# ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks

# ECA-Net：深度卷积神经网络的高效通道注意力

Qilong Wang , Banggu Wu , Pengfei Zhu , Peihua Li , Wangmeng Zuo , Qinghua Hu

王启龙 ，吴邦国 ，朱鹏飞 ，李培华 ，左旺盟 ，胡清华

Tianjin Key Lab of Machine Learning, College of Intelligence and Computing, Tianjin University, China

天津大学智能与计算学院，天津市机器学习重点实验室，中国

Dalian University of Technology, China Harbin Institute of Technology, China

大连理工大学，中国 哈尔滨工业大学，中国

Abstract

摘要

Recently, channel attention mechanism has demonstrated to offer great potential in improving the performance of deep convolutional neural networks (CNNs). However, most existing methods dedicate to developing more sophisticated attention modules for achieving better performance, which inevitably increase model complexity. To overcome the paradox of performance and complexity trade-off, this paper proposes an Efficient Channel Attention (ECA) module, which only involves a handful of parameters while bringing clear performance gain. By dissecting the channel attention module in SENet, we empirically show avoiding dimensionality reduction is important for learning channel attention, and appropriate cross-channel interaction can preserve performance while significantly decreasing model complexity. Therefore, we propose a local cross-channel interaction strategy without dimensionality reduction, which can be efficiently implemented via 1D convolution. Furthermore, we develop a method to adaptively select kernel size of convolution, determining coverage of local cross-channel interaction. The proposed ECA module is efficient yet effective, e.g., the parameters and computations of our modules against backbone of ResNet50 are 80 vs. 24.37M and 4.7e-4 GFLOPs vs. 3.86 GFLOPs, respectively, and the performance boost is more than in terms of Top-1 accuracy. We extensively evaluate our ECA module on image classification, object detection and instance segmentation with backbones of ResNets and MobileNetV2. The experimental results show our module is more efficient while performing favorably against its counterparts.

最近，通道注意力机制在提高深度卷积神经网络（CNNs）性能方面显示出巨大的潜力。然而，现有的大部分方法都致力于开发更复杂的注意力模块以实现更好的性能，这不可避免地增加了模型的复杂性。为了克服性能和复杂度权衡之间的矛盾，本文提出了一个高效的通道注意力（ECA）模块，该模块仅涉及少数参数，却能带来明显的性能提升。通过对SENet中的通道注意力模块进行剖析，我们从经验上表明避免维度降低对于学习通道注意力很重要，并且适当的跨通道交互可以在显著降低模型复杂度的同时保持性能。因此，我们提出了一种无需维度降低的局部跨通道交互策略，可以通过1D卷积高效实现。此外，我们开发了一种方法来自适应选择 卷积的核大小，确定局部跨通道交互的覆盖范围。所提出的ECA模块既高效又有效，例如，我们的模块与ResNet50基干的参数和计算量分别为80 vs. 24.37M和4.7e-4 GFLOPs vs. 3.86 GFLOPs，性能提升超过 的Top-1准确度。我们在图像分类、目标检测和实例分割任务上，使用ResNets和MobileNetV2作为基干，对ECA模块进行了广泛评估。实验结果表明，我们的模块在保持与现有方法相当性能的同时，具有更高的效率。

# 1. Introduction

# 1. 引言

Deep convolutional neural networks (CNNs) have been widely used in computer vision community, and have achieved great progress in a broad range of tasks, e.g., image classification, object detection and semantic segmentation. Starting from the groundbreaking AlexNet [17], many researches are continuously investigated to further improve the performance of deep CNNs [29, 30, 11, 15, 19, 20, 32]. Recently, incorporation of channel attention into convolution blocks has attracted a lot of interests, showing great potential in performance improvement [14, 33, 13, 4, 9, 18, 7]. One of the representative methods is squeeze-and-excitation networks (SENet) [14], which learns channel attention for each convolution block, bringing clear performance gain for various deep CNN architectures.

深度卷积神经网络（CNNs）在计算机视觉领域被广泛应用，并在广泛的任务中取得了巨大进步，例如图像分类、目标检测和语义分割。从开创性的AlexNet [17] 开始，许多研究不断深入，以进一步提高深度CNNs的性能 [29, 30, 11, 15, 19, 20, 32]。最近，将通道注意力融入卷积块中受到了很多关注，显示出在性能提升上的巨大潜力 [14, 33, 13, 4, 9, 18, 7]。其中一种具有代表性的方法是挤压-激励网络（SENet）[14]，它为每个卷积块学习通道注意力，为各种深度CNN架构带来了明显的性能提升。

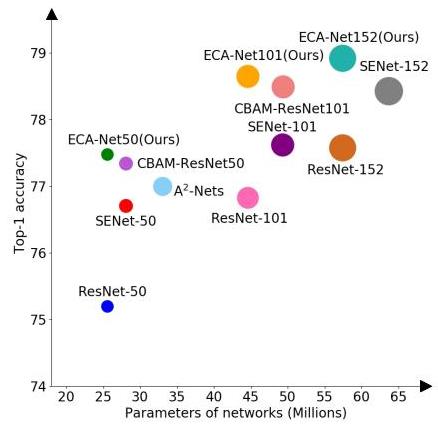


Figure 1. Comparison of various attention modules (i.e., SENet [14], CBAM [33], -Nets [4] and ECA-Net) using ResNets [11] as backbone models in terms of classification accuracy, network parameters and FLOPs, indicated by radiuses of circles. Note that our ECA-Net obtains higher accuracy while having less model complexity.

图1. 使用ResNets [11] 作为基础模型，在分类准确度、网络参数和FLOPs方面，比较了各种注意力模块（即SENet [14]、CBAM [33]、 -Nets [4] 和ECA-Net）的性能，圆圈半径表示。注意，我们的ECA-Net在具有较低模型复杂度的同时获得了更高的准确度。

Following the setting of squeeze (i.e., feature aggregation) and excitation (i.e., feature recalibration) in SENet [14], some researches improve SE block by capturing more sophisticated channel-wise dependencies [33, 4, 9, 7] or by combining with additional spatial attention [33, 13, 7]. Although these methods have achieved higher accuracy, they often bring higher model complexity and suffer from heavier computational burden. Different from the aforementioned methods that achieve better performance at the cost of higher model complexity, this paper focuses instead on a question: Can one learn effective channel attention in a more efficient way?

遵循SENet [14]中设置的挤压（即特征聚合）和激励（即特征重校准），一些研究通过捕获更复杂的通道依赖关系 [33, 4, 9, 7] 或结合额外的空间注意力 [33, 13, 7] 来改进SE块。尽管这些方法取得了更高的准确度，但它们常常带来更高的模型复杂度，并承受更重的计算负担。与上述以更高模型复杂度为代价获得更好性能的方法不同，本文专注于一个问题：是否可以以一种更高效的方式学习有效的通道注意力？

\* Qinghua Hu is the corresponding author.

\* 郭清华是通讯作者。

Email: {qlwang, wubanggu, huqinghua }@tju.edu.cn. The work was supported by the National Natural Science Foundation of China (Grant No. 61806140, 61876127, 61925602, 61971086, U19A2073, 61732011), Major Scientific Research Project of Zhejiang Lab (2019DB0ZX01). Q. Wang was supported by National Postdoctoral Program for Innovative Talents.

电子邮件: {qlwang, wubanggu, huqinghua }@tju.edu.cn。本研究得到国家自然科学基金（项目编号：61806140, 61876127, 61925602, 61971086, U19A2073, 61732011）、浙江省实验室重大科研项目（2019DB0ZX01）的支持。Q. Wang 得到了国家博士后创新人才计划的支持。

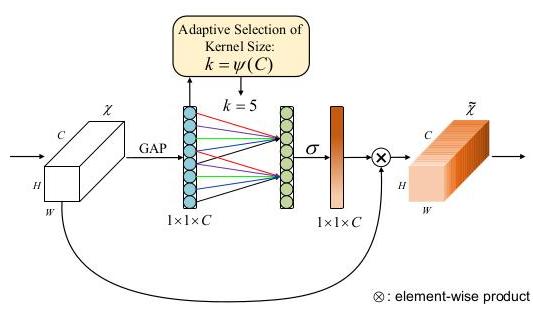


Figure 2. Diagram of our efficient channel attention (ECA) module. Given the aggregated features obtained by global average pooling (GAP), ECA generates channel weights by performing a fast convolution of size , where is adaptively determined via a mapping of channel dimension .

图 2。我们高效通道注意力（ECA）模块的示意图。在获得全局平均池化（GAP）得到的聚合特征后，ECA 通过执行大小为 的快速 卷积来生成通道权重，其中 是通过通道维度的映射自适应确定的 。

To answer this question, we first revisit the channel attention module in SENet. Specifically, given the input features, SE block first employs a global average pooling for each channel independently, then two fully-connected (FC) layers with non-linearity followed by a Sigmoid function are used to generate channel weights. The two FC layers are designed to capture non-linear cross-channel interaction, which involve dimensionality reduction for controlling model complexity. Although this strategy is widely used in subsequent channel attention modules [33, 13, 9], our empirical studies show dimensionality reduction brings side effect on channel attention prediction, and it is inefficient and unnecessary to capture dependencies across all channels.

