*
hujie@momenta.ai

Li Shen[2]*
lishen@robots.ox.ac.uk

Gang Sun[1]
sungang@momenta.ai

[1] 时刻

[2]牛津大学工程科学系

## 摘要

*卷积神经网络建立在卷积运算的基础上，它通过在局部感受野中融合空间和信道信息来提取信息特征。为了提高网络的表征能力，最近有几种方法显示了增强空间编码的好处。在这项工作中，我们将重点放在通道关系上，并提出了一种新颖的结构单元，我们称之为 "挤压和激发"（SE）区块，它通过明确模拟通道之间的相互依存关系，自适应地重新校准通道特征响应。我们证明，通过将这些区块堆叠在一起，我们可以构建 SENet 架构，该架构在具有挑战性的数据集上具有极好的通用性。最重要的是，我们发现 SE 模块能以最低的额外计算成本显著提高现有先进深度架构的性能。SENets 是我们提交的 2017 年 ILSVRC 分类报告的基础，该报告获得了第一名，并将前五名的错误率大幅降低到 2.251%，比 2016 年的获奖报告相对提高了 25%。代码和模型可在 https://github.com/hujie-frank/SENet.*

## 1. 导言

事实证明，卷积神经网络（CNN）是处理各种视觉任务的有效模型[21, 27, 33, 45]。对于每个卷积层，都要学习一组滤波器，以表达沿输入通道的局部空间连接模式。换句话说，卷积滤波器是通过将空间信息和通道信息融合到低感受野中而形成的信息组合。通过堆叠一系列卷积层，交错使用非线性和降采样，CNN 能够捕捉具有全局感受野的分层模式，作为强大的图像描述。最近的研究表明，通过明确嵌入学习功能，可以提高网络的性能。

*平等贡献。

这些机制有助于捕捉空间相关性，而不需要额外的监督。Inception 架构[16, 43]就推广了这样一种方法，该架构表明，通过在其模块中嵌入多尺度过程，网络可以实现具有竞争力的准确性。最近的研究试图更好地模拟空间依赖性[1, 31]，并纳入空间注意力[19]。

在本文中，我们通过引入一种新的结构单元（我们称之为 "*挤压与激励*"（SE）模块）来研究结构设计的另一个方面--通道关系。我们的目标是通过明确模拟卷积特征通道之间的相互依存关系，提高网络的再现能力。为了实现这一目标，我们提出了一种允许网络执行*特征重新校准的*机制，通过这种机制，网络可以学会使用全局信息来强调有信息价值的特征，并抑制不那么有用的特征。

SE 构建模块的基本结构如图 1 所示。对于任何给定的变换 $F_{tr} : X \rightarrow U$、$X \in R^{H \times W \times C}$，$U \in R^{H \times W \times C}$，（例如一个卷积或一组卷积），我们可以构建一个相应的 SE 块来执行特征重校准，如下所示。首先将特征 U 通过一个*挤压*操作（squeeze opera-tion），该操作将跨空间范围 $H \times W$ *的*特征图聚合在一起，生成一个信道描述符（channel descriptor）。该描述符包含了通道式特征响应的总体分布，从而使网络的下层结构能够利用来自总体再感受场的信息。随后是*激发*操作，通过基于通道依赖性的自门控机制学习每个通道的特定样本激活，控制每个通道的激发。然后对特征图 U 进行重新加权，生成 SE 模块的输出，然后直接输入到后续层。

只需堆叠一组 SE 构件，即可生成 SE 网络。SE 构件还可以在架构的*任何深度*上替代原始构件。然而，正如我们在第 6.4 节中所展示的那样，虽然构建模块的模板是通用的，但它在不同深度所扮演的角色却能适应网络的需要。在早期层，它学会激发信息

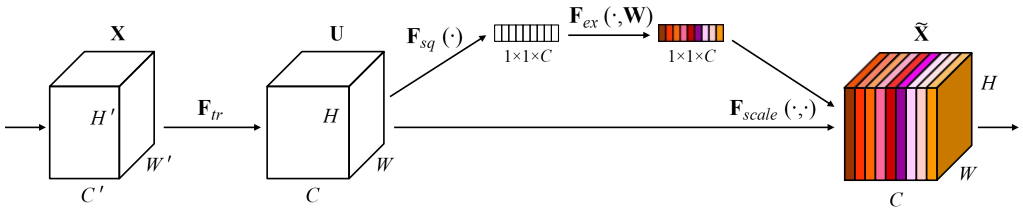



图 1：挤压-激发模块。

这些特征与类别无关，从而提高了共享的低层表征的质量。在后面的层中，SE 块变得越来越专业化，并以高度*特定于类别的*方式对不同的输入做出响应。因此，由 SE 块进行特征重新校准所带来的好处可以在整个网络中累积。

开发新的 CNN 架构是一项艰巨的工程任务，通常需要选择许多新的超参数和层配置。相比之下，上述 SE 模块的设计非常简单，可直接用于现有的最先进架构，其模块可通过直接替换为 SE 对应模块而得到加强。

此外，如第 4 章所示，SE 块在计算上是轻量级的，只略微增加了模型复杂度和计算负担。为了支持这些说法，我们开发了多个 SENets，并在 ImageNet 2012 数据集 [34] 上进行了广泛评估。为了证明其普遍适用性，我们还展示了 ImageNet 以外的结果，表明所提出的方法并不局限于特定的数据集或任务。

利用 SENets，我们赢得了 2017 年 ILSVRC 分类竞赛的第一名。我们表现最好的模型集合在测试集上取得了 2.251% 的前五名错误率。[1].这与上一年的冠军作品（前五名的误差为 2.991%）相比，相对提高了 25%。

## 2. 相关工作

深度架构。VGGNets [39] 和 Inception mod- els [43] 证明了增加深度的好处。批量规范化（BN）[16] 通过插入单元来调节层输入，稳定了学习过程，从而改进了梯度预测。ResNets [10, 11] 通过使用基于身份的跳转连接，展示了学习更深层网络的有效性。高速公路网络 [40] 采用门控机制来调节捷径连接。网络层间连接的重构[5, 14]已被证明能进一步改善深度网络的学习和表征特性。

