# -Nets：双注意力网络

云鹏 陈\*

新加坡国立大学

chenyunpeng@u.nus.edu

Yannis Kalantidis

Facebook 研究院

yannisk@fb.com

李 建书

新加坡国立大学

jianshu@u.nus.edu

严 水成

360 AI 研究院

新加坡国立大学

eleyans@nus.edu.sg

冯 建石

新加坡国立大学

elefjia@nus.edu.sg

# 摘要

学习捕获长距离关系对于图像/视频识别至关重要。现有的CNN模型通常依赖于增加深度来建模这样的关系，这是非常低效的。在这项工作中，我们提出了“双注意力模块”，这是一种新颖的组件，它从输入图像/视频的整个时空空间聚集和传播信息全局特征，使得后续的卷积层能够有效地访问整个空间中的特征。该组件设计有两个步骤的双注意力机制，第一步通过二阶注意力池化将整个空间中的特征聚集到一个紧凑集合中，第二步通过另一个注意力机制自适应选择并将特征分布到每个位置。所提出的双注意力模块易于采用，可以方便地插入到现有的深度神经网络中。我们在图像和视频识别任务上进行了广泛的消融研究和实验，以评估其性能。在图像识别任务中，配备了我们双注意力模块的ResNet-50在ImageNet-1k数据集上超过了参数数量和FLOPs都更多的ResNet-152架构。在动作识别任务上，我们提出的模型在Kinetics和UCF-101数据集上取得了比近期工作显著更高效率的最先进结果。

# 1 引言

深度卷积神经网络（CNNs）在过去几年中已成功应用于图像和视频理解。许多新的网络拓扑结构已被开发出来以减轻优化困难 并增加学习容量 ，这对图像 和视频 [23] 的识别性能产生了显著的提升。

然而，卷积神经网络（CNNs）由于其卷积操作符的限制而天生具有局限性，这些操作符专门用于捕捉局部特征和关系，例如来自 区域，而在建模长距离依赖性方面效率低下。尽管堆叠多个卷积操作符可以扩大感受野，但在实际应用中也会带来一系列不利问题。首先，堆叠多个操作符使得模型不必要地加深加大，导致更高的计算和内存成本以及增加过拟合的风险。其次，远离特定位置的特征必须通过多层堆叠才能在正向传播和反向传播中影响该位置，增加了训练过程中的优化难度。第三，远距离位置可见的特征实际上是来自后面几层的“延迟”特征，导致推理效率低下。尽管一些最近的研究工作 可以部分缓解上述问题，但它们要么灵活性不足 [11]，要么计算成本高昂 [25]。

\*这部分工作是在Facebook研究实习期间完成的。

在这项工作中，我们通过引入一种新的网络组件来克服这些限制，该组件使得卷积层能够立即感知到相邻层的整个时空空间 。核心思想是首先从整个空间中收集关键特征到一个紧凑集合中，然后自适应地将它们分布到每个位置，以便后续的卷积层即使没有大的感受野也能感知到整个空间中的特征。我们为此目的开发了一个通用函数，并使用高效的双重注意力机制来实现它。第一种二阶注意力池化操作选择性地从整个空间中收集关键特征，而第二种采用另一种注意力机制来自适应地分配有助于补充每个时空位置的高级任务的关键特征子集。我们将我们提出的双重注意力块称为 -block，其生成的网络称为 -Net。

双重注意力模块与近期许多工作相关，包括 Squeeze-and-Excitation Networks [11]、协方差池化 [14]、非局部神经网络 [25] 以及 [24] 的 Transformer 架构。然而，与这些现有工作相比，它具有几个独特的优势：其第一次注意力操作隐式计算了池化特征的第二阶统计量，能够捕获 SENet [11] 中使用的全局平均池化无法捕获的复杂外观和运动相关性。其第二次注意力操作自适应地从紧凑的包中分配特征，这比 中将所有位置的特征与每个特定位置进行详尽相关性分析要高效。在图像和视频识别任务上的大量实验清楚地验证了我们提出方法的上述优势。