为了回答这个问题，我们首先重新审视了 SENet 中的通道注意力模块。具体来说，SE 块首先对每个通道独立地应用全局平均池化，然后使用两个带有非线性激活函数的全连接（FC）层，随后是一个 Sigmoid 函数来生成通道权重。这两个 FC 层被设计来捕获非线性跨通道交互，这涉及到维数降低以控制模型复杂度。尽管这种策略在后续的通道注意力模块中得到了广泛应用 [33, 13, 9]，但我们的实证研究表明维数降低对通道注意力预测有副作用，而且捕获所有通道之间的依赖关系既低效又没有必要。

Therefore, this paper proposes an Efficient Channel Attention (ECA) module for deep CNNs, which avoids dimensionality reduction and captures cross-channel interaction in an efficient way. As illustrated in Figure 2 after channel-wise global average pooling without dimensionality reduction, our ECA captures local cross-channel interaction by considering every channel and its neighbors. Such method is proven to guarantee both efficiency and effectiveness. Note that our ECA can be efficiently implemented by fast convolution of size , where kernel size represents the coverage of local cross-channel interaction, i.e., how many neighbors participate in attention prediction of one channel. To avoid manual tuning of via cross-validation, we develop a method to adaptively determine , where coverage of interaction (i.e., kernel size ) is proportional to channel dimension. As shown in Figure 1 and Table 3, as opposed to the backbone models [11], deep CNNs with our ECA module (called ECA-Net) introduce very few additional parameters and negligible computations, while bringing notable performance gain. For example, for ResNet-50 with parameters and GFLOPs, the additional parameters and computations of ECA-Net50 are 80 and 4.7e- 4 GFLOPs, respectively; meanwhile, ECA-Net50 outperforms ResNet-50 by in terms of Top-1 accuracy.

因此，本文为深度卷积神经网络（CNN）提出了一种高效通道注意力（ECA）模块，该模块避免了维度降低，并以高效的方式捕捉跨通道交互。如图2所示，在通道-wise全局平均池化后未进行维度降低的情况下，我们的ECA通过考虑每个通道及其 邻居来捕捉局部跨通道交互。这种方法被证明既保证了效率又保证了有效性。注意，我们的ECA可以通过大小为 的快速 卷积高效实现，其中卷积核大小 表示局部跨通道交互的覆盖范围，即有多少邻居参与一个通道的注意力预测。为了避免通过交叉验证手动调整 ，我们开发了一种自适应确定 的方法，其中交互的覆盖范围（即卷积核大小 ）与通道维度成比例。如图1和表3所示，与基础模型 [11] 相比，使用我们ECA模块的深度CNN（称为ECA-Net）引入的额外参数和可忽略的计算量很少，同时带来了显著的性能提升。例如，对于具有 参数和 GFLOPs 的 ResNet-50，ECA-Net50 的额外参数和计算量分别为80和4.7e-4 GFLOPs；同时，ECA-Net50 在Top-1 准确性方面比 ResNet-50 高出 。

| Model | No DR | Cross-channel Interaction | Lightweight |
| --- | --- | --- | --- |
| SENet [14] CBAM [33] GE- [13] GE- [13] GE- [13] -Net [4] GSoP-Net [9] ECA-Net (Ours) | × |  |  |
| X |  |  |
|  |  |  |
|  | × |  |
| X |  |  |
| X |  |  |
| X |  |  |
|  |  |  |

Table 1. Comparison of existing attention modules in terms of whether no channel dimensionality reduction (No DR), cross-channel interaction and less parameters than SE (indicated by lightweight) or not.

表1. 按是否无通道维度降低（No DR）、是否存在跨通道交互以及是否比SE（由轻量级表示）参数更少来比较现有注意力模块。

Table 1 summarizes existing attention modules in terms of whether channel dimensionality reduction (DR), cross-channel interaction and lightweight model, where we can see that our ECA module learn effective channel attention by avoiding channel dimensionality reduction while capturing cross-channel interaction in an extremely lightweight way. To evaluate our method, we conduct experiments on ImageNet-1K [6] and MS COCO [23] in a variety of tasks using different deep CNN architectures.

表1总结了现有的注意力模块，包括是否进行了通道维度降低（DR）、跨通道交互以及轻量级模型，我们可以看到我们的ECA模块通过避免通道维度降低并极其轻量地捕获跨通道交互来学习有效的通道注意力。为了评估我们的方法，我们在ImageNet-1K [6] 和 MS COCO [23] 上使用不同的深度CNN架构进行多种任务的实验。

The contributions of this paper are summarized as follows. (1) We dissect the SE block and empirically demonstrate avoiding dimensionality reduction and appropriate cross-channel interaction are important to learn effective and efficient channel attention, respectively. (2) Based on above analysis, we make an attempt to develop an extremely lightweight channel attention module for deep CNNs by proposing an Efficient Channel Attention (ECA), which increases little model complexity while bringing clear improvement. (3) The experimental results on ImageNet-1K and MS COCO demonstrate our method has lower model complexity than state-of-the-arts while achieving very competitive performance.

本文的贡献可以概括如下：（1）我们剖析了SE块，并通过实验证明了避免维度降低和适当的跨通道交互分别是学习有效和高效通道注意力的关键；（2）基于上述分析，我们尝试提出了一个极其轻量级的通道注意力模块（ECA），用于深度CNN，它在增加很少的模型复杂性的同时带来了明显的改进；（3）在ImageNet-1K和MS COCO上的实验结果表明，我们的方法在达到非常有竞争力的性能的同时，模型复杂度低于现有技术水平。

# 2. Related Work

# 2. 相关工作

Attention mechanism has proven to be a potential means to enhance deep CNNs. SE-Net [14] presents for the first time an effective mechanism to learn channel attention and achieves promising performance. Subsequently, development of attention modules can be roughly divided into two directions: (1) enhancement of feature aggregation; (2) combination of channel and spatial attentions. Specifically, CBAM [33] employs both average and max pooling to aggregate features. GSoP [9] introduces a second-order pooling for more effective feature aggregation. GE [13] explores spatial extension using a depth-wise convolution [5] to aggregate features. CBAM [33] and scSE [27] compute spatial attention using a convolution of kernel size , then combine it with channel attention. Sharing similar philosophy with Non-Local (NL) neural networks [32], GC-Net [2] develops a simplified NL network and integrates with the SE block, resulting in a lightweight module to model long-range dependency. Double Attention Networks [4] introduces a novel relation function for NL blocks for image or video recognition. Dual Attention Network (DAN) [7] simultaneously considers NL-based channel and spatial attentions for semantic segmentation. However, most above NL-based attention modules can only be used in a single or a few convolution blocks due to their high model complexity. Obviously, all of the above methods focus on developing sophisticated attention modules for better performance. Different from them, our ECA aims at learning effective channel attention with low model complexity.

注意力机制已被证明是提高深度 CNN 的潜在方法。SE-Net [14] 首次提出了一个有效的机制来学习通道注意力并取得了有希望的性能。随后，注意力模块的发展可以大致分为两个方向：（1）增强特征聚合；（2）通道和空间注意力的结合。具体来说，CBAM [33] 使用平均和最大池化来聚合特征。GSoP [9] 引入了二阶池化以更有效地聚合特征。GE [13] 探索使用深度卷积 [5] 进行空间扩展以聚合特征。CBAM [33] 和 scSE [27] 使用 卷积的核大小 计算空间注意力，然后将其与通道注意力结合。GC-Net [2] 与非局部（NL）神经网络 [32] 有相似的哲学，开发了一个简化的 NL 网络，并与 SE 块集成，形成了一个轻量级模块来建模长距离依赖。双重注意力网络 [4] 为图像或视频识别的 NL 块引入了一种新颖的关系函数。双注意力网络（DAN）[7] 同时考虑基于 NL 的通道和空间注意力进行语义分割。然而，上述大多数基于 NL 的注意力模块由于其模型复杂性高，只能用于单个或少数卷积块中。显然，以上所有方法都专注于开发复杂的注意力模块以获得更好的性能。与它们不同，我们的 ECA 旨在学习具有低模型复杂度的有效通道注意力。

Our work is also related to efficient convolutions, which are designed for lightweight CNNs. Two widely used efficient convolutions are group convolutions [37, 34, 16] and depth-wise separable convolutions [5, 28, 38, 24]. As given in Table 2, although these efficient convolutions involve less parameters, they show little effectiveness in attention module. Our ECA module aims at capturing local cross-channel interaction, which shares some similarities with channel local convolutions [36] and channel-wise convolutions [8]; different from them, our method investigates a convolution with adaptive kernel size to replace FC layers in channel attention module. Comparing with group and depthwise separable convolutions, our method achieves better performance with lower model complexity.

我们的工作也与高效卷积相关，这种卷积是为轻量级CNN设计的。两种广泛使用的高效卷积是分组卷积 [37, 34, 16] 和深度可分离卷积 [5, 28, 38, 24]。如表2所示，尽管这些高效卷积涉及的参数较少，但它们在注意力模块中显示出较小的有效性。我们的ECA模块旨在捕捉局部跨通道交互，这与通道局部卷积 [36] 和通道卷积 [8] 有一些相似之处；不同于它们的是，我们的方法研究了一种 卷积，使用自适应核大小来替换通道注意力模块中的全连接层。与分组卷积和深度可分离卷积相比，我们的方法在较低的模型复杂度下实现了更好的性能。

# 3. Proposed Method

# 3. 提出方法

In this section, we first revisit the channel attention module in SENet [14] (i.e., SE block). Then, we make a empirical diagnosis of SE block by analyzing effects of dimensionality reduction and cross-channel interaction. This motivates us to propose our ECA module. In addition, we develop a method to adaptively determine parameter of our ECA, and finally show how to adopt it for deep CNNs.

在这一节中，我们首先重新审视了SENet [14] 中的通道注意力模块（即SE块）。然后，我们通过分析维度降低和跨通道交互的效果对SE块进行了实证诊断。这激发了我们提出ECA模块。此外，我们开发了一种方法来自适应地确定ECA的参数，并最终展示了如何将其应用于深度CNN。

# 3.1. Revisiting Channel Attention in SE Block

# 3.1. 重新审视SE块中的通道注意力

Let the output of one convolution block be , where and are width, height and channel dimension (i.e., number of filters). Accordingly, the weights of channels in SE block can be computed as

设一个卷积块的输出为 ，其中 和 分别是宽度、高度和通道维度（即滤波器数量）。相应地，SE块中通道的权重可以计算为

where is channel-wise global average pooling (GAP) and is a Sigmoid function. Let

其中 是通道全局平均池化（GAP）， 是Sigmoid函数。让

| Methods | Attention | #.Param. | Top-1 | Top-5 |
| --- | --- | --- | --- | --- |
| Vanilla | N/A | 0 | 75.20 | 92.25 |
| SE |  |  | 76.71 | 93.38 |
| SE-Var1 |  | 0 | 76.00 | 92.90 |
| SE-Var2 |  |  | 77.07 | 93.31 |
| SE-Var3 |  |  | 77.42 | 93.64 |
| SE-GC1 |  |  | 76.95 | 93.47 |
| SE-GC2 |  |  | 76.98 | 93.31 |
| SE-GC3 |  |  | 76.96 | 93.38 |
| ECA-NS | with Eq.(7) |  | 77.35 | 93.61 |
| ECA (Ours) |  |  | 77.43 | 93.65 |

Table 2. Comparison of various channel attention modules using ResNet-50 as backbone model on ImageNet. #.Param. indicates number of parameters of the channel attention module; indicates element-wise product; GC and C1D indicate group convolutions and convolution, respectively; is kernel size of C1D.