另一个研究方向是探索如何调整网络模块组件的功能形式。分组卷积可用于提高汽车

[1] http://image-net.org/challenges/LSVRC/2017/results

dinality（变换集的大小）[15, 47]。多分支卷积可以解释为这一概念的概括，使运算符的组合更加灵活[16, 42, 43, 44]。最近，以自动方式学习的组合[26, 54, 55]已经显示出具有竞争力的性能。跨信道卷积关系通常被映射为新的特征组合，可以独立于空间结构[6, 20]，也可以通过使用标准卷积滤波器[24]与 1×1 卷积联合映射。这些工作大多集中在降低模型和计算复杂度的目标上，反映了一种假设，即信道关系可以表述为具有局部感受野的实例无关函数的组合。相比之下，我们认为，为单元提供一种机制，利用全局形成明确地模拟通道之间的动态非线性依赖关系，可以简化学习过程，并显著增强网络的表征能力。

注意和门控机制。从广义上讲，注意力可以被视为一种工具，用于将可用的处理资源向输入信号中信息量最大的部分倾斜分配[17, 18, 22, 29, 32]。从图像定位和理解 [3, 19] 到基于序列的模型 [2, 28]，这种机制的优势已在一系列任务中得到证实。它通常与门控函数（如软最大值或sigmoid）和序列技术结合使 用 [12, 41]。研究表明，它适用于图像字幕[4, 48]和读唇[7]等任务。在这些应用中，它通常被用在一个或多个代表更高层次抽象的层之上，以实现不同模态之间的适应。Wang 等人[46]利用沙漏模块[31]引入了一种功能强大的主干和掩码注意机制。这种高容量单元被插入到中间阶段之间的深度残 差 网络中。相比之下，我们提出的 SE 模块是一种轻量级门控机制，专门用于以计算效率高的方式模拟通道关系，旨在增强整个网络中基本模块的代表能力。

## 3. 挤压和激发块

*挤压-激发*模块是一个计算单元，可以为任何给定的变换构建

$f_{tr}: x \rightarrow u,\ x \in r^{H \times W \times C'},\ u \in r^{H \times W \times C}$。对于

为简单起见，在接下来的符号中，我们将 $F_{tr}$ 视为卷积算子。让 $V = [v_1, v_2, \ldots, v_C]$ 表示学习到的滤波器内核集，其中 $v_c$ 指*的是第* c *个*滤波器的参数。然后，我们可以将 $F_{tr}$ 的输出写成 $U = [u_1, u_2, \ldots, u_C]$，其中

$$u_c = v_c * X = \sum_{s=1}^{C'} v_c^s * x^s. \quad (1)$$

这里 $*$ 表示卷积，$v_c = [v_c^1, v_c^2, \ldots, v_c^{C'}]$ 和 $X = [x^1, x^2, \ldots, x^{C'}]$（为简化符号，偏置项省略），而 $v_c^s$ 是一个二维空间核，因此代表 $v_c$ 它对相关的由于输出是由通过所有信道求和，信道依赖于 $v_c$ 中隐含了这些依赖关系，但这些依赖关系与滤波器捕捉到的空间相关性纠缠在一起。我们的目标是确保网络能够提高对信息特征的灵敏度，以便在后续转换中加以利用，并抑制不那么有用的特征。为了实现这一目标，我们建议对信道相互依存关系进行明确建模，以便在将滤波器响应反馈到下一次转换之前，分*挤压*和*激励*两个步骤对滤波器响应进行重新校准。图 1 是 SE 构建模块的示意图。

### 3.1. 挤压全球信息嵌入

为了解决利用通道依赖性的问题，我们首先考虑输出特性中每个通道的信号。每个学习到的滤波器都有一个本地感受野，因此，转形成输出 U 的每个单元都无法利用该区域之外的上下文信息。这个问题在低层网络中变得更加严重，因为低层网络的感受野很小。

为缓解这一问题，我们建议将全局空间信息*挤压*到信道描述符中。这是通过使用全局平均池生成通道统计来实现的。形式上，统计量 $z \in R^C$ 是通过空间维度 $H \times W$ 缩减 U 而生成的，其中 z 的*第 c 个*元素通过以下方式计算：

$$z_c = F_{平方}(u_c) = \frac{1}{高 \times 宽} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j). \quad (2)$$

*讨论*转换输出 U 可以被理解为局部描述符的集合，其统计信息对整个图像都有表现力。利用此类信息在特征工程工作中非常普遍 [35, 38, 49]。我们选择了最简单的全局平均集合，同时也注意到这里也可以采用更复杂的集合策略。

### 3.2. 激励：自适应重新校准

利用*挤压*中汇总的信息操作之后，我们还要进行第二个操作，目的是以充分捕捉渠道间的依赖关系。为了实现这一目标，函数必须满足两个标准：首先，它必须具有灵活性（特别是，它必须能够学习通道之间的非线性相互作用）；其次，它必须学习非相互排斥的关系，因为我们希望确保允许多个通道被激活，而不是一次性激活。为了满足这些要求，我们选择采用一种简单的门控机制，它具有以下功能

乙状结肠激活

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)), \quad (3)$$

其中 $\delta$ 指 ReLU [30] 函数，$W_1 \in R^{\frac{C}{r} \times C}$ 和 $W_2 \in R^{C \times \frac{C}{r}}$。为了限制模型的复杂性并帮助我们对门控机制进行了参数化，在非线性周围形成了一个具有两个全连接 (FC) 层的瓶颈，即一个具有 $W_1$ 和缩减比 $r$ 的缩减维度层（参数选择将在第 6.4 节中讨论）、一个 ReLU，然后是一个具有 $W_2$ 的增大维度层。块的最终输出是通过将转换输出 U 与激活进行重新缩放而获得的：

$$\tilde{x}_c = F_{标度}(u_c, s_c) = s_c \cdot u_c, \quad (4)$$

其中，$\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_C]$，$F_{scale}(u_c, s_c)$ 指的是特征图 $u_c \in R^{H \times W}$ 与标量 $s_c$ 之间的信道乘法。

*讨论*在这方面，SE 块内在地引入了以输入为条件的动态，有助于提高特征的可辨别性。

### 3.3.示例：SE-Inception 和 SE-ResNet

将 SE 模块应用于 AlexNet 非常简单 [21] 和 VGGNet [39]。SE 块的灵活性意味着它可以直接应用于标准卷积以外的变换。为了说明这一点，我们通过将 SE 块集成到具有复杂设计的现代架构中来开发 SENets。

