Electronic Research Archive

ERA, 32(2): 1145–1159.
DOI: 10.3934/era.2024055
Received: 16 October 2023
Revised: 27 December 2023
Accepted: 11 January 2024
Published: 29 January 2024

http://www.aimspress.com/journal/ERA

Research article

Lightweight high-performance pose recognition network: HR-LiteNet

Zhiming Cai[1,2,*], Liping Zhuang[1], Jin Chen[1] and Jinhua Jiang[1]

[1] School of Electronics, Electrical Engineering and Physics, Fujian University of Technology, Fuzhou 350118, China

[2] National Demonstration Center for Experimental Electronic Information and Electrical Technology Education, Fujian University of Technology, Fuzhou 350118, China

* **Correspondence:** Email: caizm@fjut.edu.cn.

Abstract: To address the limited resources of mobile devices and embedded platforms, we propose a lightweight pose recognition network named HR-LiteNet. Built upon a high-resolution architecture, the network incorporates depthwise separable convolutions, Ghost modules, and the Convolutional Block Attention Module to construct L_block and L_basic modules, aiming to reduce network parameters and computational complexity while maintaining high accuracy. Experimental results demonstrate that on the MPII validation dataset, HR-LiteNet achieves an accuracy of 83.643% while reducing the parameter count by approximately 26.58 M and lowering computational complexity by 8.04 GFLOPs compared to the HRNet network. Moreover, HR-LiteNet outperforms other lightweight models in terms of parameter count and computational requirements while maintaining high accuracy. This design provides a novel solution for pose recognition in resource-constrained environments, striking a balance between accuracy and lightweight demands.

Keywords: human pose estimation; high-resolution network; Ghost; lightweight module

1. Introduction

Human Pose Estimation, as a critical task in the field of computer vision, aims to accurately capture and analyze the posture and joint information of the human body from images or videos. This task finds widespread applications across various domains, including human-computer interaction, motion analysis, rehabilitation training, and virtual reality [1–3].

In recent years, the rapid development of deep learning technologies has significantly improved the accuracy of deep learning models in tasks such as image classification, object detection, and human pose estimation. Simultaneously, it has achieved remarkable success in various fields, including biomedicine [4,5], natural language processing, sports and fitness. With the rapid advancement of deep learning technologies, especially the rapid development of Convolutional Neural Networks (CNNs) [6], significant breakthroughs have been made in human pose estimation, gradually transitioning from traditional methods based on handcrafted features to approaches built upon deep neural networks [7,8]. In 2014, Toshev et al. [9] introduced DeepPose, representing a pioneering attempt to apply deep learning to human pose estimation. Innovatively, it transformed the keypoint detection problem into a regression problem. The core of the network, based on AlexNet, directly regressed the coordinates of human body keypoints from images, outperforming other methods at the time and laying the foundation for subsequent developments in deep learning for pose estimation. Tompson et al. [10] observed that directly regressing the coordinates of human body keypoints could lead to overfitting. To address this issue, they introduced an approach based on Gaussian-distributed keypoint heatmaps as an alternative to keypoint coordinate maps. This method significantly improved the network's training efficiency and accuracy. Currently, Gaussian heatmaps remain a mainstream encoding method for keypoints in deep CNN-based human pose estimation. Wei et al. [11] proposed a model called the Convolution Pose Machine (CPM). This method employed large convolutional kernels to capture contextual spatial relationships, and it mitigated the vanishing gradient problem through a multi-stage and intermediate supervision training strategy, positively impacting network performance.

In multi-person human pose estimation, algorithms can be categorized into two processing paradigms: Top-down and bottom-up [12,13]. Top-down methods begin by detecting the entire human body within the image and subsequently perform keypoint detection for each detected individual. The pose model proposed by Papandreou et al. [14] consists of two stages: the first stage detects human bodies using Faster R-CNN [15], while the second stage utilizes a fully convolutional ResNet [16] to predict heatmaps and offsets for each keypoint. They introduced a novel fusion process to achieve highly precise keypoint predictions. The High-Resolution Network (HRNet) model, proposed by Sun et al. [17], is a typical example of a top-down approach. HRNet employs a parallel connection structure, starting with a high-resolution subnetwork as the first stage and gradually introducing parallel subnetworks from high to low resolution in subsequent stages. This strategy maintains high-resolution features while performing multiple multi-scale fusions, significantly enhancing the accuracy of human pose estimations. Bottom-up methods, on the other hand, directly detect all possible keypoints within the image, cluster these keypoints using algorithms, and finally connect different keypoints of different individuals to detect distinct individuals. Pishchulin et al. [18] introduced a rapid body part detector called DeepCut, one of the earliest two-step bottom-up approaches. It initially detects all candidate body parts, labels each part, and uses Integer Linear Programming (ILP) to combine these parts into the final pose. The OpenPose model, proposed by Cao et al. [19], is a representative bottom-up algorithm. This model first predicts keypoint heatmaps using the CPM, introducing Part Affinity Fields (PAFs) to represent joint position and orientation information. By processing the detected keypoints and the areas of joint connections, this model rapidly associates these keypoints with different individuals.

