Journal of Electronic Research and Application, 2025, Volume 9, Issue 5
http://ojs.bbwpublisher.com/index.php/JERA
ISSN Online: 2208-3510
ISSN Print: 2208-3502

SW-YOLO: Lightweight Attitude Estimation Algorithm Based on Weighted Convolution and Star Network

Qian Xu*

School of Computer Science and Technology, Taiyuan Normal University, Jinzhong 030619, China

**Author to whom correspondence should be addressed.*

Abstract: This paper proposes SW-YOLO (StarNet Weighted-Conv YOLO), a lightweight human pose estimation network for edge devices. Current mainstream pose estimation algorithms are computationally inefficient and have poor feature capture capabilities for complex poses and occlusion scenarios. This work introduces a lightweight backbone architecture that integrates WConv (Weighted Convolution) and StarNet modules to address these issues. Leveraging StarNet's superior capabilities in multi-level feature fusion and long-range dependency modeling, this architecture enhances the model's spatial perception of human joint structures and contextual information integration. These improvements significantly enhance robustness in complex scenarios involving occlusion and deformation. Additionally, the introduction of WConv convolution operations, based on weight recalibration and receptive field optimization, dynamically adjusts feature importance during convolution. This reduces redundant computations while maintaining or enhancing feature representation capabilities at an extremely low computational cost. Consequently, SW-YOLO substantially reduces model complexity and inference latency while preserving high accuracy, significantly outperforming existing lightweight networks.

Keywords: YOLO11-Pose; WConv; StarNet; Lightweight algorithms; Feature fusion

Online publication:

1. Introduction

As a core and highly practical foundational task in computer vision, human pose estimation has demonstrated robust application potential across diverse scenarios in recent years. Technologically, it has become deeply integrated into multiple critical domains, including abnormal behavior analysis in intelligent surveillance systems, gesture recognition for human-machine interaction in consumer electronics, posture correction assessment in medical rehabilitation, and safety monitoring for human-machine collaboration in industrial environments. However, these algorithms are primarily used on edge devices, such as embedded terminals, IoT sensors, and

mobile terminals. These devices typically have limited computational power, memory, and storage capacity, which makes supporting the complex computational demands of traditional, high-precision pose estimation algorithms challenging. This contradiction directly drives the evolution of technical requirements; the traditional pursuit of accuracy alone is no longer sufficient for edge scenarios. The industry now demands more stringent standards for the lightweight nature, real-time inference efficiency, and resource compatibility of pose estimation algorithms. Against this backdrop, the critical challenge—and the core focus of this paper's research—is to design lightweight human pose estimation algorithms that can overcome computational resource limitations on edge devices. These algorithms must simultaneously ensure the accurate detection of key pose information while meeting demands for low computational power and a minimal storage footprint.

Both academia and industry have proposed various lightweight human pose estimation solutions for edge devices [1–4]. Among these, OpenPose uses Part Affinity Fields (PAFs) to achieve high-precision joint association. However, even when adapting its optimized version for edge devices and replacing feature extraction modules with lightweight alternatives to reduce computational overhead, its efficiency remains insufficient for real-time deployment. Furthermore, OpenPose struggles with keypoint feature capture and contextual integration in scenarios involving occlusion and joint deformation [5]. MobileNetV4 is commonly used as a backbone network due to its parameter compression and enhanced local features via grouped convolution, decoupling, and channel attention mechanisms [6]. However, it suffers from weaknesses in multi-level feature fusion and modeling long-range dependencies. It has a poor perception of human joint spatial structures, and the insufficient dynamic adjustment of feature importance during convolution leads to computational redundancy. In summary, existing solutions have yet to address the core challenges of insufficient computational efficiency on edge devices and weak keypoint feature capture under complex poses. This provides direction for improving the SW-YOLO net.

This paper addresses the issues of insufficient computational efficiency in edge devices and reduced accuracy in human pose estimation under complex scenarios by proposing a lightweight pose estimation network, SW-YOLO (StarNet Weighted-Conv YOLO). By incorporating the StarNet module into the backbone network, the model's ability to represent multi-level features is enhanced, significantly improving robustness in challenging scenarios such as occlusion and motion blur [7]. Simultaneously, replacing standard convolutions with weighted convolutions (WConv) reduces computational overhead while preserving sensitivity to key feature points. Experimental results demonstrate that this approach effectively improves the accuracy and real-time performance of human pose estimation on edge devices while maintaining low computational complexity [8].

In summary, the SW-YOLO model proposed in this paper contributes as follows:

(1) The introduction of lightweight StarNet modules constructs an efficient backbone network. Through its sparse connections and parameter sharing mechanism, it significantly reduces model complexity while enhancing multi-scale feature fusion capabilities, effectively mitigating feature discrimination bias in occlusion and deformation scenarios.

(2) Replacing standard convolution with WConv allows for dynamic kernel weight optimization and receptive field adjustment. This reduces computational overhead while enhancing the ability to capture key point details, further improving the balance between model accuracy and efficiency.