表2. 使用ResNet-50作为基础模型在ImageNet上比较各种通道注意力模块。#.Param.表示通道注意力模块的参数数量； 表示逐元素乘积；GC和C1D分别表示组卷积和 卷积； 是C1D的核大小。

takes the form

的形式为

where ReLU indicates the Rectified Linear Unit [25]. To avoid high model complexity, sizes of and are set to and , respectively. We can see that involves all parameters of channel attention block. While dimensionality reduction in Eq. (2) can reduce model complexity, it destroys the direct correspondence between channel and its weight. For example, one single FC layer predicts weight of each channel using a linear combination of all channels. But Eq. (2) first projects channel features into a low-dimensional space and then maps them back, making correspondence between channel and its weight be indirect.

其中ReLU表示修正线性单元[25]。为了避免高模型复杂性， 和 的大小分别设置为 和 。我们可以看到 包含了通道注意力块的所有参数。虽然方程（2）中的维度降低可以减少模型复杂性，但它破坏了通道与其权重之间的直接对应关系。例如，一个单独的全连接层通过所有通道的线性组合预测每个通道的权重。但是方程（2）首先将通道特征投影到低维空间，然后将其映射回来，使得通道与其权重之间的对应关系变得间接。

# 3.2. Efficient Channel Attention (ECA) Module

# 3.2. 高效通道注意力（ECA）模块

After revisiting SE block, we conduct empirical comparisons for analyzing effects of channel dimensionality reduction and cross-channel interaction on channel attention learning. According to these analyses, we propose our efficient channel attention (ECA) module.

在重新审视SE块之后，我们进行了实证比较，以分析通道维度降低和跨通道交互对通道注意力学习的影响。根据这些分析，我们提出了我们的高效通道注意力（ECA）模块。

# 3.2.1 Avoiding Dimensionality Reduction

# 3.2.1 避免维度降低

As discussed above, dimensionality reduction in Eq. (2) makes correspondence between channel and its weight be indirect. To verify its effect, we compare the original SE block with its three variants (i.e., SE-Var1, SE-Var2 and SE-Var3), all of which do not perform dimensionality reduction. As presented in Table 2, SE-Var1 with no parameter is still superior to the original network, indicating channel attention has ability to improve performance of deep CNNs. Meanwhile, SE-Var2 learns the weight of each channel independently, which is slightly superior to SE block while involving less parameters. It may suggest that channel and its weight needs a direct correspondence while avoiding dimensionality reduction is more important than consideration of nonlinear channel dependencies. Additionally, SE-Var3 employing one single FC layer performs better than two FC layers with dimensionality reduction in SE block. All of above results clearly demonstrate avoiding dimensionality reduction is helpful to learn effective channel attention. Therefore, we develop our ECA module without channel dimensionality reduction.

如上所述，式（2）中的维度降低使得通道与其权重之间的对应关系变得间接。为了验证其效果，我们比较了原始SE块与其三个变种（即SE-Var1、SE-Var2和SE-Var3），它们均未执行维度降低。如表2所示，没有参数的SE-Var1仍然优于原始网络，表明通道注意力有能力提升深度CNN的性能。同时，SE-Var2独立地学习每个通道的权重，其性能略优于SE块，同时涉及的参数更少。这可能表明，在避免维度降低的同时，通道及其权重需要直接对应比考虑非线性通道依赖性更为重要。另外，采用单个全连接层（FC层）的SE-Var3比SE块中的两个带维度降低的FC层表现更好。以上所有结果清楚地表明，避免维度降低有助于学习有效的通道注意力。因此，我们开发了不进行通道维度降低的ECA模块。

# 3.2.2 Local Cross-Channel Interaction

# 3.2.2 局部跨通道交互

Given the aggregated feature without dimensionality reduction, channel attention can be learned by

给定未进行维度降低的聚合特征 ，可以通过以下方式学习通道注意力

where is a parameter matrix. In particular, for SE-Var2 and SE-Var3 we have

其中 是一个 参数矩阵。特别是对于SE-Var2和SE-Var3，我们有

where for SE-Var2 is a diagonal matrix, involving parameters; for SE-Var3 is a full matrix, involving parameters. As shown in Eq. (4), the key difference is that SE-Var3 considers cross-channel interaction while SE-Var2 does not, and consequently SE-Var3 achieves better performance. This result indicates that cross-channel interaction is beneficial to learn channel attention. However, SE-Var3 requires a mass of parameters, leading to high model complexity, especially for large channel numbers.

其中SE-Var2的 是一个对角矩阵，包含 参数；SE-Var3的 是一个满矩阵，包含 参数。如式（4）所示，关键的区别在于SE-Var3考虑了跨通道交互，而SE-Var2没有，因此SE-Var3实现了更好的性能。这一结果表明，跨通道交互有助于学习通道注意力。然而，SE-Var3需要大量的参数，导致模型复杂度高，特别是在通道数量大时。

A possible compromise between SE-Var2 and SE-Var3 is extension of to a block diagonal matrix, i.e.,

SE-Var2 和 SE-Var3 之间可能的妥协是将 扩展为块对角矩阵，即，

where Eq. (5) divides channel into groups each of which includes channels, and learns channel attention in each group independently, which captures cross-channel interaction in a local manner. Accordingly, it involves parameters. From perspective of convolution, SE-Var2, SE-Var3 and Eq. (5) can be regarded as a depth-wise separable convolution, a FC layer and group convolutions, respectively. Here, SE block with group convolutions (SE-GC) is indicated by . However, as shown in [24], excessive group convolutions will increase memory access cost and so decrease computational efficiency. Furthermore, as shown in Table 2, SE-GC with varying groups bring no gain over SE-Var2, indicating it is not an effective scheme to capture local cross-channel interaction. The reason may be that SE-GC completely discards dependences among different groups.

其中，等式 (5) 将通道分为 组，每组包含 个通道，并独立地在每个组中学习通道注意力，这以局部方式捕获了跨通道的交互。相应地，它涉及 个参数。从卷积的角度来看，SE-Var2、SE-Var3 和等式 (5) 可以分别被视为深度可分卷积、全连接层和组卷积。这里，带有组卷积的 SE 块（SE-GC）由 表示。然而，如文献 [24] 所示，过多的组卷积会增加内存访问成本，从而降低计算效率。此外，如表 2 所示，使用不同组数的 SE-GC 相比 SE-Var2 没有任何收益，表明捕获局部跨通道交互并不是一个有效方案。原因可能是 SE-GC 完全丢弃了不同组之间的依赖关系。

In this paper, we explore another method to capture local cross-channel interaction, aiming at guaranteeing both efficiency and effectiveness. Specifically, we employ a band matrix to learn channel attention, and has

在本文中，我们探索了另一种捕获局部跨通道交互的方法，旨在保证效率和有效性。具体来说，我们使用带状矩阵 来学习通道注意力，并且 具有

Clearly, in Eq. (6) involves parameters, which is usually less than those of Eq. (5). Furthermore, Eq. (6) avoids complete independence among different groups in Eq. (5). As compared in Table 2, the method in Eq. (6) (namely ECA-NS) outperforms SE-GC of Eq. (5). As for Eq. (6), the weight of is calculated by only considering interaction between and its neighbors, i.e.,

显然，等式 (6) 中的 涉及 个参数，这通常小于等式 (5) 中的参数数量。此外，等式 (6) 避免了等式 (5) 中不同组之间的完全独立。如表 2 所示，等式 (6) 中的方法（即 ECA-NS）优于等式 (5) 的 SE-GC。对于等式 (6)， 的权重仅通过考虑 与其 邻居之间的交互来计算，即，

where indicates the set of adjacent channels of .

其中 表示 相邻通道的集合 。

A more efficient way is to make all channels share the same learning parameters, i.e.,

一种更有效的方法是让所有通道共享相同的学习参数，即，

Note that such strategy can be readily implemented by a fast convolution with kernel size of , i.e.,

注意，这种策略可以通过快速的 卷积轻松实现，其卷积核大小为 ，即，

where C1D indicates convolution. Here, the method in Eq. (9) is called by efficient channel attention (ECA) module, which only involves parameters. As presented in Table 2, our ECA module with achieves similar results with SE-var3 while having much lower model complexity, which guarantees both efficiency and effectiveness by appropriately capturing local cross-channel interaction.