Most of the aforementioned researchers have primarily focused on improving the accuracy of human pose algorithms, often neglecting the model parameters and computational requirements. However, in practical applications, due to computational and memory constraints, deploying large-scale network algorithms on ordinary devices is often infeasible. Therefore, reducing model parameters and computational demands is crucial for practical applications [20,21]. Currently, some scholars are gradually conducting systematic research on the lightweighting of human pose estimation [22].

In this context, we introduce a lightweight pose recognition approach based on a HRNet framework. This approach proposes L_Block and L_Basic modules that incorporate technologies such as depthwise separable convolutions and Ghost modules. While maintaining a certain level of pose estimation accuracy, this innovative lightweight design significantly reduces the network's computational complexity and parameter count. This novel lightweight design offers feasibility for deploying human pose estimation algorithms on resource-constrained devices.

The major contributions of this paper are as follows:

(1) Introducing a novel L_Block module and L_Basic module that ingeniously integrate lightweight technologies such as depth-wise separable convolutions and Ghost modules. The innovative design of these modules not only effectively reduces computational complexity but also enhances the model's perception of intricate features in human pose recognition.

(2) Employing innovative technical strategies, the paper seamlessly combines lightweight techniques with multi-branch parallelism and multi-scale feature fusion. This integration forms a novel and synergistic human pose recognition model. By simultaneously applying lightweight techniques and incorporating them harmoniously with multi-branch parallelism and multi-scale feature fusion, the model optimizes performance and computational efficiency, making it well-suited for deployment in resource-constrained environments.

(3) Emphasizing precision in pose estimation, the proposed model places a particular focus on design considerations. Through judicious technical integration and lightweight design, the model achieves a balance between computational complexity and parameter count. This balanced design ensures high accuracy in pose estimation while addressing computational resource constraints.

The structure of this paper is as follows: In Section 2, a comprehensive analysis and investigation of the HRNet are conducted. Section 3 focuses on presenting the network architecture and design concepts of the proposed lightweight HR-LiteNet. Section 4 presents the experimental results of our model on the MPII dataset and compares it with other lightweight models. Finally, in Section 5, a comprehensive summary of the paper is provided.

2. HRNet

HRNet is a network model jointly proposed by the University of Science and Technology of China and Microsoft Research Asia. It is widely used in computer vision tasks such as human pose estimation, image classification, and semantic segmentation. This network employs a multi-branch parallel approach, progressively fusing high-resolution features with low-resolution feature branches, thereby maintaining high-resolution feature representations throughout the entire process and achieving multi-scale feature fusion. This design effectively enhances the model's performance, leading to significant achievements in various tasks.

The structure of the HRNet is illustrated in Figure 1. The network consists of four stages, each with branches of different resolutions. In the first stage, feature extraction is accomplished through repeated BottleNeck modules, with each BottleNeck module comprising two 1 × 1 convolution layers, one 3 × 3 convolution layer, and a skip connection, as shown on the left in Figure 2. In the second, third, and fourth stages, feature extraction is achieved through repeated BasicBlock modules and multi-scale fusion structures, with each BasicBlock module consisting of two 3 × 3 convolution layers and a skip connection, As shown on the right side of Figure 2.

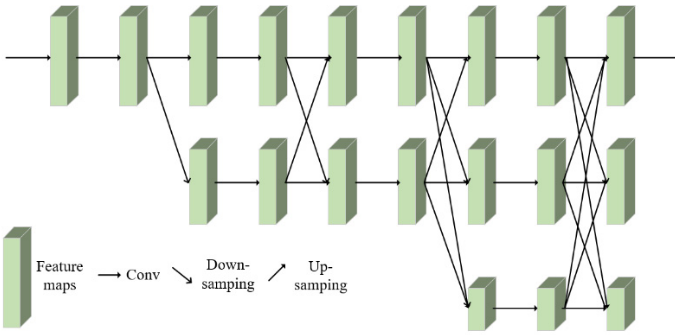


Figure 1. HRNet architecture.

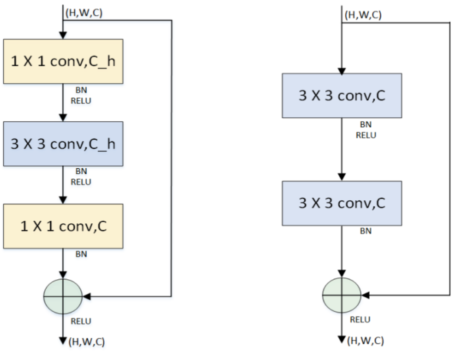


Figure 2. BottleNeck module and BasicBlock module.

This design structure enables HRNet to effectively extract and fuse multi-scale features at different stages, allowing it to excel in various computer vision tasks.

3. Lightweight HR-LiteNet

3.1. Network architecture design

HRNet is a powerful CNN known for its proficiency in handling high-resolution images and complex tasks. However, the extensive convolutional layers within this network result in relatively large parameter counts and computational requirements, posing challenges when deploying it on low computational power devices.