我们将我们的贡献总结如下：

* 我们提出了一种通用的公式，用于通过普遍的聚集和分配函数捕获长距离特征相互依赖性。
* 我们提出了双重注意力模块，用于收集和分配长距离特征，这是一种高效的架构，能够捕获第二阶特征统计量并进行自适应特征分配。该模块能够在低计算和内存占用的情况下建模长距离相互依赖性，同时显著提升图像/视频识别性能。
* 我们通过大量的消融研究探讨了我们所提出的 -Net 的效果，并通过与图像识别和视频动作识别任务（包括 ImageNet-1k、Kinetics 和 UCF-101）的多个公共基准上的现有技术水平进行比较，证明了其优越性能。

本文的其余部分组织如下。我们首先在第 2 节中激发并介绍我们的方法，在那里我们还讨论了我们的方法与近期工作的关系。然后我们在第 3 节中评估并报告结果，并在第 4 节中结束本文。

# 2 方法

卷积运算符被设计为关注局部邻域，因此无法“感知”整个空间和/或时间空间，例如整个输入帧或跨多个帧的某一位置。因此，CNN模型通常会采用多个卷积层（或循环单元 ）以捕捉输入的全局特征。同时，自注意力机制和相关性运算符如二阶池化最近已被证明在广泛的任务中表现良好 [24, 14, 15]。在本节中，我们提出了一个能够收集并分配全局特征到输入的每个空间时间位置的组件，帮助后续的卷积层立即感知整个空间并捕捉复杂关系。我们首先通过提供一种通用公式正式描述了这个期望组件，然后介绍了我们的双重注意力块，这是此类组件的一种高效实例化。最后，我们讨论了我们的方法与其他近期相关方法的关联。

在这里，“空间”指的是输入帧的整个特征图和视频序列的完整空间时间特征。

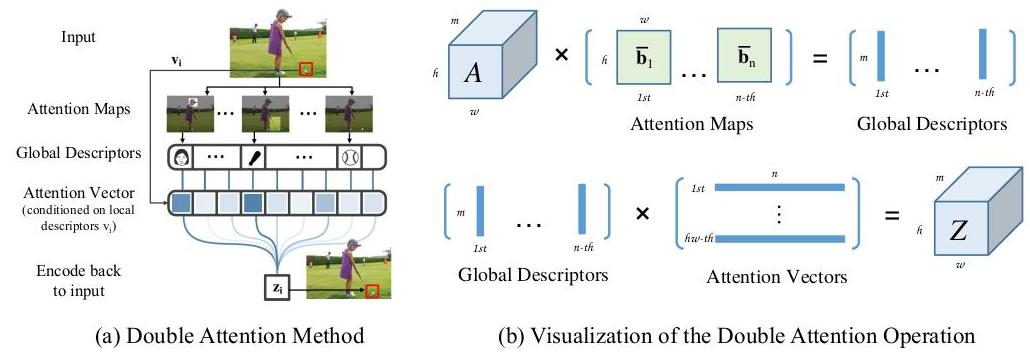


图1：双重注意力机制的说明。（a）一个解释我们双重注意力方法思想的单一帧输入示例，其中全局特征集只计算一次，然后由所有位置共享。同时，每个位置 将根据其局部特征 的需求生成自己的注意力向量，以选择有助于补充当前位置并形成特征 的期望全局特征子集。（b）在三维输入数组 上的双重注意力操作。第一个注意力步骤显示在顶部，并产生一组全局特征。在位置 ，第二个注意力步骤生成了新的局部特征 ，如下所示。

令 表示一个时空（3D）卷积层的输入张量，其中 表示通道数， 表示时间维度， 和 是输入帧的空间维度。对于每一个时空输入位置 及其局部特征 ，我们定义

为一个操作符的输出，该操作符首先在整个空间内收集特征，然后将其重新分配到每个输入位置 ，同时考虑该位置的局部特征 。具体来说， 自适应地聚合整个输入空间的特征，而 则根据局部特征向量 的条件，将收集到的信息分配到每个位置 。

