

# 论文 Lite-HRNet: A Lightweight High-Resolution Network 复现

## 摘要

本报告旨在复现《Lite-HRNet: A Lightweight High-Resolution Network》论文中的高效高分辨率网络 Lite-HRNet。Lite-HRNet 作为一种轻量级的高分辨率神经网络，能够在保持高分辨率特征的同时减少计算成本，适用于移动设备等资源受限的环境。我们详细分析了 Lite-HRNet 的架构设计，并根据原论文实现了该模型，通过一系列实验验证其性能。实验结果表明，Lite-HRNet 在保持较低计算量的同时，仍能够在多个任务中达到竞争力的性能。

**关键词：**高效网络架构； Lite-HRNet

## 1. 引言

随着深度学习技术的快速发展，卷积神经网络（CNN）已成为计算机视觉领域中最重要工具之一。高分辨率网络（如 HRNet）通过保留高分辨率特征图，显著提高了模型的表现力和精度。然而，随着网络分辨率的提高，模型的计算复杂度和参数量也随之增加，导致了训练和推理阶段的高昂计算成本。这对于资源受限的设备（如移动设备、嵌入式系统等）提出了挑战。因此，如何在保证高分辨率特征学习的同时，降低计算开销，成为了当前视觉领域亟待解决的问题。

Lite-HRNet 应运而生，它结合了高分辨率网络和轻量级网络的优点，提出了一种高效的网络架构。Lite-HRNet 通过创新的设计在减少计算资源消耗的同时，保持了良好的视觉表现能力。这一方案为在资源受限的设备上部署深度学习模型提供了一个新的解决思路，并有望推动计算机视觉技术在实际应用中的广泛应用。

通过复现和分析 Lite-HRNet，我深入理解了高分辨率网络在轻量化设计方面的创新，验证其在实际任务中的表现，并为进一步优化和应用这一技术提供参考。Lite-HRNet 不仅在传统的高分辨率网络架构上进行了重要的改进，还在计算效率和应用场景的适配上做出了贡献，为深度学习模型在移动设备、嵌入式平台等计算资源有限的环境中的应用打开了新的可能性。因此，研究 Lite-HRNet 具有重要的学术价值和实际意义。

## 2. 相关工作

### 2.1. 高分辨率网络（HRNet）

高分辨率网络（HRNet）作为近年来在视觉任务中取得重要进展的网络架构，因其能够保留高分辨率特征而广泛应用于目标检测、图像分割、人体姿态估计等任务。HRNet 的核心思想是

通过多分辨率特征的并行处理与融合，最大化地保留不同尺度的信息，从而提高模型的感知能力和精度。其虽然能保持高分辨率特征，增强了低级特征的表达能力，并且通过多分辨率特征融合，有效解决分辨率与感受野之间的平衡问题，但是由于需要大量的计算资源和内存，HRNet难以直接应用于资源受限的设备，如移动端、嵌入式设备等。

## 2.2. 轻量级网络架构

为了解决高分辨率网络在计算资源上的消耗，许多研究集中于设计轻量级神经网络。轻量级网络架构通过减少网络的参数量、优化计算操作（如深度可分离卷积、组卷积等），显著降低了模型的计算成本，尤其适用于低计算能力的设备。目前主要有的一些解决办法主要有 MobileNet 提出了深度可分离卷积，显著减少了卷积操作的计算量，成为轻量级网络设计的重要参考<sup>[1]</sup>。ShuffleNet<sup>[2]</sup>通过组卷积和通道重排技术进一步减少了计算复杂度，具有更高的效率。EfficientNet: 通过神经架构搜索（NAS）方法，自动优化网络结构，获得了良好的计算效率和性能平衡。虽然这些方法能够有效减少计算量，但在精度和性能上可能会存在一定的妥协，尤其是在高分辨率图像处理任务中。