In this paper, we utilized the HRNet framework as the foundation for network design and introduced two fundamental lightweight modules: The L_Block module and the L_Basic module. We integrated these two modules with the HRNet, creating an entirely new lightweight human skeletal keypoint detection network model known as HR-LiteNet. The architecture of this model is depicted in Figure 3.

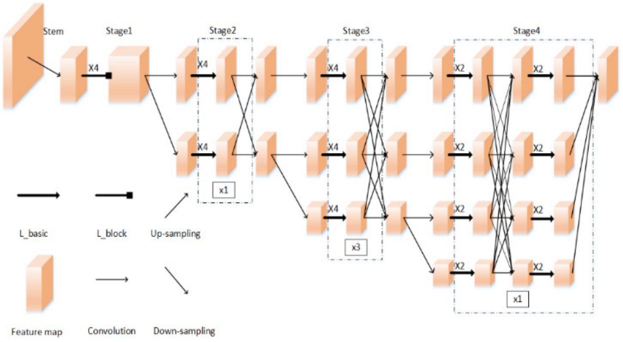


Figure 3. HR-LiteNet network architecture.

HR-LiteNet adopts a multi-branch parallel and multi-scale feature fusion approach in its architectural design. In the data preprocessing stage, the original input image with dimensions (H_o, W_o, 3) is first cropped to a fixed standard height (H) and width (W) in pixels to ensure compatibility with the input requirements of the HR-LiteNet model. Subsequently, the resized image is passed through the HR-LiteNet model.

In the Stem stage, two 3 × 3 convolutions with a stride of 2 are employed to reduce the resolution of the input image by a factor of 4 and increase the channel count from 3 to 64. Then, feature extraction is performed in four stages, each comprising 1, 2, 3, or 4 parallel branches with different resolutions. In the first stage, four L_block modules are predominantly used for feature extraction, with the first L_block module increasing the channel count from 64 to 256.In the second, third, and fourth stages, there are 1, 3, and 1 swap blocks, respectively. Each swap block consists of 4 L_basic modules and a multi-scale feature fusion module, enabling improved feature extraction and fusion. In the latter part of the fourth stage, the features from branches 2, 3, and 4 are upsampled to the same resolution as branch 1 and then summed together to generate high-resolution features.

The specific architectural details of the HR-LiteNet network are presented in Table 1.

Table 1. Specific Structural Information of HR-LiteNet Network.

Network Layer	Number of Branches	Output Feature Size	Module Composition	Channel Count	Number of Swap Blocks
Stem	1	(H/4,W/4)	Two 3 × 3 Convolutions with Stride 2	64	–
Stage 1	1	(H/4,W/4)	L_block × 4	256	–
Stage 2	2	(H/4,W/4) (H/8,W/8)	L_basic × 4 Feature Fusion Module	32,64	1
Stage 3	3	(H/4,W/4) (H/8,W/8) (H/16,W/16)	L_basic × 4 Feature Fusion Module	32,64,128	3
Stage 4	4	(H/4,W/4) (H/8,W/8) (H/16,W/16) (H/32,W/32)	L_basic × 4 Feature Fusion Module	32,64,128,256	1

3.2. Introduction to key modules

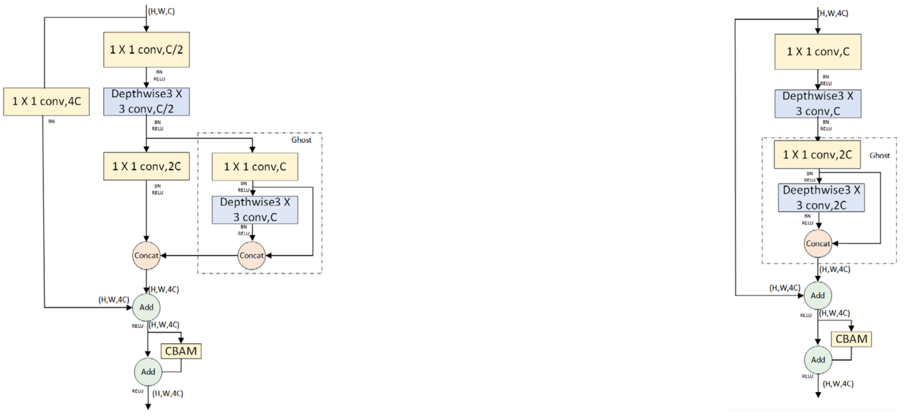
We two lightweight modules: L_Block and L_Basic. L_Block There are two structural forms according to the relationship between the number of output channels and the number of input channels. In the first form, when the output channel is four times the input channel, 1 × 1 and 3 × 3 deep separable convolutions are involved, fused by branches with residual connections and Ghost module operations. In the second form, the input and output channels are equal, simplifying the left branch, connecting directly, and using the Ghost module operation on the right branch. Both forms incorporate the Convolutional Block Attention Module (CBAM) attention mechanism. L_Basic consists of three branches: The bottom (CBAM attention), the middle (two depthwise separable convolutional layers), and the top (residual connection). Designed to enhance network understanding, feature extraction, and training stability.