其中 C1D 表示 卷积。在这里，方程 (9) 中的方法被高效通道注意力（ECA）模块调用，该模块仅涉及 参数。如表 2 所示，我们的 ECA 模块与 达到了与 SE-var3 相似的结果，同时具有更低的模型复杂度，通过适当捕获局部跨通道交互，保证了效率和有效性。

# 3.2.3 Coverage of Local Cross-Channel Interaction

# 3.2.3 局部跨通道交互的覆盖范围

Since our ECA module (9) aims at appropriately capturing local cross-channel interaction, so the coverage of interaction (i.e., kernel size of convolution) needs to be determined. The optimized coverage of interaction could be tuned manually for convolution blocks with different channel numbers in various CNN architectures. However, manual tuning via cross-validation will cost a lot of computing resources. Group convolutions have been successfully adopted to improve CNN architectures [37, 34, 16], where high-dimensional (low-dimensional) channels involve long range (short range) convolutions given the fixed number of groups. Sharing the similar philosophy, it is reasonable that the coverage of interaction (i.e., kernel size of convolution) is proportional to channel dimension . In other words, there may exist a mapping between and :

由于我们的 ECA 模块（9）旨在适当捕获局部跨通道交互，因此交互的覆盖范围（即， 的 卷积核大小）需要确定。对于不同卷积神经网络架构中具有不同通道数的卷积块，可以手动调整交互的最佳覆盖范围。然而，通过交叉验证进行手动调整将消耗大量计算资源。分组卷积已成功应用于改进 CNN 架构 [37, 34, 16]，在高维（低维）通道涉及长距离（短距离）卷积的情况下，给定固定数量的组。基于类似的哲学，交互的覆盖范围（即， 的 卷积核大小）与通道维度 成比例是合理的。换句话说，可能存在一个从 到 的映射 ：

def EfficientChannelAttention :

# : input features with shape

# gamma, b: parameters of mapping function

. size ()

if else

avg\_pool nn. AdaptiveAvgPool2d (1)

conv nn.Conv1d , kernel\_size=k, padding=int ,

bias=False)

sigmoid nn. Sigmoid ()

. squeeze . transpose )

. transpose . unsqueeze

return .expand\_as

Figure 3. PyTorch code of our ECA module.

The simplest mapping is a linear function, i.e., . However, the relations characterized by linear function are too limited. On the other hand, it is well known that channel dimension (i.e., number of filters) usually is set to power of 2 . Therefore, we introduce a possible solution by extending the linear function to a non-linear one, i.e.,

最简单的映射是一个线性函数，即 。然而，由线性函数表征的关系过于受限。另一方面，众所周知，通道维度 （即滤波器的数量）通常设置为2的幂。因此，我们通过将线性函数 扩展为非线性函数，提出了一种可能的解决方案，即，

Then, given channel dimension , kernel size can be adaptively determined by

然后，给定通道维度 ，卷积核大小 可以通过以下方式自适应确定：

where indicates the nearest odd number of . In this paper, we set and to 2 and 1 throughout all the experiments, respectively. Clearly, through the mapping , high-dimensional channels have longer range interaction while low-dimensional ones undergo shorter range interaction by using a non-linear mapping.

其中 表示 的最接近的奇数。在本文中，我们设置 和 在所有实验中分别为2和1。显然，通过映射 ，高维通道具有更长的范围交互，而低维通道通过使用非线性映射进行较短的范围交互。

# 3.3. ECA Module for Deep CNNs

# 3.3. 用于深度CNN的ECA模块

Figure 2 illustrates the overview of our ECA module. After aggregating convolution features using GAP without dimensionality reduction, ECA module first adaptively determines kernel size , and then performs convolution followed by a Sigmoid function to learn channel attention. For applying our ECA to deep CNNs, we replace SE block by our ECA module following the same configuration in [14]. The resulting networks are named by ECA-Net. Figure 3 gives PyTorch code of our ECA.

图2展示了我们ECA模块的概览。在无需降维的情况下，使用全局平均池化（GAP）聚合卷积特征后，ECA模块首先自适应地确定卷积核大小 ，然后执行 卷积，并跟随Sigmoid函数来学习通道注意力。将我们的ECA应用于深度CNN时，我们用ECA模块替换SE块，并遵循文献[14]中的相同配置。得到的网络命名为ECA-Net。图3给出了我们ECA的PyTorch代码。

# 4. Experiments

# 4. 实验

In this section, we evaluate the proposed method on large-scale image classification, object detection and instance segmentation using ImageNet [6] and MS COCO [23], respectively. Specifically, we first assess the effect of kernel size on our ECA module, and compare with state-of-the-art counterparts on ImageNet. Then, we verify the effectiveness of our ECA-Net on MS COCO using Faster R-CNN [26], Mask R-CNN [10] and RetinaNet [22].

在本节中，我们分别在ImageNet [6] 和 MS COCO [23] 上评估所提出的方法在大规模图像分类、目标检测和实例分割上的表现。具体来说，我们首先评估核大小对我们ECA模块的影响，并在ImageNet上与最先进的方法进行比较。然后，我们使用Faster R-CNN [26]、Mask R-CNN [10] 和 RetinaNet [22] 在MS COCO上验证我们ECA-Net的有效性。

# 4.1. Implementation Details

# 4.1. 实施细节

To evaluate our ECA-Net on ImageNet classification, we employ four widely used CNNs as backbone models, including ResNet-50 [11], ResNet-101 [11], ResNet-512 [11] and MobileNetV2 [28]. For training ResNets with our ECA, we adopt exactly the same data augmentation and hyper-parameter settings in [11, 14]. Specifically, the input images are randomly cropped to with random horizontal flipping. The parameters of networks are optimized by stochastic gradient descent (SGD) with weight decay of 1e-4, momentum of 0.9 and mini-batch size of 256 . All models are trained within 100 epochs by setting the initial learning rate to 0.1 , which is decreased by a factor of 10 per 30 epochs. For training MobileNetV2 with our ECA, we follow the settings in [28], where networks are trained within 400 epochs using SGD with weight decay of 4e-5, momentum of 0.9 and mini-batch size of 96 . The initial learning rate is set to 0.045 , and is decreased by a linear decay rate of 0.98 . For testing on the validation set, the shorter side of an input image is first resized to 256 and a center crop of is used for evaluation. All models are implemented by PyTorch toolkit

为了评估我们的 ECA-Net 在 ImageNet 分类任务上的表现，我们使用了四种广泛应用的卷积神经网络（CNN）作为基础模型，包括 ResNet-50 [11]、ResNet-101 [11]、ResNet-512 [11] 和 MobileNetV2 [28]。在用我们的 ECA 训练 ResNets 时，我们采用了与 [11, 14] 中完全相同的数据增强和超参数设置。具体来说，输入图像被随机裁剪为 并进行随机水平翻转。网络参数通过带有权重衰减 1e-4、动量 0.9 和小批量大小 256 的随机梯度下降（SGD）进行优化。所有模型都在 100 个周期内训练，初始学习率设置为 0.1，每 30 个周期减少 10 倍。对于使用我们的 ECA 训练 MobileNetV2，我们遵循 [28] 中的设置，其中网络在 400 个周期内使用 SGD 进行训练，权重衰减为 4e-5，动量为 0.9，小批量大小为 96。初始学习率设置为 0.045，并通过线性衰减率 0.98 进行减少。在验证集上进行测试时，输入图像的短边首先调整大小为 256，然后使用中心裁剪的 进行评估。所有模型都是通过 PyTorch 工具包实现的。

We further evaluate our method on MS COCO using Faster R-CNN [26], Mask R-CNN [10] and RetinaNet [22], where ResNet-50 and ResNet-101 along with FPN [21] are used as backbone models. We implement all detectors by using MMDetection toolkit [3] and employ the default settings. Specifically, the shorter side of input images are resized to 800 , then all models are optimized using SGD with weight decay of 1e-4, momentum of 0.9 and mini-batch size of 8 (4 GPUs with 2 images per GPU). The learning rate is initialized to 0.01 and is decreased by a factor of 10 after 8 and 11 epochs, respectively. We train all detectors within 12 epochs on train2017 of COCO and report the results on val2017 for comparison. All programs run on a PC equipped with four RTX 2080Ti GPUs and an Intel(R)

我们进一步在 MS COCO 上使用 Faster R-CNN [26]、Mask R-CNN [10] 和 RetinaNet [22] 评估我们的方法，其中 ResNet-50 和 ResNet-101 以及 FPN [21] 被用作基础模型。我们通过使用 MMDetection 工具包 [3] 实现所有检测器并采用默认设置。具体来说，输入图像的短边被调整到 800 像素，然后所有模型都使用带有权重衰减 1e-4、动量 0.9 和小批量大小 8（每个 GPU 2 张图像，共 4 个 GPUs）的 SGD 进行优化。学习率初始化为 0.01，在第 8 和 11 个周期后分别以 10 倍的因子减少。我们在 COCO 的 train2017 上训练所有检测器 12 个周期，并将结果报告在 val2017 上进行比较。所有程序都在配备有四个 RTX 2080Ti GPUs 和 Intel(R) Xeon Silver 4112 CPU@2.60GHz 的个人计算机上运行。

Ihttps://github.com/BangguWu/ECANet

https://github.com/BangguWu/ECANet

| Method | Backbone Models | #.Param. | FLOPs | Training | Inference | Top-1 | Top-5 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ResNet [11] SENet [14] CBAM [33] -Nets [4] GCNet [2] GSoP-Net1 [9] AA-Net ECA-Net (Ours) | ResNet-50 | 24.37M | 3.86G | 1024 FPS | 1855 FPS | 75.20 | 92.52 |
| 26.77M | 3.87G | 759 FPS | 1620 FPS | 76.71 | 93.38 |
| 26.77M | 3.87G | 472 FPS | 1213 FPS | 77.34 | 93.69 |
|  | 6.50G | N/A | N/A | 77.00 | 93.50 |
|  | 3.87G | N/A | N/A | 77.70 | 93.66 |
| 28.05M | 6.18G | 596 FPS | 1383 FPS | 77.68 | 93.98 |
| 25.80M | 4.15G | N/A | N/A | 77.70 | 93.80 |
| 24.37M | 3.86G | 785 FPS | 1805 FPS | 77.48 | 93.68 |
| ResNet [11] | ResNet-101 | 42.49M | 7.34G | 386 FPS | 1174 FPS | 76.83 | 93.48 |
| SENet [14] | 47.01M | 7.35G | 367 FPS | 1044 FPS | 77.62 | 93.93 |
| CBAM [33] | 47.01M | 7.35G | 270 FPS | 635 FPS | 78.49 | 94.31 |
| AA-Net | 45.40M | 8.05G | N/A | N/A | 78.70 | 94.40 |
| ECA-Net (Ours) | 42.49M | 7.35G | 380 FPS | 1089 FPS | 78.65 | 94.34 |
| ResNet [11] | ResNet-152 | 57.40M | 10.82G | 281 FPS | 815 FPS | 77.58 | 93.66 |
| SENet [14] | 63.68M | 10.85G | 268 FPS | 761 FPS | 78.43 | 94.27 |
| ECA-Net (Ours) | 57.40M | 10.83G | 279 FPS | 785 FPS | 78.92 | 94.55 |
| MobileNetV2 [28] | MobileNetV2 | 3.34M | 319.4M | 711 FPS | 2086 FPS | 71.64 | 90.20 |
| SENet | 3.40M | 320.1M | 671 FPS | 2000 FPS | 72.42 | 90.67 |
| ECA-Net (Ours) | 3.34M | 319.9M | 676 FPS | 2010 FPS | 72.56 | 90.81 |

Table 3. Comparison of different attention methods on ImageNet in terms of network parameters (#.Param.), floating point operations per second (FLOPs), training or inference speed (frame per second, FPS), and Top-1/Top-5 accuracy (in %). : Since the source code and models of -Nets and AA-Net are publicly unavailable, we do not compare their running time. : AA-Net is trained with Inception data augmentation and different setting of learning rates.