收集和分配信息这一想法受到了挤压-激励网络（SENet）[11]的启发。然而，方程（1）以一种更一般的形式呈现了这一想法，从而带来了一些有趣的见解和优化。在[11]中，全局平均池化被用于收集过程，而得到的单个全局特征被分配到所有位置，忽略了不同位置的不同需求。鉴于这些不足，我们引入了这种基因表述，并提出了双重注意力块，其中全局信息首先通过二阶注意力池化（而不是一阶平均池化）进行收集，收集到的全局特征根据当前局部特征 的需求，通过第二次注意力机制自适应地分配。这样，一组紧凑的特征可以捕捉更复杂的全局关系，每个位置都可以接收到与其现有局部特征互补的定制化全局信息，从而促进学习更复杂的关系。所提出组件的示意图如图1（a）所示。下面，我们首先详细描述其架构，然后讨论一些实例化及其与其他近期相关方法的联系。

# 2.1 第一步关注：特征收集

最近的一项工作 [15] 使用双线性池化来捕获特征的二阶统计量并生成全局表示。与仅计算一阶统计量的传统平均池化和最大池化相比，双线性池化可以更好地捕获和保留复杂关系。具体来说，双线性池化给出了来自两个输入特征图 和 中所有特征向量对 的外积的二阶特征的求和池化：

对于空间（2D）卷积，即当输入是图像时， 。

其中 和 。在卷积神经网络（CNNs）中， 和 可以是来自同一层的特征图，即 ，或者是来自两层不同的特征图，即 和 ，并具有参数 和 。

通过引入双线性池化的输出变量 并将第二个特征 重写为 ，其中每个 是一个 维的行向量，我们可以将方程（2）重写为

方程（3）为双线性池化结果提供了一个新的视角：输出不仅仅是计算二阶统计量，双线性池化的输出 实际上是一个视觉原语的集合，其中每个原语 是通过收集由 加权的局部特征计算得出的。这激发我们开发了一种新的基于关注的特征收集操作。我们进一步在 上应用softmax以确保 ，即一个有效的关注加权向量，从而得到以下二阶关注池化过程：

图1(b)中的第一行显示了与公式(4)相对应的二阶注意力池化，其中 和 是两个不同卷积层对输入 进行转换的输出。在实现中，我们令 和 。二阶注意力池化为收集关键特征提供了一种有效方式：当 在所有位置上密集关注时，它能够捕捉全局特征，例如纹理和光照；而当 在特定区域上稀疏关注时，它能够捕捉特定的语义存在，例如一个物体及其部分。我们注意到在文献[7]中也有类似的见解，其中他们提出了一种与全连接分类器相关联的双线性池操作的秩1近似。然而，在我们的工作中，我们提出使用注意力池化将不同位置上的视觉基元收集到一个全局描述符包中，使用softmax注意力图，并且不应用任何低秩约束。

# 2.2 第二步注意力：特征分布

在从整个空间中收集特征之后的下一步是将其分布到输入的每个位置，以便后续的卷积层即使使用小的卷积核也能感知到全局信息。

与SENet[11]将相同的总结全局特征分布到所有位置不同，我们提出根据每个位置特征 的需求来分布自适应的视觉基元包，以获得更大的灵活性。这样，每个位置可以选择与当前特征互补的特征，这可以使训练更容易，并帮助捕捉更复杂的关系。这是通过从 中选择一组特征向量并使用软注意力来实现的：

式（5）阐述了所提出的软注意力特征选择方法。在我们的实现中，我们应用softmax函数来归一化 使其成为单位总和的形式，这被发现能够提供更好的收敛性。图1（b）的第二行展示了上述特征选择步骤。与生成注意力图的方式类似，注意力权重向量集合也是通过卷积层后接一个softmax归一化器生成的，即 其中 包含该层的参数。

# 2.3 双重注意力块

我们将上述两个注意力步骤结合起来，形成我们提出的双重注意力块，其在深度神经网络中的计算图如图2所示。为了构建双重注意力操作，我们将式（4）和式（5）代入式（1）并得到