As seen in Figure 4, based on the relationship between the number of input and output channels, there are two structural forms of the L_block module:

(1) When the output channel count is four times the input channel count, the L_block module appears as shown in Figure4(a). Its structure exhibits the following characteristics:

The left branch employs a 1 × 1 convolution operation to adjust the input channel count to match the output channel count. This branch is part of the residual connection to maintain network stability. The right branch first performs a 1 × 1 convolution followed by a 3 × 3 depthwise separable convolution operation. Subsequently, the right branch further divides into two sub-branches: one sub-branch processes a portion of features using a 1 × 1 convolution, while the other sub-branch generates a portion of feature maps using ghost module operations. These branches are then fused and stacked through Add and Concat operations. Finally, the network introduces the CBAM attention mechanism to better extract critical features.

(2) When the output channel count is equal to the input channel count, the L_block module appears as shown in Figure 4(b). Its structure differs from that of Figure 4(a) as follows: The left branch does not require a 1 × 1 convolution operation and is directly connected. The right branch performs a 1 × 1 convolution followed by a 3 × 3 depthwise separable convolution operation and exclusively employs ghost module operations to generate feature maps. The rest of the structure remains consistent. These descriptions outline the two variations of the L_block module based on the relationship between input and output channel counts.



(a) (b)

Figure 4. L_Block network structure.

Figure 5 illustrates the L_basic network structure, primarily comprising three branches, each serving distinct functions. First, the lower branch undergoes the CBAM attention mechanism [23], aiding the network in better understanding and focusing on crucial feature information from the input data. Next, the middle branch passes through two depthwise separable convolution layers, employed to extract features from the input data for improved adaptability to task requirements. Finally, the upper branch connects to the rest of the network through a residual connection, ensuring the stability of network training.

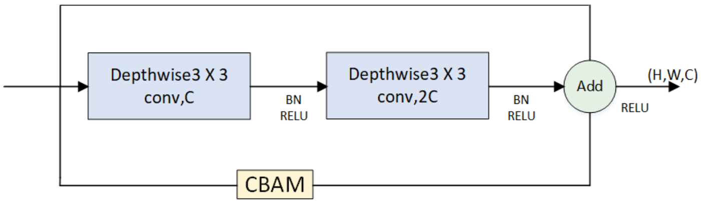


Figure 5. L_Basic network structure.

The above two modules primarily rely on depthwise separable convolution and the Ghost model for construction. The core idea behind depthwise separable convolution is to decompose the traditional convolution operation into two independent steps: Depthwise convolution and pointwise convolution. Depthwise convolution independently convolves each channel of the input feature, with each channel being operated on by a separate convolution kernel [24]. In comparison to traditional convolution, this method significantly reduces the parameter count because it no longer requires operations with all convolution kernels. Pointwise convolution employs 1 × 1 convolution kernels to convolve the output of depthwise convolution, integrating and interacting with features from different channels to generate the final output feature map.

Depthwise separable convolution, compared to traditional convolution, can substantially reduce the utilization of parameters. Assuming that depthwise separable convolution and regular convolution have the same input feature size (H_in, W_in, C_in) and output feature size (H_out, W_out, C_out), the operation of traditional convolution is as shown in Figure 6, where C_out convolution kernels of size (D, D, C_in) are used. The parameter count for traditional convolution is calculated as follows:

$$Param = D \times D \times C\_in \times C\_out \tag{1}$$

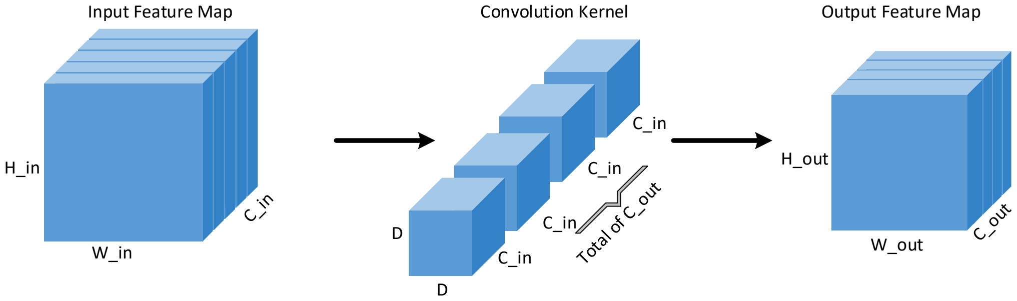
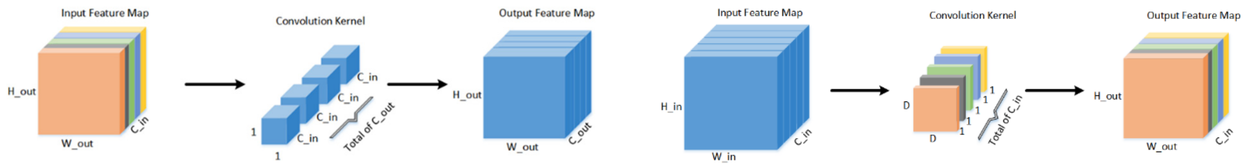


Figure 6. Convolution process in traditional convolution.