表 3. 在 ImageNet 上不同注意力方法的比较，包括网络参数 (#.Param.)、每秒浮点运算次数 (FLOPs)、训练或推理速度（每秒帧数，FPS）以及 Top-1/Top-5 准确率（百分比）。 ：由于 -Nets 和 AA-Net 的源代码和模型公开不可用，我们没有比较它们的运行时间。 ：AA-Net 是在带有 Inception 数据增强和不同学习率设置的条件下训练的。

# Xeon Silver 4112 CPU@2.60GHz.

# Xeon Silver 4112 CPU@2.60GHz。

# 4.2. Image Classification on ImageNet-1K

# 4.2. 在 ImageNet-1K 上的图像分类

Here, we first assess the effect of kernel size on our ECA module and verify the effectiveness of our method to adaptively determine kernel size, then we compare with state-of-the-art counterparts and CNN models using ResNet-50, ResNet-101, ResNet-152 and MobileNetV2.

在这里，我们首先评估核大小对我们ECA模块的影响，并验证我们方法的有效性，该方法能够自适应地确定核大小，然后我们与使用ResNet-50、ResNet-101、ResNet-152和MobileNetV2的最新对比方法和CNN模型进行比较。

# 4.2.1 Effect of Kernel Size on ECA Module

# 4.2.1 核大小 对ECA模块的影响

As shown in Eq. (9), our ECA module involves a parameter , i.e., kernel size of convolution. In this part, we evaluate its effect on our ECA module and validate the effectiveness of our method for adaptive selection of kernel size. To this end, we employ ResNet-50 and ResNet-101 as backbone models, and train them with our ECA module by setting be from 3 to 9 . The results are illustrated in Figure 4 from it we have the following observations.

如公式（9）所示，我们的ECA模块涉及一个参数 ，即 卷积的核大小。在本部分，我们评估它对我们ECA模块的影响，并验证我们方法在自适应选择核大小方面的有效性。为此，我们使用ResNet-50和ResNet-101作为基础模型，并使用我们的ECA模块对它们进行训练，设置 从3到9。结果在图4中说明，我们从中得出以下观察结果。

Firstly, when is fixed in all convolution blocks, ECA module obtains the best results at and for ResNet-50 and ResNet-101, respectively. Since ResNet- 101 has more intermediate layers that dominate performance of ResNet-101, it may prefer to small kernel size. Besides, these results show that different deep CNNs have various optimal , and has a clear effect on performance of ECA-Net. Furthermore, accuracy fluctuations of ResNet-101 are larger than those of ResNet- 50 , and we conjecture the reason is that the deeper networks are more sensitive to the fixed kernel size than the shallower ones. Additionally, kernel size that is adaptively determined by Eq. (12) usually outperforms the fixed ones, while it can avoid manual tuning of parameter via cross-validation. Above results demonstrate the effectiveness of our adaptive kernel size selection in attaining better and stable results. Finally, ECA module with various numbers of consistently outperform SE block, verifying that avoiding dimensionality reduction and local cross-channel interaction have positive effects on learning channel attention.

首先，当 在所有卷积块中固定时，ECA模块在 和 分别为 ResNet-50 和 ResNet-101 获得了最佳结果。由于 ResNet-101 具有更多影响其性能的中间层，它可能更倾向于使用较小的核大小。此外，这些结果表明不同的深度 CNN 具有各自最优的 ，且 对 ECA-Net 的性能有显著影响。进一步地，ResNet-101 的准确度波动 大于 ResNet-50 的 ，我们推测原因是更深的网络对固定的核大小比浅层网络更为敏感。另外，由公式 (12) 自适应确定的核大小通常优于固定大小，同时它可以通过交叉验证避免手动调整参数 。上述结果证明了我们自适应核大小选择在获得更好且稳定结果方面的有效性。最后，带有不同数量 的 ECA 模块始终优于 SE 模块，验证了避免维度减少和局部跨通道交互对学习通道注意力有积极影响。

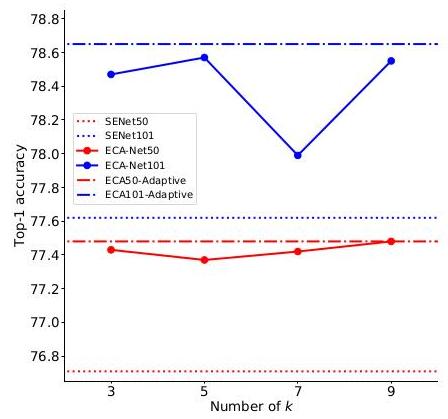


Figure 4. Results of our ECA module with various numbers of using ResNet-50 and ResNet-101 as backbone models. Here, we also give the results of ECA module with adaptive selection of kernel size and compare with SENet as baseline.

图 4. 使用 ResNet-50 和 ResNet-101 作为基础模型的 ECA 模块在不同数量 下的结果。这里，我们还给出了具有自适应核大小选择的 ECA 模块的结果，并与 SENet 作为基线进行比较。

# 4.2.2 Comparisons Using Different Deep CNNs

# 4.2.2 使用不同深度 CNN 的比较

ResNet-50 We compare our ECA module with several state-of-the-art attention methods using ResNet-50 on Im-ageNet, including SENet [14], CBAM [33], -Nets [4], AA-Net [1], GSoP-Net1 [9] and GCNet [2]. The evaluation metrics include both efficiency (i.e., network parameters, floating point operations per second (FLOPs) and training/inference speed) and effectiveness (i.e., Top-1/Top- 5 accuracy). For comparison, we duplicate the results of ResNet and SENet from [14], and report the results of other compared methods in their original papers. To test training/inference speed of various models, we employ publicly available models of the compared CNNs, and run them on the same computing platform. The results are given in Table 3, where we can see that our ECA-Net shares almost the same model complexity (i.e., network parameters, FLOPs and speed) with the original ResNet-50, while achieving gains in Top-1 accuracy. Comparing with state-of-the-art counterparts (i.e., SENet, CBAM, -Nets, AA-Net, GSoP-Net1 and GCNet), ECA-Net obtains better or competitive results while benefiting lower model complexity.

我们使用 ResNet-50 在 ImageNet 上将我们的 ECA 模块与几种最先进的注意力方法进行比较，包括 SENet [14]，CBAM [33]， -Nets [4]，AA-Net [1]，GSoP-Net1 [9] 和 GCNet [2]。评估指标包括效率（即网络参数、每秒浮点运算次数（FLOPs）以及训练/推理速度）和有效性（即 Top-1/Top-5 准确率）。为了进行比较，我们从 [14] 复制了 ResNet 和 SENet 的结果，并报告了其他比较方法在他们原始论文中的结果。为了测试各种模型的训练/推理速度，我们使用了公开可用的比较 CNN 模型，并在相同的计算平台上运行它们。结果如表 3 所示，我们可以看到我们的 ECA-Net 与原始 ResNet-50 几乎拥有相同的模型复杂度（即网络参数、FLOPs 和速度），同时在 Top-1 准确率上取得了 的提升。与最先进的对比方法（即 SENet、CBAM、 -Nets、AA-Net、GSoP-Net1 和 GCNet）相比，ECA-Net 在保持较低的模型复杂度的同时，获得了更好或具有竞争力的结果。

ResNet-101 Using ResNet-101 as backbone model, we compare our ECA-Net with SENet [14], CBAM [33] and AA-Net [1]. From Table 3 we can see that ECA-Net outperforms the original ResNet-101 by with almost the same model complexity. Sharing the same tendency on ResNet-50, ECA-Net is superior to SENet and CBAM while it is very competitive to AA-Net with lower model complexity. Note that AA-Net is trained with Inception data augmentation and different setting of learning rates.

使用ResNet-101作为基础模型，我们将我们的ECA-Net与SENet [14]、CBAM [33]和AA-Net [1]进行了比较。从表3中我们可以看到，ECA-Net在几乎相同的模型复杂度下，比原始的ResNet-101提高了 。在ResNet-50上也有相同的趋势，ECA-Net优于SENet和CBAM，而其模型复杂度低于AA-Net时，仍具有竞争力。注意AA-Net是在Inception数据增强和不同的学习率设置下训练的。

ResNet-152 Using ResNet-152 as backbone model, we compare our ECA-Net with SENet [14]. From Table 3 we can see that ECA-Net improves the original ResNet-152 over about in terms of Top-1 accuracy with almost the same model complexity. Comparing with SENet, ECA-Net achieves gain in terms of Top-1 with lower model complexity. The results with respect to ResNet-50, ResNet- 101 and ResNet-152 demonstrate the effectiveness of our ECA module on the widely used ResNet architectures.