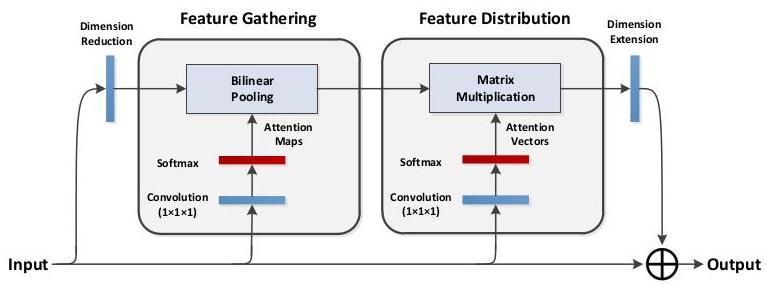


图2：所提出双重注意力块的计算图。所有卷积核的大小为 。我们将这个双重注意力块插入到现有的卷积神经网络中，例如残差网络 [9]，以形成 -Net。

图1（b）展示了组合的双重注意力操作，图2展示了相应的计算图，其中特征数组 和 是通过对输入特征数组 进行三个不同卷积层的操作并在必要时进行softmax归一化生成的。输出结果 通过进行两次矩阵乘法以及必要的重塑和转置操作得到。在这里，最后添加了一个额外的卷积层来增加输出 的通道数，以便它可以通过逐元素加法编码回输入 。在训练过程中，可以使用自动微分 和链式法则轻松计算损失函数的梯度。

实现方程（6）的计算图有两种不同的方式。一种是使用方程（6）中给出的左结合，其计算图如图2所示。另一种是进行右结合，如下所示：

我们注意到这两种不同的结合在数学上是等价的，因此将产生相同的输出。然而，它们的计算成本和内存消耗不同。方程（6）中“左结合”的第二次矩阵乘法的计算复杂度是 ，而方程（7）中的“右结合”具有 的复杂度。至于内存 ，存储第一次矩阵乘法结果的输出成本分别为左结合的 和右结合的 。在实际中，一个具有 帧和512通道大小的输入数据数组在采用右结合时很容易消耗超过 的内存，这比左结合的1MB成本要高得多。在这种情况下，左结合在计算效率上也比右结合更高。因此，在 的常见情况下，我们建议使用方程（6）中的左结合进行实现。

# 2.4 讨论

有趣的是观察到，等式（7）中带有右结合的实现可以通过最近的NL-Net [25]进一步解释，在该网络中，第一个乘法捕获局部特征之间的成对关系，并给出一个输出关系矩阵 。然后，将得到的关系矩阵应用于线性组合转换后的特征 以生成输出特征 。在成对关系函数的设计上明显存在差异，我们提出了一个新的关系函数，即 ，而不是使用嵌入式高斯公式 [24] 来捕获成对关系。同时，如上所述，任何此类方法实际上都会受到高计算和内存成本的影响，并且依赖于一些子采样技巧来降低成本，这可能会潜在地损害准确性。由于NL-Net是当前视频识别任务的最先进技术，并且与我们的研究紧密相关，因此我们在实验部分直接比较并详细讨论了两种方法的性能。结果清楚地表明，我们提出的方法不仅超越了NL-Net，而且在效率和准确性上都更胜一筹。由于我们在实验中比较的嵌入式高斯NL-Net公式在数学上等同于[24]中的自注意力公式，因此对NL-Net的结论/比较也适用于变压器网络。

所有值都存储在32位浮点数中。

表1：用于视频任务的三个骨干残差网络。ResNet-26和ResNet-29的输入尺寸为 ，而ResNet-50的输入尺寸为 。我们遵循[25]，并为ResNet-50的最后三个阶段设置 ，以减小时间尺寸，从而降低计算成本。

| stage | ResNet-26 | ResNet-29 | output│ | ResNet-50 | output |
| --- | --- | --- | --- | --- | --- |
| conv1 | , stride | , stride |  | , stride max pooling, stride |  |
| conv2 |  |  |  |  |  |
| conv3 |  |  |  |  |  |
| conv4 |  |  |  | ， |  |
| conv5 |  |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  | global average pool, fc, softmax | global average pool, fc, softmax |  | global average pool, fc, softmax |  |
| (#Params, FLOPs) | (7.0 M, 8.3 G) | (7.6 M, 9.2 G) | II | (33.4 M, 31.3 G) |  |