The two steps of depthwise separable convolution are as follows: Depthwise Convolution, as shown in Figure 7(a), employs C_in convolution kernels of size (D,D,1). The calculation of its parameter count is as described in Eq 2. Pointwise Convolution, as shown in Figure 7(b), uses C_out convolution kernels of size (1,1,C_in). The calculation of its parameter count is as described in Eq 3.

$$Param\_d = D \times D \times C\_in \tag{2}$$

$$Parsm\_p = C\_in \times C\_out \tag{3}$$



(a) Depthwise convolution

(b) Pointwise convolution

Figure 7. Depthwise separable convolution process.

The parameter count of a single depthwise separable convolution is calculated as shown in Eq 4.

$$Param\_dp = D \times D \times C\_in + C\_in \times C\_out \tag{4}$$

The ratio of the parameter count of depthwise separable convolution to that of standard convolution is given by Eq 5.

$$\frac{Param\_dp}{Param} = \frac{D{\times}D{\times}C\_in{+}C\_in{\times}C\_ou}{D{\times}D{\times}C\_in{\times}C\_out} = \frac{1}{C\_out} + \frac{1}{D{\times}D} \tag{5}$$

In the above equation, D represents the size of the convolution kernel, and C_out is the number of output feature map channels. For convolutions with large kernels and multiple channels, depthwise separable convolution operations can significantly reduce the number of parameters during convolution operations.

The Ghost Module is a key component in the lightweight deep learning network GhostNet proposed by Huawei Noah's Ark Lab [25]. Its core idea is to divide the convolution operation into three steps, as illustrated in Figure 8: 1) First, the Ghost Module employs a regular convolution operation to generate feature maps with fewer channels. 2) Then, by applying the Cheap Operation, it computes additional feature maps based on the ones generated in the previous step. 3) Finally, it concatenates the different feature maps generated in steps 1) and 2) to build a more diverse and powerful feature representation.

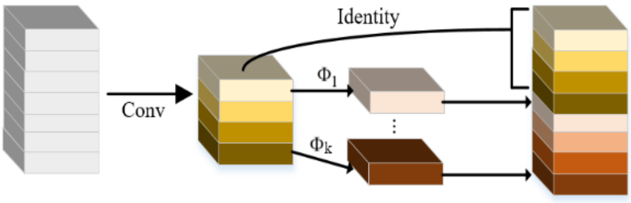


Figure 8. Ghost module structure.

The reduction in its parameter count is given by Eq 6.

$$r_c = \frac{n \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s + c - 1} \approx s \tag{6}$$

In the above equation, where r_c represents the reduction factor in parameter count, c is the input channel size, n is the output channel size, k denotes the convolution kernel size, s represents the scaling factor, and d is the size of the Cheap Operation's convolution kernel。

Therefore, the design of the L_Block and L_Basic modules proposed in this paper, based on depthwise separable convolution and the Ghost model, effectively reduces the model's parameter count and computational complexity.

4. Experimental results and analysis

4.1. Dataset

In this experiment, we chose the MPII dataset [26] to evaluate the performance of our model. The MPII Human Pose dataset is one of the leading benchmarks for evaluating joint-based human pose estimation. It contains approximately 25000 images, defines 16 types of human keypoint annotations, covers 410 common human activity scenarios, and annotates over 40000 skeletal keypoints of individuals. The MPII dataset is widely recognized in the field of human pose estimation, and many excellent pose estimation models use the Percentage of Correct Keypoints (PCK) metric to evaluate their performance on this dataset. Therefore, we selected the MPII dataset as the evaluation benchmark for our experiments to ensure the effectiveness of our model in this domain.

4.2. Evaluation

PCK is one of the commonly used performance evaluation metrics in the field of human pose estimation. It is employed to gauge the accuracy of a model in predicting the positions of key points on the human body. PCK measures the model's performance by calculating the ratio of detected keypoints to true keypoints within a normalized distance threshold. If the distance falls within the predefined threshold, the detected keypoint location is considered correct. A higher PCK value indicates better model performance. The threshold for PCK is typically set based on the relative size of body parts. A common practice is to use the diameter of the head or body as the reference size. This approach ensures a fair comparison of PCK across different human poses and sizes. Therefore, we adopt PCK as the evaluation metric for algorithm accuracy. In the MPII dataset, the head length is used as a normalization reference, and the calculation of PCK is as given in Eq 7.

$$PCK_i^k = \frac{\sum_p \delta\left(\frac{d_{pi}}{d_p{}^{def}} \leq T_k\right) \delta(v_{pi} > 0)}{\sum_{pi} \delta(v_{pi} > 0)} \tag{7}$$

In Eq 7, PCK_i^k represents the PCK metric for keypoint with id i under the threshold T_k. d_{pi} represents the Euclidean distance between the predicted ith keypoint for the current person p and the ground truth, while $d_p{}^{def}$ represents the head diameter. v_{pi} indicates the visibility of the ith keypoint for person p. T_k represents the manually set threshold.