使用ResNet-152作为基础模型，我们将我们的ECA-Net与SENet [14]进行了比较。从表3中我们可以看到，ECA-Net在几乎相同的模型复杂度下，将原始ResNet-152在Top-1准确度上提高了大约 。与SENet相比，ECA-Net在模型复杂度更低的情况下实现了 的Top-1准确度提升。关于ResNet-50、ResNet-101和ResNet-152的结果证明了我们的ECA模块在广泛使用的ResNet架构上的有效性。

MobileNetV2 Besides ResNet architectures, we also verify the effectiveness of our ECA module on lightweight CNN architectures. To this end, we employ MobileNetV2 [28] as backbone model and compare our ECA module with SE block. In particular, we integrate SE block and ECA module in convolution layer before residual connection lying in each ’bottleneck’ of MobileNetV2, and parameter of SE block is set to 8 . All models are trained using exactly the same settings. The results in Table 3 show our ECA-Net improves the original MobileNetV2 and SENet by about and in terms of Top-1 accuracy, respectively. Furthermore, our ECA-Net has smaller model size and faster training/inference speed than SENet. Above results verify the efficiency and effectiveness of our ECA module again.

MobileNetV2 除了ResNet架构外，我们还验证了我们的ECA模块在轻量级CNN架构上的有效性。为此，我们采用MobileNetV2 [28] 作为基础模型，并将我们的ECA模块与SE块进行比较。特别是，我们在MobileNetV2中每个“瓶颈”的残差连接之前的卷积层中集成了SE块和ECA模块，SE块的参数 设置为8。所有模型都使用完全相同的设置进行训练。表3的结果显示，我们的ECA-Net在Top-1准确度方面分别提高了原始MobileNetV2和SENet约 和 。此外，我们的ECA-Net比SENet具有更小的模型大小和更快的训练/推理速度。上述结果再次验证了我们的ECA模块的效率和有效性。

| CNN Models | #.Param. | FLOPs | Top-1 | Top-5 |
| --- | --- | --- | --- | --- |
| ResNet-200 | 74.45M | 14.10G | 78.20 | 94.00 |
| Inception-v3 | 25.90M | 5.36G | 77.45 | 93.56 |
| ResNeXt-101 | 46.66M | 7.53G | 78.80 | 94.40 |
| DenseNet-264 (k=32) | 31.79M | 5.52G | 77.85 | 93.78 |
| DenseNet-161 (k=48) | 27.35M | 7.34G | 77.65 | 93.80 |
| ECA-Net50 (Ours) | 24.37M | 3.86G | 77.48 | 93.68 |
| ECA-Net101 (Ours) | 42.49M | 7.35G | 78.65 | 94.34 |

Table 4. Comparisons with state-of-the-art CNNs on ImageNet.

表4。与ImageNet上最先进的CNN的比较。

# 4.2.3 Comparisons with Other CNN Models

# 4.2.3 与其他CNN模型的比较

At the end of this part, we compare our ECA-Net50 and ECA-Net101 with other state-of-the-art CNN models, including ResNet-200 [12], Inception-v3 [31], ResNeXt [34], DenseNet [15]. These CNN models have deeper and wider architectures, and their results all are copied from the original papers. As presented in Table 4 ECA-Net101 outperforms ResNet-200, indicating that our ECA-Net can improve the performance of deep CNNs using much less computational cost. Meanwhile, our ECA-Net101 is very competitive to ResNeXt-101, while the latter one employs more convolution filters and expensive group convolutions. In addition, ECA-Net50 is comparable to DenseNet-264 (k=32), DenseNet-161 (k=48) and Inception-v3, but it has lower model complexity. All above results demonstrate that our ECA-Net performs favorably against state-of-the-art CNNs while benefiting much lower model complexity. Note that our ECA also has great potential to further improve the performance of the compared CNN models.

在本部分结束时，我们将我们的ECA-Net50和ECA-Net101与其他最先进的卷积神经网络模型进行了比较，包括ResNet-200 [12]、Inception-v3 [31]、ResNeXt [34]和DenseNet [15]。这些卷积神经网络模型具有更深层和更宽泛的结构，它们的结果都是从原始论文中复制的。如表4所示，ECA-Net101优于ResNet-200，表明我们的ECA-Net能够使用更少的计算成本提高深度卷积神经网络的性能。同时，我们的ECA-Net101与ResNeXt-101非常具有竞争力，而后者使用了更多的卷积滤波器和昂贵的组卷积。此外，ECA-Net50与DenseNet-264 (k=32)、DenseNet-161 (k=48)和Inception-v3相当，但其模型复杂度更低。以上所有结果都表明，我们的ECA-Net在与最先进的卷积神经网络相比时表现优异，同时具有更低的模型复杂度。请注意，我们的ECA也有巨大的潜力进一步改进比较过的卷积神经网络模型的性能。

# 4.3. Object Detection on MS COCO

# 4.3. MS COCO上的目标检测

In this subsection, we evaluate our ECA-Net on object detection task using Faster R-CNN [26], Mask R-CNN [10] and RetinaNet [22]. We mainly compare ECA-Net with ResNet and SENet. All CNN models are pre-trained on Im-ageNet, then are transferred to MS COCO by fine-tuning.

在本小节中，我们使用Faster R-CNN [26]、Mask R-CNN [10]和RetinaNet [22]在目标检测任务上评估我们的ECA-Net。我们主要将ECA-Net与ResNet和SENet进行比较。所有卷积神经网络模型都在ImageNet上预训练，然后通过微调转移到MS COCO。

# 4.3.1 Comparisons Using Faster R-CNN

# 4.3.1 使用Faster R-CNN的比较

Using Faster R-CNN as the basic detector, we employ ResNets of 50 and 101 layers along with FPN [21] as backbone models. As shown in Table 5, integration of either SE block or our ECA module can improve performance of object detection by a clear margin. Meanwhile, our ECA outperforms SE block by and in terms of AP using ResNet-50 and ResNet-101, respectively.

使用 Faster R-CNN 作为基本检测器，我们采用具有 50 和 101 层的 ResNets 以及 FPN [21] 作为基础模型。如表 5 所示，集成 SE 块或我们的 ECA 模块都能明显提高目标检测的性能。同时，我们的 ECA 在使用 ResNet-50 和 ResNet-101 时，分别以 和 的 AP 值优于 SE 块。

# 4.3.2 Comparisons Using Mask R-CNN

# 4.3.2 使用 Mask R-CNN 进行比较

We further exploit Mask R-CNN to verify the effectiveness of our ECA-Net on object detection task. As shown in Table 5, our ECA module is superior to the original ResNet by and in terms of AP under the settings of 50 and 101 layers, respectively. Meanwhile, ECA module achieves and gains over SE block using ResNet- 50 and ResNet-101 as backbone models, respectively. Using ResNet-50, ECA is superior to one NL [32], and is comparable to GC block [2] using lower model complexity.

我们进一步利用 Mask R-CNN 验证我们的 ECA-Net 在目标检测任务上的有效性。如表 5 所示，我们的 ECA 模块在 50 和 101 层的设置下，分别以 和 的 AP 值优于原始的 ResNet。同时，ECA 模块在使用 ResNet-50 和 ResNet-101 作为基础模型时，分别比 SE 块提高了 和 。使用 ResNet-50 时，ECA 优于一个 NL [32]，并且在较低的模型复杂度下与 GC 块 [2] 相当。

| Methods | Detectors | #.Param. | GFLOPs | AP |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ResNet-50 | Faster R-CNN | 41.53 M | 207.07 | 36.4 | 58.2 | 39.2 | 21.8 | 40.0 | 46.2 |
| + SE block | 44.02 M | 207.18 | 37.7 | 60.1 | 40.9 | 22.9 | 41.9 | 48.2 |
| + ECA (Ours) | 41.53 M | 207.18 | 38.0 | 60.6 | 40.9 | 23.4 | 42.1 | 48.0 |
| ResNet-101 | 60.52 M | 283.14 | 38.7 | 60.6 | 41.9 | 22.7 | 43.2 | 50.4 |
| + SE block | 65.24 M | 283.33 | 39.6 | 62.0 | 43.1 | 23.7 | 44.0 | 51.4 |
| + ECA (Ours) | 60.52 M | 283.32 | 40.3 | 62.9 | 44.0 | 24.5 | 44.7 | 51.3 |
| ResNet-50 | Mask R-CNN | 44.18 M | 275.58 | 37.2 | 58.9 | 40.3 | 22.2 | 40.7 | 48.0 |
| + SE block | 46.67 M | 275.69 | 38.7 | 60.9 | 42.1 | 23.4 | 42.7 | 50.0 |
|  | 46.50 M | 288.70 | 38.0 | 59.8 | 41.0 | N/A | N/A | N/A |
| + GC block | 46.90 M | 279.60 | 39.4 | 61.6 | 42.4 | N/A | N/A | N/A |
| + ECA (Ours) | 44.18 M | 275.69 | 39.0 | 61.3 | 42.1 | 24.2 | 42.8 | 49.9 |
| ResNet-101 | 63.17 M | 351.65 | 39.4 | 60.9 | 43.3 | 23.0 | 43.7 | 51.4 |
| + SE block | 67.89 M | 351.84 | 40.7 | 62.5 | 44.3 | 23.9 | 45.2 | 52.8 |
| + ECA (Ours) | 63.17 M | 351.83 | 41.3 | 63.1 | 44.8 | 25.1 | 45.8 | 52.9 |
| ResNet-50 | RetinaNet | 37.74 M | 239.32 | 35.6 | 55.5 | 38.2 | 20.0 | 39.6 | 46.8 |
| + SE block | 40.23 M | 239.43 | 37.1 | 57.2 | 39.9 | 21.2 | 40.7 | 49.3 |
| + ECA (Ours) | 37.74 M | 239.43 | 37.3 | 57.7 | 39.6 | 21.9 | 41.3 | 48.9 |
| ResNet-101 | 56.74 M | 315.39 | 37.7 | 57.5 | 40.4 | 21.1 | 42.2 | 49.5 |
| + SE block | 61.45 M | 315.58 | 38.7 | 59.1 | 41.6 | 22.1 | 43.1 | 50.9 |
| + ECA (Ours) | 56.74 M | 315.57 | 39.1 | 59.9 | 41.8 | 22.8 | 43.4 | 50.6 |

Table 5. Object detection results of different methods on COCO val2017.