对于非驻留网络，如 Inception 网络、网络的 SE 块是通过提取转换 $F_{tr}$ 为整个 Inception 模块（见图 2）。 对每个这样的模块进行这样的修改

因此，我们构建了一个 *SE-Inception* 网络。此外，SE 模块具有足够的灵活性，可用于残差网络。图 3 描述了 *SE- ResNet* 模块的模式。在这里，SE 块变换 $F_{tr}$ 被视为残差模态的非同一性分支。*挤压*和*激励*都是在与同一分支相加之前起作用。按照类似的方案，还可以构建与 ResNeXt [47]、Inception-ResNet [42]、MobileNet [13] 和 ShuffleNet [52] 集成的更多变体。我们在表 1 中描述了 SE-ResNet-50 和 SE-ResNeXt-50 的架构。

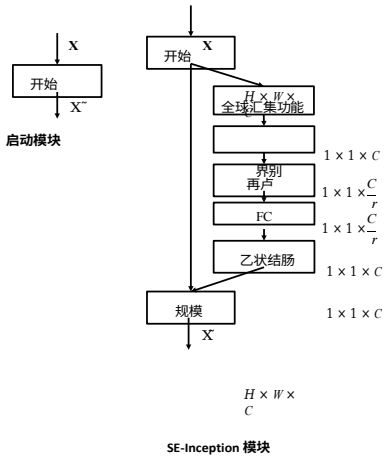



图 2：原始 Inception 模块（左）和 SE-Inception 模块（右）的结构图。

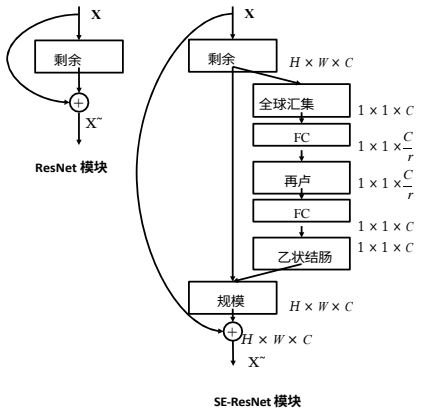



图 3：原始残差模块（左）和 SE-ResNet 模块（右）的结构图。

## 4. 模型和计算复杂性

为使所提出的 SE 模块在实践中可行，它必须在模型复杂性和性能之间进行有效权衡，这对可扩展性非常重要。在所有实验中，我们都将缩减率 $r$ 设为 16，除非另有说明（更多讨论见第 6.4 节）。为了说明模块的成本，我们以 ResNet-50 和 SE-ResNet-50 的比较为例，SE-ResNet-50 的准确度高于 ResNet-50，接近于更深的 ResNet- 101 网络（如表 2 所示）。对于 224 × 224 像素的图像，ResNet-50 一次前向处理需要 3.86 GFLOP。每个 SE 块在*挤压*阶段使用了全局平均池化操作，在*激励*阶段使用了两个小型全连接层，然后进行了廉价的通道扩展操作。总的来说，SE-ResNet-50 需要 3.87 GFLOPs，相对于原始 ResNet-50 增加了 0.26%。

在实际应用中，训练小批量图像的数量为 256 张、ResNet-50 的单次前后传递耗时 190 毫秒，而 SE-ResNet-50 的耗时为 209 毫秒（这两个计时均在配备 8 个英伟达 Titan X GPU 的服务器上执行）。我们认为这是一个合理的开销，特别是因为全局池和小型内积操作在现有 GPU 库中的优化程度较低。此外，鉴于其对嵌入式设备应用的重要性，我们还对每个模型的 CPU 推理时间进行了基准测试：对于 224 × 224 像素的输入图像，ResNet-50 需要 164 毫秒，而 SE-ResNet-50 需要 167 毫秒。SE 块所需的额外计算开销很小，但它对模型性能的贡献是合理的。

接下来，我们将考虑拟议区块引入的附加参数。所有这些参数都包含在门控机制的两个 FC 层中，只占网络总容量的一小部分。更准确地说，引入的附加参数数量为

$$\frac{2}{r}\sum_{s=1}^{S} N_s \cdot C_s^2 \tag{5}$$

其中，$r$ 表示缩减率，$S$ 指的是"..... "的数量。
$C_s$ 表示输出通道的维度。

和 $N_s$ 表示阶段 $s$ 的重复区块数。

除了 ResNet-50 所需的 2,500 万个参数外，ResNet-50 还引入了 250 万个额外参数，相当于增加了 10%。这些参数中的大部分来自网络的最后阶段，在这一阶段，激励在最大的通道维度上进行。不过，我们发现，可以以微小的性能代价（ImageNet 上的 top-1 误差小于 0.1%）移除 SE 块中相对昂贵的最后阶段，从而将相对参数增幅降至 ~4%，这在参数使用是关键考虑因素的情况下可能会很有用（见第 6.4 节的进一步讨论）。

## 5. 实施情况

每个普通网络及其相应的 SE 对应部分都采用相同的优化方案进行训练。在 ImageNet 上进行训练期间，我们按照标准做法，通过随机裁剪 [43] 到 224 × 224 像素（Inception-ResNet-v2 [42] 和 SE-Inception-ResNet-v2 为 299 × 299）和随机水平翻转来增强数据。输入图像通过均值减法进行归一化处理。此外，我们还采用了 [36] 中所述的数据平衡策略进行迷你批量采样。网络在我们的分布式学习系统 "ROCS "上进行训练，该系统旨在处理大型网络的高效并行训练。优化使用同步 SGD 进行，动量为 0.9，迷你批次大小为 1024。初