# 3 实验部分

在本节中，我们首先对提出的 -Nets 在 Kinetics [12] 视频识别数据集上进行了广泛的消融研究，并将其与最先进的 NL-Net [25] 进行比较。然后，我们在图像识别和视频识别任务上使用更深更宽的神经网络进行了更多实验，并将其与最先进的方法进行了比较。

# 3.1 实施细节

骨干卷积神经网络 我们在所有实验中使用了残差网络 [10] 作为我们的骨干卷积神经网络。表1展示了用于视频识别任务的骨干卷积神经网络的架构细节，其中我们使用 ResNet-26 进行所有消融研究，并将 ResNet-29 作为基线方法之一。计算成本是通过 FLOPs（即浮点乘加）来衡量的，模型复杂度是通过 #Params（即训练参数总数）来衡量的。ResNet-50 比 ResNet-26 几乎 更深更宽，因此仅在与其他最先进方法比较的最后几次实验中使用。对于图像识别任务，我们使用相同的 ResNet-50，但是为输入/输出数据和卷积核去除了时间维度。

训练和测试设置 我们使用 MXNet [3] 在图像分类任务上进行实验，并在视频分类任务上使用 PyTorch [18]。对于图像分类，我们报告了标准单一模型单一 中心裁剪验证准确率，遵循 。对于视频数据集的实验，我们报告了单个片段准确率和视频准确率。所有实验都是在分布式 K80 GPU 集群上进行的，网络通过同步的 SGD 进行优化。代码和训练模型将很快在 GitHub 上发布。

# 3.2 消融研究

对于在Kinetics [1]上的消融研究，我们每个实验使用32个GPU，总批量大小为512，从头开始训练。所有网络以16帧分辨率 作为输入。基础学习率设置为0.2，并在第 次迭代、第 次迭代时以0.1的因子减少，并在第 次迭代时终止。我们将三个卷积层的输出通道数 和 设置为输入通道数的 倍。注意，为了公平比较，所有方法均未采用子采样技巧。

单块表格2显示了当仅在主干网络中添加一个额外块时的结果。该块放置在某个阶段的第二个残差单元之后。从最后三行可以看出，我们提出的 -块与基线ResNet-26和更深的ResNet-29相比，性能持续提升。值得注意的是，额外的成本非常小。我们还发现，在顶层放置 -块比在低层放置带来的性能提升更显著。这可能是因为顶层提供了更适合提取全局视觉基元的语义抽象表示。相比之下，非局部网络 [25] 相比我们的方法，准确度增益较小，计算成本更高。由于非局部网络在底层阶段的计算成本呈二次增长，当块放置在Conv2时，我们甚至无法完成训练。

表2：在Kinetics数据集上单个非局部块 [25] 与单个双重注意力块的比较。顶部行显示了没有额外块的原始残差网络的表现。

| Model | + 1 Block | #Params | FLOPs | A FLOPs | Clip @1 |  | Video@1 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ResNet-26 | None | 7.043 M | 8.3 G |  | 50.4 % |  | 60.7 % |
| ResNet-29 | None | 7.620 M | 9.2 G | 900 M | 50.8 % | +0.5 % | 61.6 % |
| ResNet-26 + NL [25] | @ Conv2 | 7.061 M | 49.0 G | 40.69 G |  |  |  |
| @ Conv3 | 7.112 M | 13.7 G | 5.45 G | 51.5 % | +1.1 % | 62.0 % |
| @ Conv4 | 7.312 M | 9.3 G | 1.04 G | 51.7 % | +1.3 % | 62.3 % |
| ResNet-26 + | @ Conv2 | 7.061 M | 8.7 G | 463 M | 51.2 % | +0.8 % | 61.8 % |
| @ Conv3 | 7.112 M | 8.7 G | 463 M | 51.9 % | +1.5 % | 62.0 % |
| @ Conv4 | 7.312 M | 8.7 G | 463 M | 52.3 % | +1.9 % | 62.6 % |