We employ the number of parameters and computational complexity as evaluation metrics for model complexity. The number of parameters refers to the quantity of adjustable weights and parameters in the model that need to be learned. It is typically an important indicator for assessing model size and complexity. Computational complexity, on the other hand, refers to the number of computational operations that the model needs to perform during inference (prediction). Higher computational complexity may require more computational resources such as CPU, GPU, or TPU for efficient inference in real-world applications. Thus, evaluating computational complexity helps determine whether a model is suitable for specific hardware and application scenarios.

4.3. Experimental environment and details

4.3.1. Experimental environment

The software and hardware environment used in this experiment is shown in Table 2.

Table 2. Experimental environment configuration.

Hardware/Software	Configuration Description
Operating System	Windows10
Deep Learning Framework	Pytorch
Programming Language	Python
Memory	32GB DDR4
Processor	Intel Core i5-12490F Six-Core
GPU	NVIDIA GeForce RTX 3060Ti

4.3.2. Experimental details

During the data preprocessing stage, we resize the input images to 256 × 256 pixels and apply image scaling, rotation, and flipping for data augmentation. In the training process, we have chosen to use Gaussian heatmaps as the target type. This indicates that the model's output adopts a heatmap representation based on Gaussian distribution, where high probability regions correspond to accurate locations of the human body pose. This design enables the model to learn and predict the positions of keypoints in a probabilistic manner, enhancing its flexibility to adapt to various pose expressions. The model's output consists of 64 × 64 heatmaps, reducing computational complexity while retaining sufficient information to capture the structure and relationships between joints.

In the training configuration, we set the batch size to 32 and conducted a total of 210 training epochs. Adam was employed as the optimizer, with a learning rate (LR) set to 0.001. To further refine the model's learning process, we introduced a LR scheduling strategy. Specifically, at the 170th and 200th epochs, we adjusted the LR by a factor of LR_FACTOR (0.1). Additionally, we configured the weight decay (WD) to be 0.0001, aiming to impose regularization control on the model parameters. The momentum was set to 0.9, facilitating the consideration of historical gradients during the parameter update process.

4.4. Experimental results and analysis

Table 3 presents the performance results of our proposed method HR-LiteNet and other methods on the MPII validation dataset. The compared algorithms, Ghostnet_pose, shufflenetv2_pose, mobilenetv2_pose, and efficientnet_pose, are lightweight human pose models built on popular lightweight network architectures such as Ghostnet, shufflenetv2 [27], mobilenetv2 [28], and efficientnet [29], where the classification layers of the original models are replaced with deconvolutional and fully connected layers for human pose estimation.

Our method achieved an accuracy of 83.6%on the MPII validation dataset while reducing the parameter count by 26.58 M and decreasing the computation cost by 8.04 GFLOPs compared to the Hrnet_W32 network. In comparison to shufflenetv2_pose and efficientnet_pose networks, which maintain similar accuracy, HR-LiteNet reduced the parameter count by 5.63 M and 9.43 M, respectively, and decreased the computation cost by 0.24 GFLOPs and 0.61 GFLOPs, respectively. Compared to the mobilenetv2_pose network, HR-LiteNet improved accuracy by 2.45%, reduced the parameter count by 7.65 M, and decreased the computation cost by 0.51 GFLOPs. Our approach demonstrates satisfactory results in various performance metrics, parameter count, and computational cost, providing valuable insights for the research and application of lightweight human pose recognition models.

Table 3 provides a comparison of the experimental results on the validation dataset for various models.

Table 3. Experimental comparison of multiple algorithms.

Method	**Input**	**Pretrain**	**Params (M)**	**GFLOPs**	**PCK**
Hrnet_W32	256 × 256	Y	28.50	9.50	90.300
Ghostnet_pose	256 × 256	N	8.71	1.68	85.428
shufflenetv2_pose	256 × 256	N	7.55	1.70	84.065
mobilenetv2_pose	256 × 256	N	9.57	1.97	81.194
efficientnet_pose	256 × 256	N	11.35	2.07	83.851
HR-LiteNet	256 × 256	N	1.92	1.46	83.643

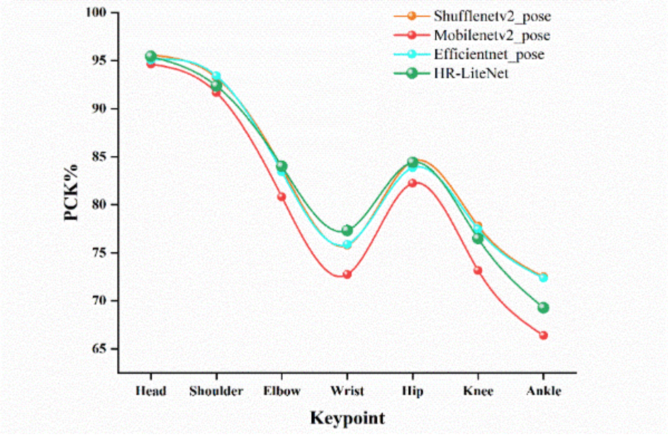


Figure 9. Keypoint accuracy of the lightweight network.