## 2.3. 高效卷积操作与特征融合

为了提高计算效率，许多研究者提出了各种高效的卷积操作和特征融合方法。这些方法能够在降低计算开销的同时，保留或提升模型的性能。例如深度可分离卷积（Depthwise Separable Convolution）：将传统卷积分解为深度卷积和逐点卷积，有效减少了计算量。通道重排（Channel Shuffling）：通过重新排列通道顺序，提升了信息流动性，并减少了计算复杂度。多尺度特征融合：通过并行多尺度处理，融合来自不同分辨率的特征，能够在保证精度的同时提高计算效率。

# 3. 本文方法

## 3.1. Naive Lite-HRNet 构造

### 3.1.1. Shuffle blocks 流程

Shuffle block 的流程如图 1 所示，该流程首先会将通道分成两个分区。一个分区不做处理。一个分区经过一个（ $1 \times 1$  卷积、 $3 \times 3$  深度卷积和  $1 \times 1$  卷积）序列，然后其输出与另一个分区连接。最后，串接的通道被 shuffled。

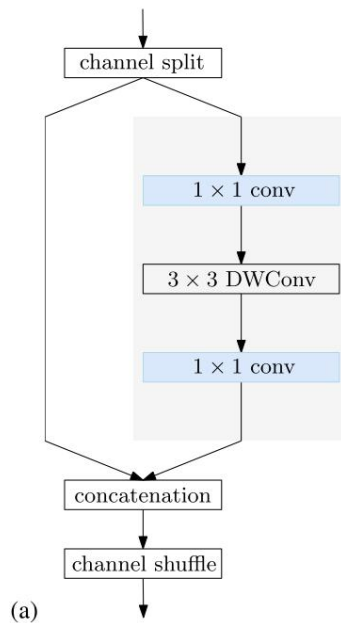


图 1 Shuffle blocks 流程图

### 3.1.2. HRNet 定义

HRNet 是从一个高分辨率卷积卷积作为第一阶段开始，逐步添加一个高到低分辨率的流作为新的阶段。多分辨率并行连接。主体由一些列的阶段组成。在每个阶段，跨分辨率的信息被反复交换。基础版 Lite-HRNet 遵循 Small HRNet 的设计，使用更少的层和更小的宽度来设计。

如图 2 所示,Small HRNet 的主干由两个步长为 2 的  $3 \times 3$  卷积组成，主体中的每个阶段包含一系列残差块和一个多分辨率融合。

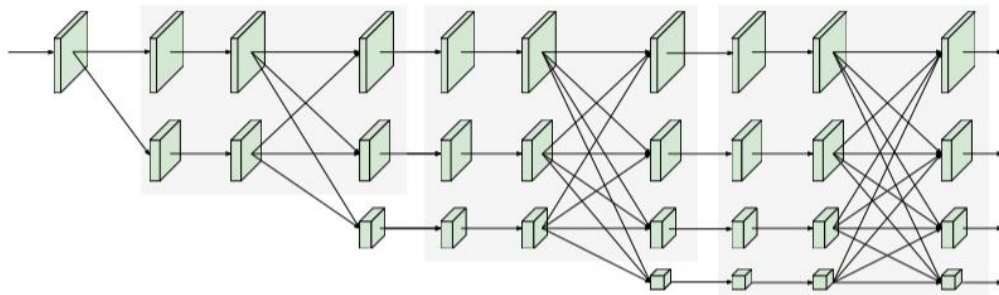


Figure 2. **Illustration of the Small HRNet architecture.** It consists of a high-resolution stem as the first stage, gradually adding high-to-low resolution streams as the main body. The main body has a sequence of stages, each containing parallel multi-resolution streams and repeated multi-resolution fusions. The details are given in Section 3.