始学习率设置为 0.6，每 30 个历元降低 10 倍。所有模型的训练

| 输出尺寸 | ResNet-50 | SE-ResNet-50 | SE-ResNeXt-50 (32 × 4d) |
|---|---|---|---|
| 112 × 112 | 信念，7 × 7，64，步长 2 | | |
| 56 × 56 | 最大泳池，3 × 3，跨步 2 | | |
| | conv，1 × 1，64<br>conv，3 × 3，64<br>conv，1 × 1，256 × 3 | conv，1 × 1，64<br>conv，3×3，64<br>conv，1 × 1，256<br>*FC*，[16，256] × 3 | conv，1 × 1，128<br>conv，3 × 3，128<br>conv，1 × 1，256<br>*FC*，[16，256] $C = 32$ × 3 |
| 28 × 28 | conv，1 × 1，128<br>conv，3 × 3，128<br>conv，1 × 1，512 × 4 | conv，1 × 1，128<br>conv，3 × 3，128<br>conv，1 × 1，512<br>*FC*，[32，512] × 4 | conv，1 × 1，256<br>conv，3×3，256<br>conv，1 × 1，512<br>*FC*，[32，512] $C = 32$ × 4 |
| 14 × 14 | conv，1 × 1，256<br>conv，3 × 3，256<br>conv，1×1，1024 × 6 | conv，1 × 1，256<br>conv，3×3，256<br>conv，1 × 1，1024<br>*FC*，[64，1024] × 6 | conv，1 × 1，512<br>conv，3×3，512<br>conv，1 × 1，1024<br>*FC*，[64，1024] $C = 32$ × 6 |
| 7×7 | conv，1 × 1，512<br>conv，3 × 3，512<br>conv，1 × 1，2048 × 3 | conv，1 × 1，512<br>conv，3×3，512<br>conv，1 × 1，2048<br>*FC*，[128，2048] × 3 | conv，1 × 1，1024<br>conv，3×3，1024<br>conv，1 × 1，2048<br>*FC*，[128，2048] $C = 32$ × 3 |
| 1 × 1 | 全局平均池，1000 d *fc*，softmax | | |

表 1：（左）ResNet-50。(中）SE-ResNet-50。(右）使用 32×4d 模板的 SE-ResNeXt-50。括号内列出了残差构件的形状和操作以及具体参数设置，括号外列出了一个阶段中堆叠构件的数量。括号内的 *fc* 表示 SE 模块中两个全连接层的输出尺寸。

| | 原创 | | 重新实施 | | | SENet | | |
|---|---|---|---|---|---|---|---|---|
| | TOP-1 Er. | 前 5 名错误。 | top-1err. | 前 5 名错误。 | GFLOPs | TOP-1 Er. | 前 5 名错误。 | GFLOPs |
| ResNet-50 [10] | 24.7 | 7.8 | 24.80 | 7.48 | 3.86 | $23.29_{(1.51)}$ | $6.62_{(0.86)}$ | 3.87 |
| ResNet-101 [10] | 23.6 | 7.1 | 23.17 | 6.52 | 7.58 | $22.38_{(0.79)}$ | $6.07_{(0.45)}$ | 7.60 |
| ResNet-152 [10] | 23.0 | 6.7 | 22.42 | 6.34 | 11.30 | $21.57_{(0.85)}$ | $5.73_{(0.61)}$ | 11.32 |
| ResNeXt-50 [47] | 22.2 | - | 22.11 | 5.90 | 4.24 | $21.10_{(1.01)}$ | $5.49_{(0.41)}$ | 4.25 |
| ResNeXt-101 [47] | 21.2 | 5.6 | 21.18 | 5.57 | 7.99 | $20.70_{(0.48)}$ | $5.01_{(0.56)}$ | 8.00 |
| VGG-16 [39] | - | - | 27.02 | 8.81 | 15.47 | $25.22_{(1.80)}$ | $7.70_{(1.11)}$ | 15.48 |
| BN-Inception [16] | 25.2 | 7.82 | 25.38 | 7.89 | 2.03 | $24.23_{(1.15)}$ | $7.14_{(0.75)}$ | 2.04 |
| Inception-ResNet-v2 [42] | 19.9† | 4.9† | 20.37 | 5.21 | 11.75 | $19.80_{(0.57)}$ | $4.79_{(0.42)}$ | 11.76 |

表 2：ImageNet 验证集上的单次裁剪错误率（%）和复杂度比较。*原始*列是指原始论文中报告的结果。为了进行公平比较，我们重新训练了基线模型，并在*重新实现*一栏中报告了得分。*SENet* 一栏指的是添加了 SE 块的相应架构。括号中的数字表示与重新实现的基线相比的性能改进。† 表示在验证集的非黑名单子集上对模型进行了评估（[42] 对此有更详细的讨论），这可能会略微改善结果。VGG-16 和 SE-VGG-16 采用批量标准化训练。

使用 [9] 中描述的权重初始化策略，从头开始计算 100 个历时。

测试时，我们对验证集进行了中心裁剪评估，从每幅图像中裁剪出 224×224 像素，其短边首先调整为 256（Inception-ResNet-v2 和 SE-Inception-ResNet-v2 从每幅图像中裁剪出 299×299 像素，其短边首先调整为 352）。

## 6. 实验

### 6.1. 图像网络分类

ImageNet 2012 数据集由来自 1000 个类别的 128 万张训练图像和 5 万张验证图像组成。我们在训练集上训练网络，并报告误差前 1 名和前 5 名的情况。

网络深度。我们首先将 SE-ResNet 与不同深度的 ResNet 架构进行比较。表 2 中的结果表明，在不同深度下，SE 块始终能提高性能，而计算复杂度却只有极小的下降。

值得注意的是，SE-ResNet-50 的单作物 Top-5 验证误差为 6.62%，比 ResNet-50 (7.48%) 高出 0.86%，接近更深的 ResNet-101 网络 (6.52% Top-5 误差) 的性能，而计算开销仅为后者的一半（3.87 GFLOPs 对 7.58 GFLOPs）。这种模式在更深的网络中重复出现，SE-ResNet-101（6.07% 的前五名错误率）不仅与更深的 ResNet-152 网络（6.34% 的前五名错误率）不相上下，而且还比后者高出 0.27%。图 4 描述了 SE-ResNet-50 和 ResNet-50 的训练和验证曲线（SE-ResNet-50 和 ResNet-50（SE-ResNet-50 和 ResNet-50（SE-ResNet-50

)) 。

| | 原创 | | 重新实施 | | | | SENet | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 第一名 呃。 | 前五名 呃。 | 第一名 呃。 | 前五名 呃。 | MFLOPs | 百万 参数 | 第一名 呃。 | 前五名 呃。 | MFLOPs | 百万 参数 |
| 移动网[13] | 29.4 | - | 29.1 | 10.1 | 569 | 4.2 | $25.3_{(3.8)}$ | $7.9_{(2.2)}$ | 572 | 4.7 |
| ShuffleNet [52] | 34.1 | - | 33.9 | 13.6 | 140 | 1.8 | $31.7_{(2.2)}$ | $11.7_{(1.9)}$ | 142 | 2.4 |