表3：多个非局部块 [25] 与多个双重注意力块在Kinetics数据集上的性能比较。我们报告了所有方法的top-1视频片段准确性和top-1视频准确性。没有额外块的原始残差网络显示在顶部行。

| Model | +N Blocks | #Params | FLOPs | FLOPs | Clip @1 | Δ Clip@1 | Video @1 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ResNet-26 | None | 7.043 M | 8.3 G |  | 50.4 % |  | 60.7 % |
| ResNet-29 | None | 7.620 M | 9.2 G | 900 M | 50.8 % | +0.5 % | 61.6 % |
| ResNet-26 + NL [25] | 1 @ Conv4 | 7.312 M | 9.3 G | 1.04 G | 51.7 % | +1.3 % | 62.3 % |
| 2 @ Conv4 | 7.581 M | 10.4 G | 2.08 G | 52.0 % | +1.6 % | 62.9 % |
| 4 @ Conv3&4 | 7.719 M | 21.3 G | 12.97 G | 52.4 % | +2.0 % | 62.8 % |
| ResNet-26 + | 1 @ Conv4 | 7.312 M | 8.7 G | 463 M | 52.3 % | +1.9 % | 62.6 % |
| 2 @ Conv4 | 7.581 M | 9.2 G | 925 M | 52.5 % | +2.1 % | 63.1 % |
| 4 @ Conv3&4 | 7.719 M | 10.1 G | 1.85 G | 53.0 % | +2.6 % | 63.5 % |

多个块 表3显示了将多个块添加到主干网络时的性能增益。从结果中可以看出，我们提出的 -Net在添加更多块时单调提高准确性，并且与竞争对手相比，计算复杂度更低。我们还发现，与将所有块添加到同一阶段相比，将块添加到不同阶段可以带来更显著的准确性增益。

# 3.3 图像识别实验

我们在ImageNet-1k [13] 图像分类数据集上评估了提出的 -Net，该数据集包含超过120万张高分辨率图像，分为1,000个类别。我们的实现基于 [5] 发布的代码，使用 GPU，批量大小为2,048。基础学习率设置为 ，当训练准确度饱和时，学习率会以0.1的因子减少。

表4：在ImageNet-1k上的最新技术水平比较。

| Model | Backbone | Top-1 | Top-5 |
| --- | --- | --- | --- |
| ResNet [9] | ResNet-50 | 75.3 % | 92.2 % |
| ResNet-152 | 77.0 % | 93.3 % |
| SENet [11] | ResNet-50 | 76.7 % | 93.4 % |
| -Net | ResNet-50 | 77.0 % | 93.5 % |

表5：在Kinetics上的最新技术水平结果比较。仅使用RGB信息作为输入。

| Model | #Frames | FLOPs | Video Q1 | Video @5 |
| --- | --- | --- | --- | --- |
| ConvNet+LSTM [1] |  |  | 63.3 % |  |
| I3D [1] | 64 | 107.9 G | 71.1 % | 89.3 % |
| R(2+1)D [23] | 32 | 152.4 G | 72.0 % | 90.0 % |
| -Net | 8 | 40.8 G | 74.6 % | 91.5 % |

表6：在UCF-101上的最新技术水平结果比较。报告了三个训练/测试分割的平均Top-1视频准确性。

| Method | Backbone | FLOPs | Video @1 |
| --- | --- | --- | --- |
| C3D [21] | VGG | 38.5 G | 82.3 % |
| Res3D [22] | ResNet-18 | 19.3 G | 85.8 % |
| I3D-RGB [1] | Inception | 107.9 G | 95.6 % |
| R(2+1)D-RGB [23] | ResNet-34 | 152.4 G | 96.8 % |
| -Net | ResNet-50 | 41.6 G | 96.4 % |

如表4所示，配备了5个额外的 -blocks 在Conv3和Conv4上的ResNet-50性能超过了更大的ResNet-152架构。我们注意到嵌入 -blocks 的ResNet-50比ResNet-152效率高出 ，并且仅消耗6.5 GFLOPs和 参数。与SENet [11] 相比， -Net也实现了更好的准确性，这证明了所提出双重注意力机制的有效性。