图 2 Small HRNet

### 3.1.3. 简单结合

简单将 shuffle 块替换 Small HRNet 的主干中的第二个  $3 \times 3$  卷积和替换掉所有正常的残差（由两个  $3 \times 3$  卷积组成）。将多分辨率融合中的正常卷积替换为可分离卷积，得到 Naive Lite-HRNet。

### 3.2. Lite-HRNet 构造

它和基础版的核心区别是将条件通道加权单元代替 shuffle block 中高代价的  $1 \times 1$  卷积，并跨通道和分辨率计算权重。

#### 3.2.1. $1 \times 1$ 卷积代价高昂

$1 \times 1$  卷积虽然在跨通道交换信息方面起着关键作用，但在 shuffle 块中，两个  $1 \times 1$  卷积的复杂度比深度卷积的复杂度高很多。图 3 给出了  $1 \times 1$  之间复杂度比较的例子卷积和深度卷积，在表中可以看到  $3 \times 3$  深度卷积执行的浮点数运算次数明显小于  $1 \times 1$  卷积

Table 2. **Computational complexity comparison:  $1 \times 1$  convolution vs. conditional channel weight.**  $X_s \in \mathcal{R}^{H_s \times W_s \times C_s}$  are the input channel maps for the  $s$  resolution,  $X_1$  corresponds to the highest resolution.  $N_s = H_s W_s$ . For example, the shape of  $X_1$  and  $X_2$  are  $64 \times 64 \times 40$  and  $32 \times 32 \times 80$ , respectively. single/cross-resolution=single/cross resolution information exchange.

model	single-resolution	cross-resolution	Theory Complexity	Example FLOPs
$1 \times 1$ convolution	✓		$\sum_1^s N_s C_s^2$	12.5M
$3 \times 3$ depthwise convolution			$\sum_1^s 9 N_s C_s$	2.1M
CCW w/ spatial weights	✓		$\sum_1^s (2C_s^2 + N_s C_s)$	0.25M
CCW w/ multi-resolution weights		✓	$2(\sum_1^s C_s)^2 + \sum_1^s N_s C_s$	0.26M
CCW	✓	✓	$2(\sum_1^s C_s)^2 + 2 \sum_1^s (C_s^2 + N_s C_s)$	0.51M

图 3

#### 3.2.2. 条件信道加权

条件信道加权单元是用来替换基础版 Lite-HRNet 中的  $1 \times 1$  卷积的。第  $s$  个分辨率分支的逐元素加权运算写为：

$$Y_s = W_s \odot X_s$$

其中  $W_s$  是权重映射，大小为  $W_s \times H_s \times C_s$  的三维张量， $\odot$  是逐元素乘法算子。

复杂度与通道数  $\Theta(C)$  呈线性关系，远低于 shuffle 块中的  $1 \times 1$  卷积。并且通过使用单一分辨率的通道和所有分辨率的通道来计算权重，也表明权重起到了跨通道和分辨率交换信息的作用。如图 4 所示，虚线指示来自其他分辨率的表示和分配给其他分辨率的权重。H=交叉分辨率加函数。F=空间加权函数。

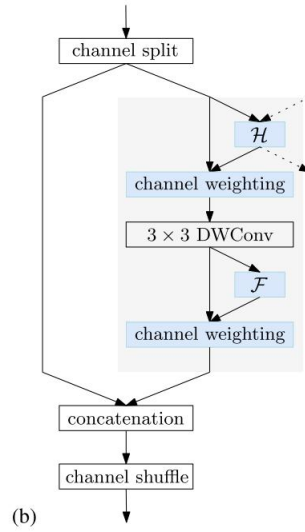


图 4

### 3.2.3. 跨分辨率权重计算

跨分辨率权重计算是一种利用不同分辨率特征图并动态加权的技术，能够有效地融合多尺度信息，从而提升深度学习模型在多尺度任务中的表现。这种方法不仅能够在多种计算机视觉任务中提升精度，还能通过计算效率的优化，使得高分辨率信息和低分辨率信息能够共同贡献于最终结果，从而改善整体性能。