表 3：ImageNet 验证集的单次裁剪错误率（%）和复杂度比较。此处，MobileNet 指 [13] 中的 "1.0 MobileNet-224"，ShuffleNet 指 [52] 中的 "ShuffleNet 1 × ($g$ = 3)"。

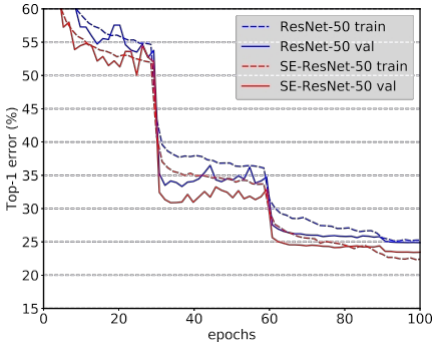



图 4：ResNet-50 和 SE-ResNet-50 在 Im- ageNet 上的训练曲线。

更多网络的曲线见补充配图）。需要指出的是，SE 块本身增加了深度，但它们是以极其节省计算量的方式增加深度的，即使在扩展基础架构深度的收益递减时也能获得良好的回报。此外，我们还发现，在一系列不同深度的训练中，性能的提升是一致的，这表明 SE 区块带来的提升可以与基础架构深度的增加结合使用。

与现代架构的整合。接下来，我们研究了将 SE 块与另两种最先进的架构 Inception-ResNet-v2 [42] 和 ResNeXt（使用 32 × 4d 设置）[47] 结合的效果，这两种架构都在模块中引入了先验结构。

我们构建了与这些网络等价的 SENet，即 SE- Inception-ResNet-v2 和 SE-ResNeXt（SE-ResNeXt-50 的配置见表 1）。表 2 中的结果表明，在这两种架构中引入 SE 块后，性能都有显著提高。特别是，SE-ResNeXt- 50 的前 5 名误差率为 5.49%，既优于其直接对应部分 ResNeXt-50（前 5 名误差率为 5.90%），也优于更深层次的 ResNeXt-101（前 5 名误差率为 5.57%），而后者的参数数量和计算开销最多是前者的两倍。至于 Inception-ResNet- v2 的实验，我们推测裁剪策略的不同可能会导致他们报告的结果与我们重新实施的结果之间存在差距，因为 [42] 中没有明确说明他们的原始图像大小，而我们从相对较大的图像中裁剪了 299 × 299 区域（其中较短的边缘被调整了大小）。

| | 224 × 224 | | 320 × 320 / 299 × 299 | |
|---|---|---|---|---|
| | 第一名 呃。 | 前五名 呃。 | 第一名 呃。 | 前五名 呃。 |
| ResNet-152 [10] | 23.0 | 6.7 | 21.3 | 5.5 |
| ResNet-200 [11] | 21.7 | 5.8 | 20.1 | 4.8 |
| 开端-v3 [44] | - | - | 21.2 | 5.6 |
| 开端-v4 [42] | - | - | 20.0 | 5.0 |
| Inception-ResNet-v2 [42] | - | - | 19.9 | 4.9 |
| ResNeXt-101 (64 × 4d) [47] | 20.4 | 5.3 | 19.1 | 4.4 |
| DenseNet-264 [14] | 22.15 | 6.12 | - | - |
| 注意-92 [46] | - | - | 19.5 | 4.8 |
| 极深聚网络 [51] † | - | - | 18.71 | 4.25 |
| PyramidNet-200 [8] | 20.1 | 5.4 | 19.2 | 4.7 |
| DPN-131 [5] | 19.93 | 5.12 | 18.55 | 4.16 |
| SENet-154 | 18.68 | 4.47 | 17.28 | 3.79 |
| NASNet-A (6@4032) [55]† | - | - | 17.3‡ | 3.8‡ |
| SENet-154（挑战后） | - | - | 16.88‡ | 3.58‡ |

表 4：最先进 CNN 在 Im-ageNet 验证集上的单作物错误率。测试作物的大小为 224 × 224 和
320 × 320 / 299 × 299，如 [11]。† 表示采用更大的裁剪 331 × 331 的模型。‡ 表示挑战后的结果。SENet- 154（挑战后）是用较大的输入尺寸 320 × 320 训练出来的
与输入大小为 224 × 224 的原始图像相比。

到 352）。SE-Inception-ResNet-v2（前五名错误率为 4.79%）比我们重新实现的 Inception-ResNet-v2（前五名错误率为 5.21%）高出 0.42%（相对改进 8.1%），也比 [42] 中报告的结果高出 0.42%。

我们还通过 VGG-16[39] 和 BN-Inception 架构[16]的实验，评估了 SE 区块在*非残差*网络上运行时的效果。由于深度网络的优化非常棘手[16, 39]，为了便于从头开始训练 VGG-16，我们在每次卷积后都添加了一个批量归一化层。我们采用相同的方案训练 SE-VGG-16。比较结果如表 2 所示，与残差架构中出现的现象相同。

最后，我们在表 3 中对 MobileNet [13] 和 ShuffleNet [52] 这两种具有代表性的高效体系结构进行了评估，结果表明 SE 区块能以最小的计算成本持续大幅提高准确性。这些实验表明，SE 块引起的改进可以与多种架构结合使用。此外，这一结果对残差和非残差基础都适用。

| | TOP-1 Er. | 前 5 名错误。 |
|---|---|---|
| 地点-365-CNN [37] | 41.07 | 11.48 |
| ResNet-152 （我们的） | 41.15 | 11.61 |
| SE-ResNet-152 | 40.37 | 11.01 |

表 5：Places365 验证集上的单作物错误率（%）。

ILSVRC 2017 分类竞赛结果。SENets 是我们参加比赛的基础，我们在比赛中获得了第一名。我们的获奖作品由一个小型 SENets 组合组成，该组合采用了标准的多尺度和多作物融合策略，在测试集上获得了 2.251% 的前五名误差。其中一个高性能网络（我们称之为 *SENet-154*）是通过将 SE 块与改进的 ResNeXt