2. Related work

YOLO11-Pose performs well on the COCO2017 dataset, striking an optimal balance between accuracy and speed.

However, the model faces computational efficiency challenges on edge devices and is susceptible to occlusion, motion blur, and background interference in complex scenes. These issues lead to degraded performance in detecting human keypoints. To address these issues, this paper presents a lightweight pose estimation model called SW-YOLO (StarNet Weighted-Conv YOLO). The model incorporates the StarNet module to enhance multi-scale feature fusion and spatial structure perception. This effectively suppresses complex background interference while improving keypoint discrimination under occlusion and deformation. The model simultaneously replaces standard convolutions with WConv structures to significantly reduce parameters and computational load while preserving sensitivity to fine features. Experiments demonstrate that SW-YOLO substantially improves robustness and estimation accuracy in complex, real-world scenarios while reducing computational complexity.

The backbone network of YOLO11-Pose was restructured by introducing the StarNet module and the WConv convolution operation to construct the lightweight and efficient SW-YOLO model [9]. The SW-YOLO model leverages StarNet's strengths in multi-level feature fusion and long-range dependency modeling to improve the perception of human keypoints in complex backgrounds. It simultaneously employs WConv weighted convolutions to dynamically adjust receptive fields and recalibrate features, reducing the number of parameters while improving the model's sensitivity to fine-grained features. The proposed model architecture is illustrated in **Figure 1**.

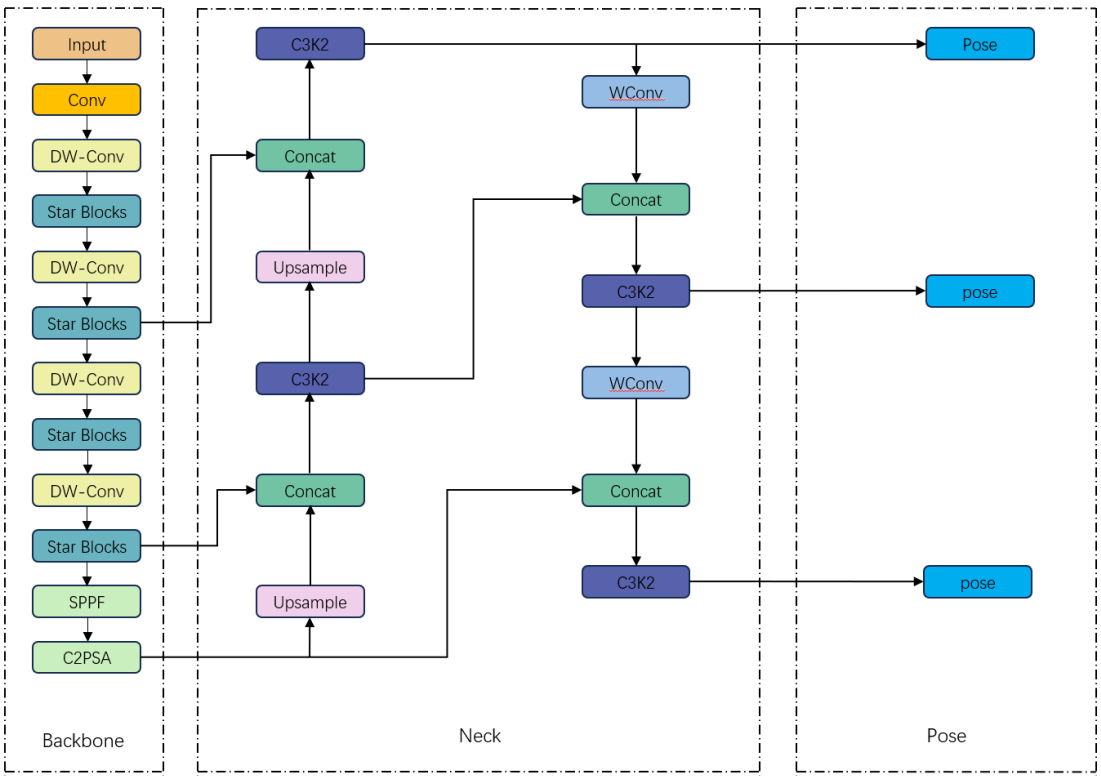

Figure 1. SW-YOLO

2.1. WConv

The WConv module is an efficient convolution operation specifically designed for lightweight neural networks. Its core functionality lies in introducing a dynamic channel attention mechanism that adaptsively recalibrates the importance of each channel in the input feature map through attention weights generated online. This module

enables the network to autonomously amplify information-rich feature channels while suppressing responses from redundant or noisy channels, thereby simulating a "visual focus" effect. This process significantly enhances the model's ability to capture and distinguish critical detail features in complex scenarios, such as occlusion, deformation, and background interference. Crucially, WConv cleverly encapsulates this powerful feature enhancement within minimal theoretical computational overhead. This design significantly boosts standard convolution representation capabilities while perfectly meeting edge devices' stringent efficiency demands, achieving an exceptional balance between accuracy and speed. Consequently, WConv replaces standard convolutions in Yolo11-Pose.