Figure 9 illustrates the keypoint accuracy performance of our proposed HR-LiteNet method compared to other lightweight methods on the MPII validation dataset. As observed from the figure, our constructed lightweight model performs well in the recognition of challenging keypoints such as the Elbow, Wrist, and Hip.

Table 4 presents the performance results of HR-LiteNet and other lightweight methods on the MPII test set. HR-LiteNet achieved an accuracy of 85% on the test set, which is 1.9% higher than that of mobilenetv2_pose. While reducing the number of parameters by 7.65 M and the computational cost by 0.51 GFLOPs compared to mobilenetv2_pose. In comparison to efficientnet_pose, HR-LiteNet achieved nearly identical accuracy while reducing the number of parameters by 9.43 M and lowering the computational cost by 0.61 GFLOPs.

Figure 10 displays the visual recognition results of our proposed human pose recognition method on a series of human image samples. As seen in the images, our method has achieved good performance in identifying human key points. However, we also need to acknowledge some limitations: First, our method currently supports only single-person recognition; second, under conditions of high occlusion, there may be instances of misidentification. In light of these limitations, we will strive for further improvements and exploration in our future work.

The experimental results demonstrate that in scenarios with low occlusion, we ensure the accuracy of model performance while successfully reducing the model's parameter count and computational complexity. Regarding practical video recognition applications, our model achieves a high efficiency of 15 frames per second on a device equipped with a 3060ti graphics card, ensuring smooth and stutter-free recognition of video frames. This provides a viable solution for real-time video recognition under constrained hardware conditions.

Table 4. Comparative experimental results of various models on the MPII test set.

Method	Params (M)	GFLOPs	Head	Shoulder	Elbow	Wrist	Hip	PCK
Ghostnet_pose	8.71	1.68	97.4	94.0	86.7	80.6	86.2	86.5
shufflenetv2_pose	7.55	1.70	97.1	93.7	85.5	78.7	85.4	85.4
mobilenetv2_pose	9.57	1.97	96.7	92.3	83.2	76.1	84.2	83.1
efficientnet_pose	11.35	2.07	97.0	93.5	84.7	79.0	85.8	85.1
HR-LiteNet	1.92	1.46	97.2	93.3	85.7	79.6	86.1	85.0

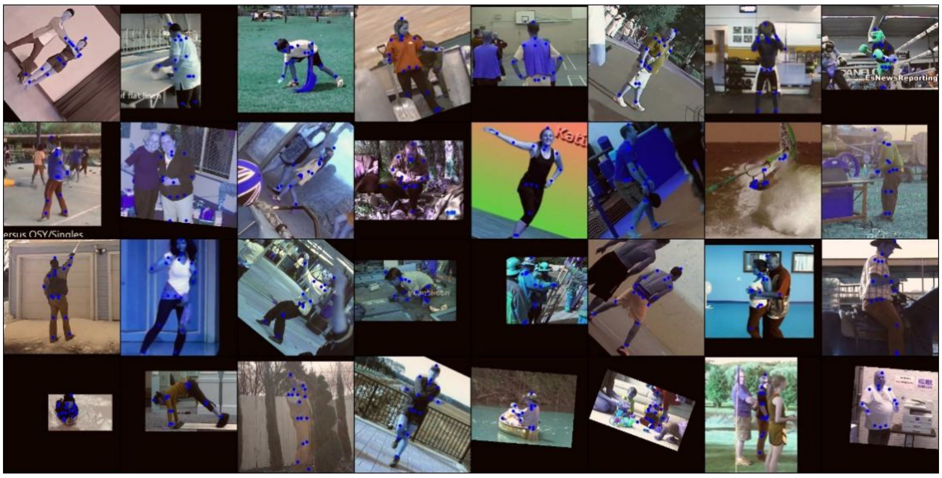


Figure 10. Some visualization result graphs of our method.

5. Conclusions

In practical applications, deploying large-scale network algorithms on regular devices can be challenging due to computational and memory constraints. To make models more lightweight and suitable for resource-limited environments, this study introduced a novel lightweight human skeleton keypoint detection network model, HR-LiteNet, based on the HRNet. HR-LiteNet incorporates lightweight L_Block and L_Basic modules. Experimental results on the MPII dataset demonstrate that HR-LiteNet maintains high accuracy compared to other lightweight networks while significantly reducing the number of parameters and computational complexity. This design provides a new solution for pose recognition in resource-constrained environments, striking a balance between accuracy and lightweight requirements.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the Research Project of Fujian University of Technology (No. GY-H-22190).