[47]（详情见补充材料），其目标是在不太强调模型复杂性的情况下达到尽可能高的准确率。我们在表 4 中将其与 ImageNet 估值集上表现最好的已发布模型进行了比较。我们的模型在 224 × 224 中心裁剪评估中取得了 18.68% 的前 1 名误差和 4.47% 的前 5 名误差。为了进行公平比较，我们提供了 320 × 320 中心裁剪评估，结果显示与之前的工作相比，性能有了显著提高。比赛结束后，我们使用更大的输入尺寸 320 × 320 训练 SENet-154，在前 1 名（16.88%）和前 5 名（3.58%）误差指标下都取得了较低的误差率。

### 6.2. 场景分类

我们使用 Places365-Challenge 数据集 [53] 进行场景分类实验。该数据集包括 800 万张训练图像和 365 个类别中的 36500 张验证图像。与分类相比，场景理解任务可以更好地评估模型的泛化和抽象能力，因为它需要捕捉更复杂的数据关联，并对更大程度的外观变化具有鲁棒性。

我们使用 ResNet-152 作为评估 SE 区块有效性的强基线，并遵循 [37] 中的训练和评估协议。表 5 显示了 ResNet-152 和 SE-ResNet-152 的结果。具体来说，SE-ResNet- 152（前五名错误率为 11.01%）的验证错误率低于 ResNet-152（前五名错误率为 11.61%），这证明 SE 块在不同数据集上都能表现出色。该 SENet 超过了之前的最先进模型 Places- 365-CNN[37]，后者在该任务中的前 5 名错误率为 11.48%。

### 6.3. COCO 上的物体检测

COCO 数据集包含 8 万张训练图像和 4 万张验证图像。我们使用 Faster R-CNN [33] 作为检测方法，并沿用 [10] 中

| | AP@IoU=0.5 | 美联社 |
|---|---|---|
| ResNet-50 | 45.2 | 25.1 |
| SE-ResNet-50 | 46.8 | 26.4 |
| ResNet-101 | 48.4 | 27.2 |
| SE-ResNet-101 | 49.2 | 27.9 |

的基本实现方法。

表 6：使用基本 Faster R-CNN 在 COCO 40k 验证集上的物体检测结果。

在这里，我们的目的是评估用 SE-ResNet 替换基本架构 ResNet 的好处，以便将改进归功于更好的表示。第 6 页显示了在验证集上分别使用 ResNet-50、ResNet-101 及其 SE 对应的结果。在 COCO 的标准指标 AP 和 AP@IoU=0.5 上，SE-ResNet-50 分别比 ResNet-50 高出 1.3%（相对提高 5.2%）和 1.6%。重要的是，在 AP 指标上，SE 块能使深度架构 ResNet-101 获益 0.7%（相对提高 2.6%）。

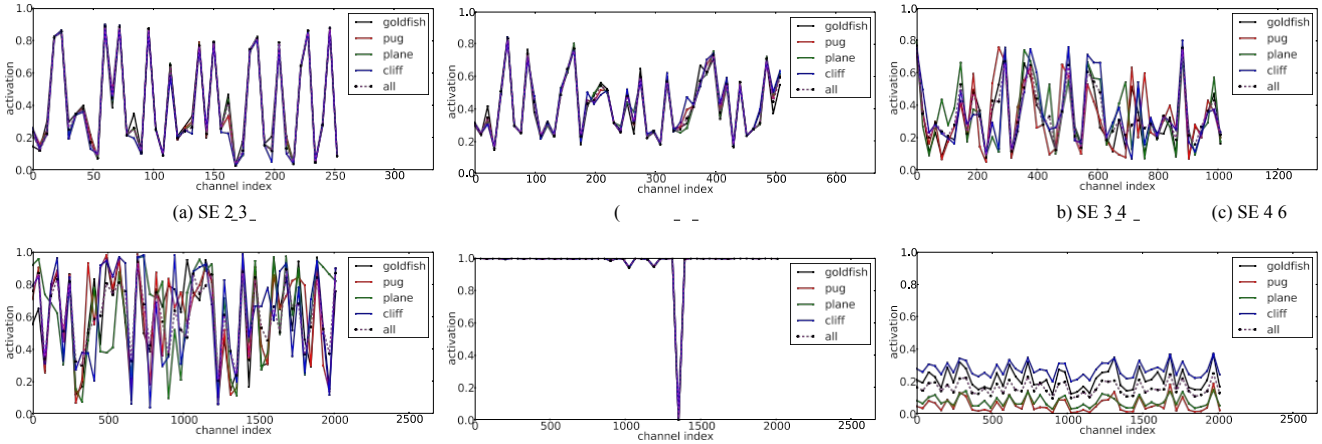
### 6.4. 分析与解释

缩减率。公式 (5) 中引入的缩减率 $r$ 是一个重要的超参数，它允许我们改变模型中 SE 块的容量和计算成本。为了研究这种关系，我们基于 SE-ResNet-50 对一系列不同的 $r$ 值进行了实验。表 7 中的对比显示，性能并没有随着容量的增加而单调提高。这可能是由于 SE 模块过度拟合了训练集的信道相互依赖性。特别是，我们发现设置 $r = 16$ 可以在准确性和复杂性之间取得良好的平衡，因此我们在所有实验中都使用了这个值。

激励的作用。虽然经验表明 SE 区块可以提高网络性能，但我们也希望了解自触发激励机制在实践中是如何运作的。为了更清楚地了解 SE 区块的行为，我们在本节中研究了 SE-ResNet-50 模型中的激活样本，并考察了它们在不同区块中不同类别的分布情况。具体来说，我们从 ImageNet 数据集中抽取了四个类别，这些类别在语义和外观上都表现出不同的特征。

| 比率 $r$ | TOP-1 Er. | 前 5 名错误。 | 百万参数 |
|---|---|---|---|
| 4 | 23.21 | 6.63 | 35.7 |
| 8 | 23.19 | 6.64 | 30.7 |
| 16 | 23.29 | 6.62 | 28.1 |
| 32 | 23.40 | 6.77 | 26.9 |
| 原创 | 24.80 | 7.48 | 25.6 |