表 5. 不同方法在 COCO val2017 上的目标检测结果。

# 4.3.3 Comparisons Using RetinaNet

# 4.3.3 使用 RetinaNet 进行比较

Additionally, we verify the effectiveness of our ECA-Net on object detection using one-stage detector, i.e., RetinaNet. As compared in Table 5, our ECA-Net outperforms the original ResNet by and in terms of AP for the networks of 50 and 101 layers, respectively. Meanwhile, ECA-Net improves SE-Net over and for ResNet-50 and ResNet-101, respectively. In summary, the results in Table 5 demonstrate that our ECA-Net can well generalize to object detection task. Specifically, ECA module brings clear improvement over the original ResNet, while outperforming SE block using lower model complexity. In particular, our ECA module achieves more gains for small objects, which are usually more difficult to be detected.

此外，我们通过使用单阶段检测器 RetinaNet 验证了我们的 ECA-Net 在目标检测上的有效性。如表 5 所示，我们的 ECA-Net 在 50 层和 101 层的网络中，分别以 和 的 AP 超越了原始的 ResNet。同时，ECA-Net 分别比 SE-Net 在 ResNet-50 和 ResNet-101 上提高了 和 。总之，表 5 的结果证明了我们的 ECA-Net 可以很好地泛化到目标检测任务上。特别是，ECA 模块在原始 ResNet 基础上带来了明显的改进，同时在使用更低的模型复杂度下超越了 SE 块。特别是，我们的 ECA 模块对于小目标的提升更大，这些目标通常更难以检测。

# 4.4. Instance Segmentation on MS COCO

# 4.4. 在 MS COCO 上的实例分割

Then, we give instance segmentation results of our ECA module using Mask R-CNN on MS COCO. As compared in Table 6, ECA module achieves notable gain over the original ResNet while performing better than SE block with less model complexity. For ResNet-50 as backbone, ECA with lower model complexity is superior one NL [32], and is comparable to GC block [2]. These results verify our ECA module has good generalization ability for various tasks.

接着，我们使用 Mask R-CNN 在 MS COCO 上给出了我们的 ECA 模块的实例分割结果。如表 6 所示，ECA 模块在保持较低的模型复杂度的同时，比原始 ResNet 和 SE 块表现更佳。对于以 ResNet-50 为骨干网络，具有更低模型复杂度的 ECA 优于 NL [32]，并且与 GC 块 [2] 相当。这些结果验证了我们的 ECA 模块对于各种任务具有良好的泛化能力。

| Methods | AP |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ResNet-50 | 34.1 | 55.5 | 36.2 | 16.1 | 36.7 | 50.0 |
| + SE block | 35.4 | 57.4 | 37.8 | 17.1 | 38.6 | 51.8 |
| + 1 NL | 34.7 | 56.7 | 36.6 | N/A | N/A | N/A |
| + GC block | 35.7 | 58.4 | 37.6 | N/A | N/A | N/A |
| + ECA (Ours) | 35.6 | 58.1 | 37.7 | 17.6 | 39.0 | 51.8 |
| ResNet-101 | 35.9 | 57.7 | 38.4 | 16.8 | 39.1 | 53.6 |
| + SE block | 36.8 | 59.3 | 39.2 | 17.2 | 40.3 | 53.6 |
| + ECA (Ours) | 37.4 | 59.9 | 39.8 | 18.1 | 41.1 | 54.1 |

Table 6. Instance segmentation results of different methods using Mask R-CNN on COCO val2017.

表 6. 使用 Mask R-CNN 在 COCO val2017 上不同方法的实例分割结果。

# 5. Conclusion

# 5. 结论

In this paper, we focus on learning effective channel attention for deep CNNs with low model complexity. To this end, we propose an efficient channel attention (ECA) module, which generates channel attention through a fast convolution, whose kernel size can be adaptively determined by a non-linear mapping of channel dimension. Experimental results demonstrate our ECA is an extremely lightweight plug-and-play block to improve the performance of various deep CNN architectures, including the widely used ResNets and lightweight MobileNetV2. Moreover, our ECA-Net exhibits good generalization ability in object detection and instance segmentation tasks. In future, we will apply our ECA module to more CNN architectures (e.g., ResNeXt and Inception [31]) and further investigate incorporation of ECA with spatial attention module.

在本文中，我们关注于为深度CNNs（卷积神经网络）学习有效的通道注意力，同时保持较低的模型复杂度。为此，我们提出了一个高效的通道注意力（ECA）模块，该模块通过快速的 卷积生成通道注意力，其卷积核大小可以通过通道维度的非线性映射自适应确定。实验结果表明，我们的ECA是一个极其轻量级的即插即用模块，能够提高各种深度CNN架构的性能，包括广泛使用的ResNets和轻量级的MobileNetV2。此外，我们的ECA-Net在目标检测和实例分割任务中表现出良好的泛化能力。未来，我们将把我们的ECA模块应用到更多的CNN架构中（例如ResNeXt和Inception [31]），并进一步研究ECA与空间注意力模块的结合。

# References

# 参考文献

[1] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. arXiv:1904.09925, 2019.

[2] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Genet: Non-local networks meet squeeze-excitation networks and beyond. In ICCV Workshops, 2019.

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDe-tection: Open mmlab detection toolbox and benchmark. arXiv:1906.07155, 2019.

[4] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A -Nets: Double attention networks. In NIPS, 2018.

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In CVPR, 2017.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In CVPR, 2009.

[7] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In CVPR, 2019.

[8] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. Chan-nelnets: Compact and efficient convolutional neural networks via channel-wise convolutions. In NeurIPS, 2018.

[9] Zilin Gao, Jiangtao Xie, Qilong Wang, and Peihua Li. Global second-order pooling convolutional networks. In CVPR, 2019.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In ICCV, pages 2980-2988, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In ECCV, 2016.

[13] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In NeurIPS, 2018.

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In CVPR, 2018.

[15] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In CVPR, 2017.

[16] Yani Ioannou, Duncan Robertson, Roberto Cipolla, and Antonio Criminisi. Deep roots: Improving cnn efficiency with hierarchical filter groups. In CVPR, 2017.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In NIPS, 2012.

[18] Huayu Li. Channel locality block: A variant of squeeze-and-excitation. arXiv, 1901.01493, 2019.

[19] Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. Is second-order information helpful for large-scale visual recognition? In ICCV, 2017.

[20] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Factorized bilinear models for image recognition. In ICCV, 2017.

[21] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In CVPR, pages 936-944, 2017.

[22] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In ICCV, 2017.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In ECCV, 2014.

[24] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: Practical guidelines for efficient CNN architecture design. In ECCV, 2018.

[25] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.

[26] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell., 39(6):1137-1149, 2017.

[27] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Recalibrating fully convolutional networks with spatial and channel "squeeze and excitation" blocks. IEEE Trans. Med. Imaging, 38(2):540–549, 2019.

[28] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In CVPR, 2018.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In CVPR, 2015.

[31] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In CVPR, 2016.

[32] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaim-ing He. Non-local neural networks. In CVPR, 2018.

[33] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In ECCV, 2018.

[34] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In CVPR, 2017.

[35] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In ECCV, pages 818-833, 2014.

[36] Dong-Qing Zhang. Clenet: Improving the efficiency of convolutional neural network using channel local convolutions. In CVPR, 2018.

| Method | CNNs | #.Param. | GFLOPs | Top-1 | Top-5 |
| --- | --- | --- | --- | --- | --- |
| ResNet [11] | R-18 | 11.148M | 1.699 | 70.40 | 89.45 |
| SENet [14] | 11.231M | 1.700 | 70.59 | 89.78 |
| CBAM [33] | 11.234M | 1.700 | 70.73 | 89.91 |
| ECA-Net (Ours) | 11.148M | 1.700 | 70.78 | 89.92 |
| ResNet [11] | R-34 | 20.788M | 3.427 | 73.31 | 91.40 |
| SENet [14] | 20.938M | 3.428 | 73.87 | 91.65 |
| CBAM [33] | 20.943M | 3.428 | 74.01 | 91.76 |
| ECA-Net (Ours) | 20.788M | 3.428 | 74.21 | 91.83 |

Table 7. Comparison of different methods using ResNet-18 (R-18) and ResNet-34 (R-34) on ImageNet in terms of network parameters (#.Param.), floating point operations per second (FLOPs), and Top-1/Top-5 accuracy (in %).

表7. 使用ResNet-18（R-18）和ResNet-34（R-34）在ImageNet上的不同方法在以下方面的比较：网络参数（#.Param.），每秒浮点运算次数（FLOPs）以及Top-1/Top-5准确率（百分比）。

[37] Ting Zhang, Guo-Jun Qi, Bin Xiao, and Jingdong Wang. Interleaved group convolutions. In .

[37] 张婷，祁国军，肖斌，汪精东。交错组卷积。在 中。

[38] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In CVPR, 2018.

[38] 张翔宇，周新宇，林梦晓，孙剑。Shufflenet：一种针对移动设备的极其高效的卷积神经网络。在CVPR，2018年。

# Appendix I: Comparison of Different Methods using ResNet-18 and ResNet-34 on ImageNet

# 附录I：使用ResNet-18和ResNet-34在ImageNet上不同方法的比较

Here, we compare different attention methods using ResNet-18 and ResNet-34 on ImageNet. The results are listed in Table 7, where the results of ResNet, SENet and CBAM are duplicated from [33], and we train ECA-Net using the settings of hyper-parameters with [33]. From Table 7, we can see that our ECA-Net improves the original ResNet-18 and ResNet-34 over abd in Top-1 accuracy, respectively. Comparing with SENet and CBAM, our ECA-Net achieves better performance using less model complexity, showing the effectiveness of the proposed ECA module.

在这里，我们使用 ResNet-18 和 ResNet-34 在 ImageNet 上比较了不同的注意力方法。结果列在表7中，其中 ResNet、SENet 和 CBAM 的结果是从 [33] 复制的，并且我们使用 [33] 中的超参数设置来训练 ECA-Net。从表7中，我们可以看到我们的 ECA-Net 分别在 Top-1 准确度上超过了原始的 ResNet-18 和 ResNet-34 的 和 。与 SENet 和 CBAM 相比，我们的 ECA-Net 在使用更少的模型复杂度的情况下实现了更好的性能，显示了所提出 ECA 模块的有效性。

# Appendix II: Stacking More 1D Convolutions in ECA Module

# 附录II：在ECA模块中堆叠更多1D卷积

Intuitively, more 1D convolutions stacked in ECA module may bring further improvement, due to increase of modeling capability. Actually, we found that one extra 1D convolution brings trivial gains at the cost of slightly increasing complexity, but more 1D convolutions degrade performance, which may be caused by that more 1D convolutions make gradient backpropagation more difficult. Therefore, our final ECA module contains only one 1D convolution.