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. S. Wu, Z. Wang, B. Shen, J. Wang, D. Li, Human-computer interaction based on machine vision of a smart assembly workbench, *Assem. Autom.*, **40** (2020), 475–482. https://doi.org/10.1108/AA-10-2018-0170
2. B. Debnath, M. O'brien, M. Yamaguchi, A. Behera, A review of computer vision-based approaches for physical rehabilitation and assessment, *Multimedia Syst.*, **28** (2022), 209–239. https://doi.org/10.1007/s00530-021-00815-4
3. N. Lyons, Deep learning-based computer vision algorithms, immersive analytics and simulation software, and virtual reality modeling tools in digital twin-driven smart manufacturing, *Econ. Manage. Financ. Mark.*, **17** (2022), 67–81. https://doi.org/10.22381/emfm17220224
4. Q. Kha, Q. Ho, N. Q. K. Le, Identifying snare proteins using an alignment-free method based on multiscan convolutional neural network and PSSM profiles, *J. Chem. Inf. Model.*, **62** (2022), 4820–4826. https://doi.org/10.1021/acs.jcim.2c01034
5. Z. Zhao, J. Gui, A. Yao, N. Q. K. Le, M. C. H. Chua, Improved prediction model of protein and peptide toxicity by integrating channel attention into a convolutional neural network and gated recurrent units, *ACS Omega*, **7** (2022), 40569–40577. https://doi.org/10.1021/acsomega.2c05881

6. Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A survey of convolutional neural networks: analysis, applications, and prospects, *IEEE Trans. Neural Networks Learn. Syst.*, **33** (2022), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827
7. C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, Z. Ding, 3D human pose estimation with spatial and temporal transformers, preprint, arXiv:2103.10455.
8. C. Li, G. H. Lee, Generating multiple hypotheses for 3D human pose estimation with mixture density network, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 9879–9887. https://doi.org/10.1109/CVPR.2019.01012
9. A. Toshev, C. Szegedy, Deeppose: Human pose estimation via deep neural networks, in 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, (2014), 1653–1660. https://doi.org/10.1109/CVPR.2014.214
10. J. Tompson, A. Jain, Y. LeCun, C. Bregler, Joint training of a convolutional network and a graphical model for human pose estimation, preprint, arXiv: 1406.2984.
11. S. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, Convolutional pose machines, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 4724–4732. https://doi.org/10.1109/CVPR.2016.511
12. Y. Chen, Y. Tian, M. He, Monocular human pose estimation: A survey of deep learning-based methods, *Comput. Vision Image Understanding*, **192** (2020), 102897. https://doi.org/10.1016/j.cviu.2019.102897
13. C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, et al., Deep learning-based human pose estimation: A survey, *ACM Comput. Surv.*, **56** (2023), 1–37. https://doi.org/10.1145/3603618
14. G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, et al., Towards accurate multi-person pose estimation in the wild, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 4903–4911. https://doi.org/10.1109/CVPR.2017.395
15. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39** (2017), 1137–1149. https://doi.org/10.1109/tpami.2016.2577031
16. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90
17. K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 5693–5703. https://doi.org/10.1109/CVPR.2019.00584
18. L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, et al., Deepcut: Joint subset partition and labeling for multi person pose estimation, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 4929–4937. https://doi.org/10.1109/CVPR.2016.533
19. Z. Cao, T. Simon, S. Wei, Y. Sheikh, Realtime multi-person 2D pose estimation using part affinity fields, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 1302–1310. https://doi.org/10.1109/CVPR.2017.143
20. F. Zhang, X. Zhu, M. Ye, Fast human pose estimation, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 3512–3521. https://doi.org/10.1109/CVPR.2019.00363

21. D. Xu, R. Zhang, L. Guo, C. Feng, S. Gao, LDNet: Lightweight dynamic convolution network for human pose estimation, *Adv. Eng. Inf.*, **54** (2022), 101785. https://doi.org/10.1016/j.aei.2022.101785
22. C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang, et al., Lite-HRNet: A lightweight high-resolution network, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 10435–10445. https://doi.org/10.1109/CVPR46437.2021.01030
23. S. Woo, J. Park, J. Lee, I. S. Kweon, Cbam: Convolutional block attention module, in *European Conference on Computer Vision*, **11211** (2018), 3–19. https://doi.org/10.1007/978-3-030-01234-2_1
24. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, et al., MobileNets: Efficient convolutional neural networks for mobile vision applications, preprint, arXiv: 1704.04861.
25. K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, Ghostnet: More features from cheap operations, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 1577–1586. https://doi.org/10.1109/CVPR42600.2020.00165
26. M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2D human pose estimation: New benchmark and state of the art analysis, in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, (2014), 3686–3693. https://doi.org/10.1109/CVPR.2014.471
27. N. Ma, X. Zhang, H. Zheng, J. Sun, Shufflenet v2: Practical guidelines for efficient cnn architecture design, in *European Conference on Computer Vision*, **11218** (2018), 122–138. https://doi.org/10.1007/978-3-030-01264-9_8
28. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2018), 4510–4520. https://doi.org/10.1109/CVPR.2018.00474
29. M Tan, Q. V. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, preprint, arXiv:1905.11946.