表 7：不同缩减率下 SE-ResNet-50 在 ImageNet 验证集上的单作物错误率（%）和参数大小
$r$.这里的*原始数据*指的是 ResNet-50。




(a) SE 2_3_ ( _ _ b) SE 3_4_ _ (c) SE 4 6

(d) SE 5-1- (e) SE 5-2- (f) SE 5-3-

图 5：ImageNet 上 SE-ResNet-50 不同模块中*激发*引起的激活。 模块命名为 "*SE stageID blockID*"。

这些类别包括*金鱼*、*哈巴狗*、 *飞机*和*悬崖*（这些类别的图像示例见补充材料）。然后，我们从验证集中为每个类别抽取 50 个样本，计算每个阶段最后一个 SE 块（即下采样之前）中 50 个均匀采样通道的平均激活度，并将其分布绘制在图 5 中。为便于参考，我们还绘制了所有 1000 个类别的平均激活分布图。

关于 *"激发 "*的作用，我们有以下三点看法。首先，在较低层中，不同类别的分布几乎相同，例如 SE_2_3。这表明，在网络的早期阶段，不同类别可能共享特征通道的 重要性。但有趣的是，第二个观察结果表明，在更深的层级中，每个通道的价值变得更具有类别特异性，因为不同类别对特征的区分价值表现出不同的偏好，例如 SE 4 6 和 SE 5 1。这两项观察结果与之前的研究结果一致[23, 50]，即低层特征通常更具一般性（即在分类中与类别无关），而高层特征则更具特异性。因此，表征学习得益于由 SE 块引起的重新校准，它能在需要的范围内适应性地促进特征提取和特化。最后，我们在网络的最后阶段观察到了一些不同的现象。SE 5 2 显示出一种有趣的饱和状态趋势，即大部分激活值接近 1，其余接近 0。在 SE 5 3 中的网络末端（紧接着是在分类器之前的全局池化先验），不同类别出现了类似的模式，但规模略有变化（可通过"..... "或"..... "调整）。

分类器）。这表明，SE 5 2 和 SE 5 3 在为网络提供重新校准方面的重要性低于之前的区块。这一发现与第 4 章中的实证调查结果显示的结果一致，即通过移除最后阶段的 SE 块，可以显著减少总体参数数量，而性能损失微乎其微。

## 7. 结论

在本文中，我们提出了 SE 块，这是一个新颖的结构单元，旨在通过执行动态信道特征重新校准来提高网络的表征能力。广泛的实验证明了 SENets 的有效性，它在多个数据集上实现了最先进的性能。此外，这些实验还让我们深入了解了以前的架构在模拟信道特征依赖性方面存在的局限性，我们希望这些局限性能被证明对其他需要强分辨特征的任务有用。最后，SE 块引起的特征重要性可能会对压缩网络剪枝等相关领域有所帮助。

致谢。我们要感谢安德鲁-齐瑟曼教授（Professor Andrew Zisserman）的有益评论，以及塞缪尔-阿尔巴尼（Samuel Albanie）的讨论和论文写作编辑。我们还要感谢李超在训练系统中做出的贡献。沈力得到了美国国家情报总监办公室（ODNI）、情报高级研究项目活动（IARPA）的支持，合同号为 2014-14071600010。本文所含观点和结论仅代表作者本人，不应被解释为必然代表 ODNI、IARPA 或美国政府明示或暗示的官方政策或认可。美国政府有权为政府目的再版和分发重印本，但须注明版权。

## 参考资料


[1] S.贝尔、C. L. 齐特尼克、K. 巴拉和 R. 吉尔希克。内外网：利用跳池 和递归神经网络检测上下文中的物体。In *CVPR*, 2016.1

[2] T.Bluche.联合线段分割和转录以实现末端端手写段落识别。In *NIPS*, 2016.2

[3] C.Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang、L.Wang, C. Huang, W. Xu, D. Ramanan, and T. S. Huang.一看二想：用反馈卷积神经网络捕捉自上而下的视觉注意力。In *ICCV*, 2015.2

[4] L.Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.Chua.SCA-CNN：用于图像字幕的卷积网络中的空间和信道注意。In *CVPR*, 2017.2

[5] Y.Chen, J. Li, H. Xiao, X. Jin, S. Yan, and J. Feng.双路径网络。In *NIPS*, 2017.2, 6

[6] F.Chollet.Xception： Deep learning with depthwise separa- ble convolutions.In *CVPR*, 2017.2

[7] J.S. Chung、A. Senior、O. Vinyals 和 A. Zisserman。唇在野外阅读句子。In *CVPR*, 2017.2

[8] D.Han, J. Kim, and J. Kim.Deep pyramidal residual net- works.In *CVPR*, 2017.6

[9] K.He, X. Zhang, S. Ren, and J. Sun.深入研究rec分类器：超越人类水平的 ImageNet 分类性能。In *ICCV*, 2015.5

[10] K.He, X. Zhang, S. Ren, and J. Sun.深度残差学习用于图像识别。In *CVPR*, 2016.2, 5, 6, 7

[11] K.He, X. Zhang, S. Ren, and J. Sun. 深度残差网络中的身份映射。In *ECCV*, 2016.2, 6

[12] S.Hochreiter 和 J. Schmidhuber.长短期记忆。*神经计算*，1997 年。2

[13] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang、T.Weyand, M. Andreetto, and H. Adam.移动网络：用于移动视觉应用的高效卷积神经网络 cations. *arXiv:1704.04861*, 2017.3, 6

[14] G. Huang, Z. Liu, K. Q. Weinberger, and L. Maaten.密集连接的卷积网络。In *CVPR*, 2017.2, 6

[15] Y.Ioannou, D. Robertson, R. Cipolla, and A. Criminisi.深根：利用分层滤波 ter 组提高 CNN 效率。In *CVPR*, 2017.2

[16] S.Ioffe 和 C. Szegedy.批量规范化：加速通过减少内部协变量偏移来进行深度网络训练。在 *ICML*, 2015.1, 2, 5, 6

[17] L.Itti and C. Koch.视觉的计算建模 tention.*自然神经科学评论》*，2001 年。2

[18] L.Itti, C. Koch, and E. Niebur.基于显著性的用于快速场景分析的视觉注意力。*IEEE TPAMI*，1998。2

[19] M. Jaderberg, K. Simonyan, A. Zisserman、和 K.Kavukcuoglu. Spatial transformer networks. 在 *NIPS*, 2015.1, 2