# 3.4 视频识别实验结果

在本节中，我们评估了所提出的方法在视频表征学习上的表现。我们考虑了静态图像特征预训练但运动特征从头开始学习的场景，即在一个大规模的Kinetics [1] 数据集上训练模型，以及将训练良好的运动特征迁移到小规模的UCF-101 [20] 数据集的场景。

在Kinetics上从头开始学习运动特征 我们使用在ImageNet上预训练的ResNet-50并添加5个随机初始化的 -blocks来构建3D卷积网络。相应的主体网络如表1所示。网络以8帧（采样步长：8）作为输入，并使用 进行 次迭代训练，总批量大小为512。初始学习率设置为0.04，当训练准确度饱和时以步进方式减少。最终结果如表5所示。与最先进的I3D [1] 和R(2+1)D [23] 相比，我们提出的模型即使在较少的采样帧数下也显示出更高的准确性，这再次证实了所提出双重注意力机制的优越性。

将学习到的特征迁移到 UCF-101 数据集。UCF-101 包含大约 13,320 个来自 101 个动作类别的视频，并具有三种训练/测试划分。UCF-101 的训练集比 Kinetics 数据集小得多，我们用它来评估在 Kinetics 上预训练的模型学到的特征的通用性和鲁棒性。网络以基础学习率 0.01 进行训练，该学习率使用 GPU 以批大小为 104 个片段进行三次降低，每次降低系数为 0.1，并在单尺度上使用 输入分辨率进行测试。表 6 展示了我们所提出模型的成果以及与现有最佳技术的比较。与上述结果一致， -Net 在显著降低计算成本的情况下实现了领先性能。这表明 -Net 学到的特征具有鲁棒性，并且可以以非常低的成本有效地迁移到新数据集，与现有方法相比具有优势。

# 4 结论

在这项工作中，我们为深度 CNN 提出了一个双重注意力机制，以克服局部卷积操作的局限性。所提出的双重注意力方法有效地捕获全局信息，并以两步注意力方式将其分布到每个位置。我们很好地制定了所提出的方法，并将其实例化为一个轻量级块，可以轻松地插入到现有的 CNN 中，且计算开销很小。在包括 ImageNet-1k、Kinetics 和 UCF-101 在内的大量基准数据集上的广泛消融研究和实验，证实了所提出的 -Net 在 图像识别任务和 视频识别任务上的有效性。在未来，我们希望探索将双重注意力集成到最近的紧凑网络架构 中，以利用所提出方法的表现力，为更小、更适合移动的模型带来优势。

# 参考文献

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4724-4733. IEEE, 2017.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915, 2016.

[3] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274, 2015.

[4] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Multi-fiber networks for video recognition. In European Conference on Computer Vision (ECCV), 2018.

[5] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In Advances in Neural Information Processing Systems, pages 4470-4478, 2017.

[6] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2625-2634, 2015.

[7] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In Advances in Neural Information Processing Systems, pages 33-44, 2017.

[8] Ross Girshick. Fast r-cnn. arXiv preprint arXiv:1504.08083, 2015.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770-778, 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In European Conference on Computer Vision, pages 630-645. Springer, 2016.

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[12] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijaya-narasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017.

[13] 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.

[14] Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. Is second-order information helpful for large-scale visual recognition? arXiv preprint arXiv:1703.08050, 2017.

[15] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In Proceedings of the IEEE International Conference on Computer Vision, pages 1449-1457, 2015.

[16] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. arXiv preprint arXiv:1807.11164, 2018.

[17] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, pages 4694-4702. IEEE, 2015.

[18] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. Pytorch, 2017.

[19] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. arXiv preprint arXiv:1801.04381, 2018.

[20] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402, 2012.

[21] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In Computer Vision (ICCV), 2015 IEEE International Conference on, pages 4489-4497. IEEE, 2015.

[22] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. arXiv preprint arXiv:1708.05038, 2017.

[23] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, pages 6000-6010, 2017.

[25] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In Computer Vision and Pattern Recognition (CVPR), 2018.

[26] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5987-5995. IEEE, 2017.