文章实现了一个轻量级函数  $H_s(\cdot)$ ，首先在  $\{X_1, X_2, \dots, X_{s-1}\}$  上执行自适应平均池 AAP,  $X'_1 = AAP(X_1), X'_2 = AAP(X_2), \dots, X'_{s-1} = AAP(X_{s-1})$ ，其中 AAP 将任何输入大小池化为给定的输出大小  $W_s \times H_s$ ，然后将  $\{X'_1, X'_2, \dots, X'_{s-1}\}$  和  $X_s$  连接到一起，接着是  $1 \times 1$  卷积，ReLU,  $1 \times 1$  卷积和 sigmoid 的序列，生成由  $s$  个分区组成的权重图<sup>[3]</sup>。

### 3.2.4. 空间权重计算

Lite-HRNet 的网络设计使用了轻量级的结构来处理高分辨率图像，并通过特定的空间注意力机制来加权和融合不同分辨率的特征图。这些特征图中的每个空间位置都会根据其最终任务（如关键点检测）的贡献程度，分配不同的权重。这种设计能够在降低计算成本的同时，保留关键信息，从而提升模型在人体姿态估计等任务中的精度。

文章首先通过深度卷积网络生成不同分辨率的特征图。这些特征图包含了从低级到高级的不同层次的信息。

然后采用了一种有效的空间注意力机制，用于动态调整每个空间位置的重要性。空间注意力模块生成一个权重图，表示不同位置在高分辨率和低分辨率特征图中的重要性。通过这种方式，网络能够聚焦于更关键区域，提高关键点的定位精度。

最后根据空间注意力计算的权重，Lite-HRNet 对特征图进行加权融合。这不仅优化了计算资源的利用，还增强了模型对高分辨率区域的关注，从而改善了精度和鲁棒性。

## 4. 复现细节

### 4.1. 与已有开源代码对比

在这次的复现主要基于开源的部分代码的基础上，探究其改进的可能性。

## 4.2. 实验环境搭建

- Linux (Windows is not officially supported)
- Python 3.6+
- PyTorch 1.3+
- CUDA 9.2+ (If you build PyTorch from source, CUDA 9.0 is also compatible)
- GCC 5+
- mmcv (Please install the latest version of mmcv-full)
- Numpy
- cv2
- json\_tricks
- xtcocotools

## 4.3. 实验运行流程

开始实验之前，首先需要准备训练和测试数据集。Lite-HRNet 通常用于姿态估计任务，因此最常用的数据集包括：COCO (Common Objects in Context) 数据集：这个数据集包含了丰富的姿态标注，适用于多种视觉任务，包括人体姿态估计。MPII Human Pose 数据集：专注于人体姿态估计的标准数据集，具有高质量的标注。

然后进行数据预处理过程，对数据进行标准化处理，确保每张图片的输入尺寸统一（例如，384x288），并进行数据增强操作（如随机裁剪、翻转、旋转等）以提高模型的泛化能力。