[20] M.Jaderberg, A. Vedaldi, and A. Zisserman.用低级展开加速卷积神经网络。 *BMVC*，2014 年。2

[21] A.Krizhevsky、I. Sutskever 和 G. E. Hinton。图像网络利用深度卷积神经网络进行分类。在 *NIPS*, 2012.1, 3

[22] H.Larochelle and G. E. Hinton.用三阶波尔兹曼机学习结合眼窝瞥视。In *NIPS*、

2010.2

[23] H.Lee, R. Grosse, R. Ranganath, and A. Y. Ng.用于可扩展的无监督分层表征学习的卷积深度信念网络 ing。In *ICML*, 2009.8

[24] M.Lin, Q. Chen, and S. Yan. 网络中的网络。*arXiv:1312.4400*, 2013.2

[25] T.-Y.林Lin、M. Maire、S. Belongie、J. Hays、P. Perona、D. Ra-manan、P. Dollar 和 C. L. Zitnick。Microsoft coco：mon objects in context.*ECCV*, 2014.7

[26] H.Liu, K. Simonyan, O. Vinyals, C. Fernando, and K.Kavukcuoglu.用于高效 架构搜索的层次表示法。*arXiv:1711.00436*, 2017.2

[27] J.Long, E. Shelhamer, and T. Darrell.完全卷积网络进行语义分割。*CVPR*，2015 年。1

[28] A.Miech, I. Laptev, and J. Sivic.用于视频分类的可学习池与上下文门控。*ArXiv:1706.06905*, 2017.2

[29] V.Mnih, N. Heess, A. Graves, and K. Kavukcuoglu.循环视觉注意力的租金模型。In *NIPS*, 2014.2

[30] V.Nair 和 G. E. Hinton. 整流线性单元改进再 stricted boltzmann machines.*ICML*, 2010.3

[31] A.Newell, K. Yang, and J. Deng.堆叠沙漏网用于人体姿态估计。*ECCV*, 2016.1, 2

[32] B.A. Olshausen, C. H. Anderson, and D. C. V. Essen.基于信息动态路由的视觉注意力和不变模式识别的神经生物学模型。*神经科学杂志》*，1993 年。2

[33] S.Ren, K. He, R. Girshick, and J. Sun.更快的 R-CNN：使用区域建议网络进行实时物体检测 works.In *NIPS*, 2015.1, 7

[34] O.O. Russakovsky、J. Deng、H. Su、J. Krause、S. Satheesh、S.Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein、A.C. Berg 和 L. Fei-Fei.ImageNet 大规模视觉 识别挑战。*IJCV*，2015。2

[35] J.Sanchez, F. Perronnin, T. Mensink, and J. Verbeek.Im-使用渔夫向量进行年龄分类：理论与实践。*RR-8209，英瑞亚*，2013 年。3

[36] L.Shen, Z. Lin, and Q. Huang.有效学习深度卷积神经网络的中继反向传播。*ECCV*, 2016.4

[37] L.Shen, Z. Lin, G. Sun, and J. Hu.地点 401 和地点 365 模型。https://github.com/lishen-shirley/Places2-CNNs，2016 年。7

[38] L.Shen, G. Sun, Q. Huang, S. Wang, Z. Lin, and E. Wu. 多层次判别字典学习在大规模图像分类中的应用 tion。*IEEE TIP*, 2015.3

[39] K.Simonyan 和 A. Zisserman.深度卷积网络进行大规模图像识别。2015年，*ICLR*。2, 3, 5, 6

[40] R.K. Srivastava、K. Greff 和 J. Schmidhuber。培训深度网络。In *NIPS*, 2015.2

[41] M.F. Stollenga, J. Masci, F. Gomez, and J. Schmidhuber.通过反馈连接实现内部选择性注意的深度网络 back connections.In *NIPS*, 2014.2

[42] C.Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi.开始v4、inception-resnet 和残余连接 对学习的影响。*国际语言资源研讨会*，2016年。2, 3, 4, 5, 6

[43] C.Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed、D.Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 更深层次的卷积In *CVPR*, 2015.1, 2, 4

[44] C.Szegedy、V. Vanhoucke、S. Ioffe、J. Shlens 和 Z. Wojna。重新思考计算机视觉的初始架构。在

*CVPR*, 2016.2, 6

[45] A.Toshev 和 C. Szegedy.DeepPose: Human pose estima- tion via deep neural networks.In *CVPR*, 2014.1

[46] F.Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang、X.Wang, and X. Tang.用于图像 分类的残差注意网络。In *CVPR*, 2017.2, 6

[47] S.Xie, R. Girshick, P. Dollar, Z. Tu, and K. He.汇总深度神经网络的残差变换。In *CVPR*, 2017.2, 3, 5, 6, 7

[48] K.Xu、J. Ba、R. Kiros、K. Cho、A. Courville、R. Salakhudi-Nov、R. Zemel 和 Y. Bengio。展示、参加和讲述：神经图像标题生成与视觉注意力。In *ICML*, 2015.2

[49] J.Yang, K. Yu, Y. Gong, and T. Huang.线性空间吡利用稀疏编码进行图像分类的中间匹配。2009 年，*CVPR*。3

[50] J.Yosinski, J. Clune, Y. Bengio, and H. Lipson.深度神经网络中的特征有多易变？In *NIPS*, 2014.8

[51] X.Zhang, Z. Li, C. C. Loy, and D. Lin.Polynet：追求深度网络的结构多样性。In *CVPR*, 2017.6

[52] X.Zhang, X. Zhou, M. Lin, and J. Sun.Shufflenet: An extremely efficient convolutional neural network for mobile devices. *arXiv:1707.01083*, 2017.3, 6

[53] B.Zhou、A. Lapedriza、A. Khosla、A. Oliva 和 A. Torralba。地点用于场景识别的千万张图像数据库*IEEE TPAMI*, 2017.7

[54] B.Zoph 和 Q. V. Le.神经架构搜索与强化学习（rein- forcement learning）。In *ICLR*, 2017.2

[55] B.Zoph, V. Vasudevan, J. Shlens, and Q. V. Le.学习为可扩展的图像识别提供可转移的架构。*arXiv: 1707.07012*, 2017.2, 6