直观上，ECA模块中堆叠更多的1D卷积可能会带来进一步的改进，因为建模能力增强了。实际上，我们发现增加一个额外的1D卷积在略微增加复杂度的代价下带来了微小的收益 ，但是更多的1D卷积会降低性能，这可能是由于更多的1D卷积使得梯度反向传播更加困难。因此，我们的最终ECA模块只包含一个1D卷积。

# Appendix III: Visualization of Weights Learned by ECA Modules and SE Blocks

# 附录III：ECA模块和SE块学习到的权重的可视化

To further analyze the effect of our ECA module on learning channel attention, we visualize the weights learned by ECA modules and compare with SE blocks. Here, we employ ResNet-50 as backbone model, and illustrate weights of different convolution blocks. Specifically, we randomly sample four classes from ImageNet dataset, which are hammerhead shark, ambulance, medicine chest and butternut squash, respectively. Some example images are illustrated in Figure 5. After training the networks, for all images of each class collected from validation set of ImageNet, we compute the channel weights of convolution blocks on average. Figure 6 visualizes the channel weights of conv\_ , which indicates -th convolution block in - th stage. Besides the visualization results of four random sampled classes, we also give the distribution of the average weights across classes as reference. The channel weights learned by ECA modules and SE blocks are illustrated in bottom and top of each row, respectively.

为了进一步分析我们的ECA模块对学习通道注意力的效果，我们可视化了ECA模块学到的权重，并与SE块进行了比较。在这里，我们使用ResNet-50作为基础模型，并展示了不同卷积块的权重。具体来说，我们从ImageNet数据集中随机抽取了四个类别，分别是锤头鲨、救护车、药箱和南瓜。一些示例图像如图5所示。训练网络后，对于从ImageNet验证集中收集的每个类别的所有图像，我们计算了卷积块的平均通道权重。图6可视化了conv\_ 的通道权重，这表示第 个卷积块在第 个阶段。除了四个随机抽样类别的可视化结果外，我们还给出了跨 类别的平均权重分布作为参考。ECA模块和SE块学到的通道权重分别展示在每行的底部和顶部。

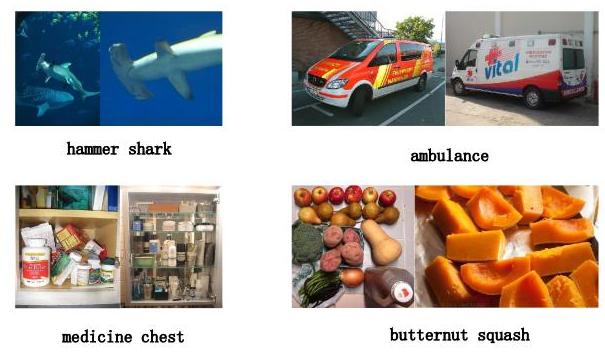


Figure 5. Example images of four random sampled classes on Im-ageNet, including hammerhead shark, ambulance, medicine chest and butternut squash.

图5. ImageNet上四个随机抽样类别的示例图像，包括锤头鲨、救护车、药箱和南瓜。

From Figure 6 we have the following observations. Firstly, for both ECA modules and SE blocks, the distributions of channel weights for different classes are very similar at the earlier layers (i.e., ones from conv\_2\_1 to conv\_3\_4), which may be by reason of that the earlier layers aim at capturing the basic elements (e.g., boundaries and corners) [35]. These features are almost similar for different classes. Such phenomenon also was described in the extended version of . Secondly, for the channel weights of different classes learned by SE blocks, most of them tend to be the same (i.e.,0.5) in conv\_4\_2 conv\_4\_5 while the differences among various classes are not obvious. On the contrary, the weights learned by ECA modules are clearly different across various channels and classes. Since convolution blocks in 4-th stage prefer to learn semantic information, so the weights learned by ECA modules can better distinguish different classes. Finally, convolution blocks in the final stage (i.e., conv\_5\_1, conv\_5\_2 and conv\_5\_3) capture high-level semantic features and they are more class-specific. Obviously, the weights learned by ECA modules are more class-specific than ones learned by SE blocks. Above results clearly demonstrate that the weights learned by our ECA modules have better discriminative ability.

从图6中我们可以得出以下观察。首先，对于ECA模块和SE块，不同类别在早期层（即从conv\_2\_1到conv\_3\_4）的通道权重分布非常相似，这可能是因为早期层旨在捕捉基本元素（例如，边界和角落）[35]。这些特征对于不同的类别几乎是相似的。这种现象也在 的扩展版本中有所描述。其次，对于SE块学习到的不同类别的通道权重，在conv\_4\_2 conv\_4\_5中，它们大多数倾向于相同（即0.5），而不同类别之间的差异不明显。相反，ECA模块学习到的权重在各个通道和类别之间明显不同。由于第四阶段的卷积块倾向于学习语义信息，因此ECA模块学习到的权重可以更好地区分不同类别。最后，最终阶段的卷积块（即conv\_5\_1、conv\_5\_2和conv\_5\_3）捕捉到高级语义特征，它们更具类别特异性。显然，ECA模块学习到的权重比SE块学习到的权重更具类别特异性。上述结果清楚地表明，我们的ECA模块学习到的权重具有更好的判别能力。

https://arxiv.org/abs/1709.01507

https://arxiv.org/abs/1709.01507

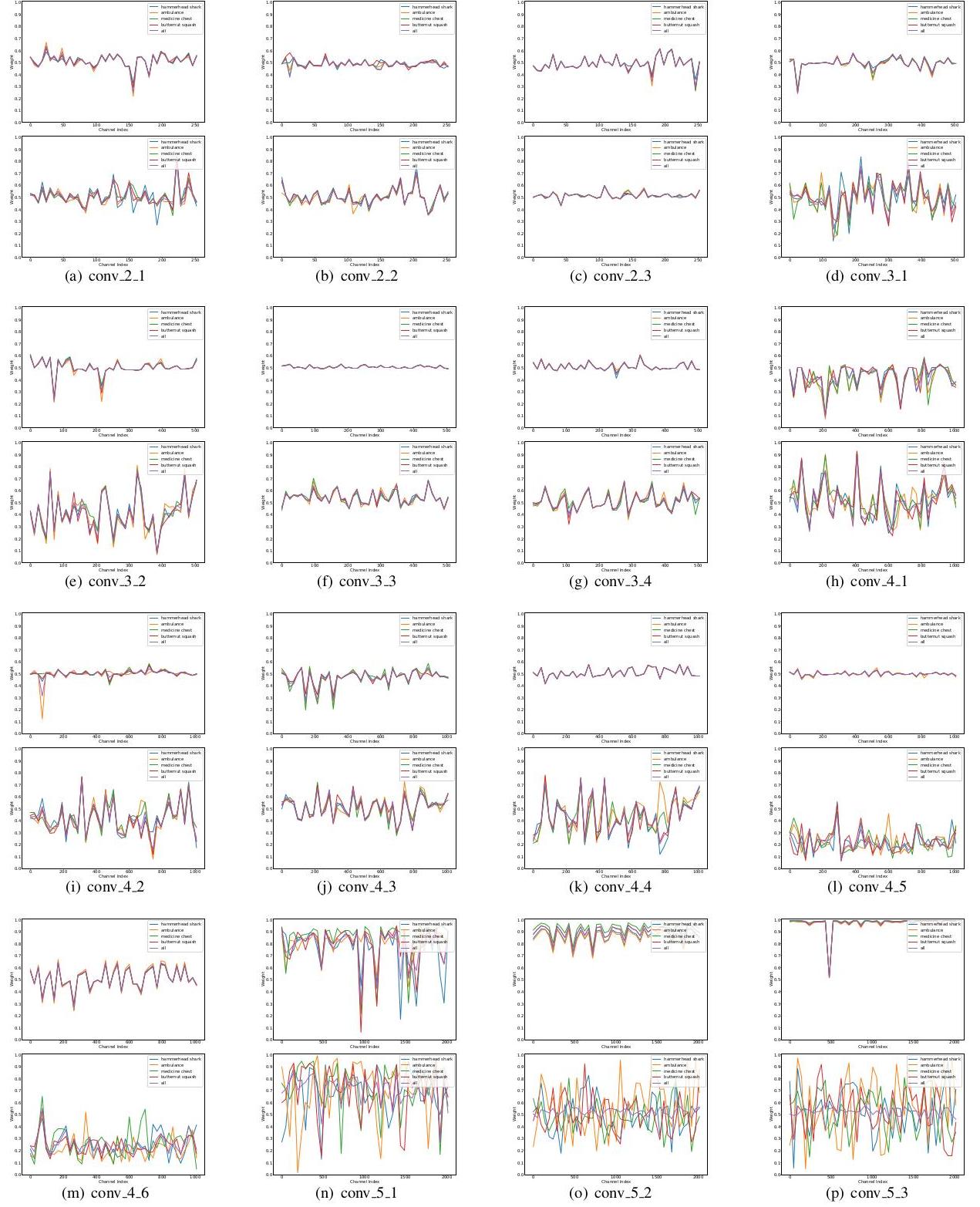


Figure 6. Visualization the channel weights of conv\_ \_ , where indicate -th stage and is -th convolution block in -th stage. The channel weights learned by ECA modules and SE blocks are illustrated in bottom and top of each row, respectively. Better view with zooming in.

图6。可视化conv\_ \_ 的通道权重，其中 表示第 阶段， 是第 阶段的第 个卷积块。ECA模块和SE块学习到的通道权重分别展示在每行的底部和顶部。放大查看效果更佳。