接着训练模型。在 Lite-HRNet 中，训练过程中的参数（如学习率、批次大小、优化器等）由配置文件控制。根据实验需求，可以修改这些配置，batch size 根据 GPU 内存大小来选择合适的批次大小。在配置好所有超参数后，执行训练命令开始模型训练。训练流程包括：数据加载通过 `torch.utils.data.DataLoader` 加载训练数据。前向传播将数据输入到 Lite-HRNet 网络中，计算每个图像的预测结果。计算损失使用定义的损失函数（如关键点回归损失）来计算预测结果与真实标注之间的差异。反向传播通过反向传播算法计算梯度并更新网络的权重。验证每隔一定的 epoch，使用验证集进行验证，计算验证损失，并进行精度评估。训练过程中，输出包括训练损失、验证精度等信息，这有助于判断训练是否收敛，如下图 5 所示。

```
2024-11-28 16:37:17,893 - mmpose - INFO - Epoch [1][150/2340] lr: 5.974e-04, eta: 4 days, 9:53:04, time: 0.331, data_time: 0.002, memory: 5827, mse_loss: 0.0022, acc_pose: 0.1610, loss: 0.0022
2024-11-28 16:37:34,741 - mmpose - INFO - Epoch [1][200/2340] lr: 7.972e-04, eta: 3 days, 18:53:42, time: 0.337, data_time: 0.002, memory: 5827, mse_loss: 0.0021, acc_pose: 0.1766, loss: 0.0021
2024-11-28 16:37:51,617 - mmpose - INFO - Epoch [1][250/2340] lr: 9.970e-04, eta: 3 days, 9:55:21, time: 0.338, data_time: 0.003, memory: 5827, mse_loss: 0.0021, acc_pose: 0.1865, loss: 0.0021
2024-11-28 16:38:08,189 - mmpose - INFO - Epoch [1][300/2340] lr: 1.197e-03, eta: 3 days, 3:47:56, time: 0.332, data_time: 0.002, memory: 5827, mse_loss: 0.0021, acc_pose: 0.1914, loss: 0.0021
2024-11-28 16:38:25,381 - mmpose - INFO - Epoch [1][350/2340] lr: 1.397e-03, eta: 2 days, 23:39:46, time: 0.344, data_time: 0.002, memory: 5827, mse_loss: 0.0021, acc_pose: 0.2007, loss: 0.0021
```

图 5 训练过程

在训练完成后，保存模型的参数文件。通常保存最佳验证精度时的模型参数：

```
python tools/eval.py --cfg configs/coco_384x288.yaml --load_model path_to_model_checkpoint.
```

完成训练后，可以对模型进行评估和推理，得到最终的实验结果。

## 5. 实验结果分析

本次复现工作因为资源有限，所以我使用的是 mpii 数据集，实验结果对比如下表。

表 1

Method	Input size	#param	Gflops	PCKh
HRNet-W32	256*256	28.5M	7.69	90.0
Small HRNet-W16	256*256	1.3M	0.72	80.2
Lite-HRNet18	256*256	1.1M	0.27	86.1
Mine	256*256	1.13M	0.29	86.1

可以看到我的实验结果与原文结果是差不多的，并且分析得出在 mpII 数据集上的实验性能对比，相同条件下，Lite-HRNet 的平均精度指标是明显优于其他轻量化方案的。虽然相比其他大网络，性能方面有差异，但 Lite-HRNet 有着超低的浮点数运算次数与参数量。

## 6. 总结与展望

在复现过程中我深刻理解 Lite-HRNet 的设计与创新，通过复现 Lite-HRNet，我不仅掌握了该模型的架构，还深入理解了其创新设计（如高分辨率特征融合、轻量化模块设计等），这些技术对于提高模型精度与效率至关重要。并且对深度学习中的高效计算方法、轻量级网络的设计思想有了更深入的认识。虽然在复现过程中，我遇到了很多困难，例如刚开始的环境配置就把我困住了许久，但是我通过这次复现，掌握了如何高效地复现论文中的方法，还在环境配置、代码实现、调试等方面积累了宝贵的经验。这些技能不仅限于单个项目，而是对未来进行深度学习模型优化、调试和部署等任务提供了基础。

复现 Lite-HRNet 让我接触到了当前计算机视觉领域中的一种先进模型，未来可以基于这次复现的经验，进一步研究其他领域中的深度学习模型，如目标检测、语义分割等。通过复现更多的经典模型，我能够更好地理解并掌握最新的研究进展。在未来的发展中，也可以尝试将深度学习与其他学科结合，如医学、金融、物联网等。特别是在医疗领域，深度学习模型的应用正在快速发展。

## 参考文献

- [1] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv, abs/1704.04861.
- [2] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2017). ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6848-6856.
- [3] Yu, C., Xiao, B., Gao, C., Yuan, L., Zhang, L., Sang, N., & Wang, J. (2021). Lite-HRNet: A Lightweight High-Resolution Network. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10435-10445.