First, perform global information embedding on the input feature map $X \in \mathbb{R}^{C\times H\times W}$ perform global average pooling (GAP). Compress it into a global context vector $z \in \mathbb{R}^{C}$. This vector encodes the global statistical information for each channel.

$$z_c = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W} x_c\,(i,j) \tag{1}$$

Then, generate weights by feeding the global context vector z into a lightweight gating mechanism, dynamically producing a channel attention weight vector $\alpha \in \mathbb{R}^{C}$.

$$\alpha = \sigma\big(\mathbf{W}_2\delta(\mathbf{W}_1 z)\big) \tag{2}$$

Among these, W_1 and W_2 represent the weights of the fully connected layer, δ denotes the ReLU function, and σ denotes the Sigmoid function. Each element of the weight vector α ranges between 0 and 1, indicating the importance of the corresponding channel.

Next, feature reweighting is performed by multiplying the generated attention weights α with the original input feature map X on a per-channel basis, yielding a calibrated, weighted feature map $\widetilde{X}$.

$$\tilde{x}_c = \alpha_c \cdot x_c \tag{3}$$

Finally, a standard convolution operation (such as a 3×3 convolution) is performed on the weighted feature map $\widetilde{X}$, yielding the output feature map Y.

$$Y = \mathrm{Conv}\big(\widetilde{X}\big) \tag{4}$$

WConv implements a "focus first, compute later" mechanism, significantly enhancing the efficiency of convolution operations and making it highly suitable for lightweight neural network designs.

2.2. StarNet

This paper introduces the StarNet module, an efficient component of the backbone network. The StarNet module uses lightweight techniques, such as separable convolutions, batch convolutions, and parameter sharing, to create a star-shaped topology with shared central nodes and multiple parallel branches. The central nodes perform feature dimension reduction and preliminary fusion with 1×1 batch convolutions, which significantly reduces redundant computations between channels. The peripheral branches use parallel separable convolutions with different receptive fields to extract multi-scale features and capture rich contextual information while avoiding the computational burden of standard dilated convolutions. Finally, an adaptive feature recalibration mechanism enhances important feature channels. This design enables StarNet to strike a remarkable balance. It effectively enhances the model's robustness against occlusion, scale changes, and non-rigid deformations with extremely low parameter counts and computational overhead. It also significantly reduces the model's overall complexity. This

enables high-precision, real-time human pose estimation on edge devices.

StarNet is a 4-layer hierarchical structure that uses convolutional layers for downsampling and modified demo blocks for feature extraction. To meet efficiency requirements, batch normalization replaces layer normalization and is placed after deep convolutions, which can be fused during inference [10,11]. Inspired by MobileNet, a depthwise convolutional layer (DWConv) is added at the end of each block [12]. The channel expansion factor is consistently set to four, doubling the network width at each stage. Within the demo blocks, GELU activation is replaced with ReLU6, following MobileNetv2's design [13]. **Figure 2** illustrates the StarNet framework.

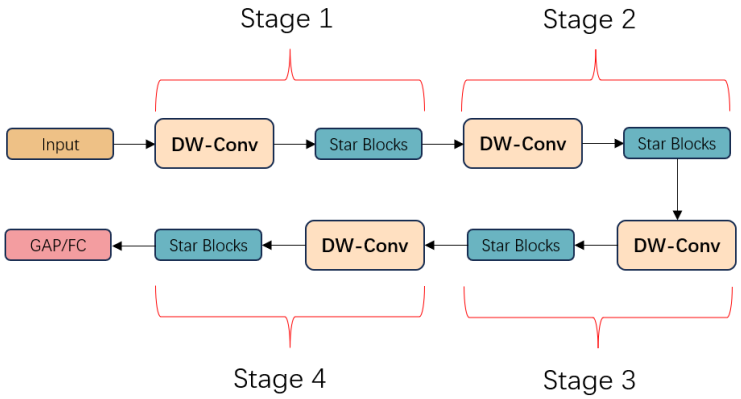

Figure 2. StarNet network

The Star Block efficiently and powerfully constructs lightweight models by stacking multiple fundamental building blocks. Its core "Star Operation" has a mathematical mechanism similar to kernel methods [14–16]. It explicitly constructs higher-order feature interaction terms through systematic, pairwise multiplication operations across different feature channels [17]. This process closely resembles the expansion of polynomial kernel functions. When embedded within neural network architectures and stacked across multiple layers, each layer enables exponential growth in the complexity of feature representations. Consequently, Star Operations achieves ultra-high-dimensional feature representations within an extremely compact feature space, even with just a few stacked layers [18]. This significantly reduces the number of parameters while preserving strong model expressiveness. The architecture is illustrated in **Figure 3**.

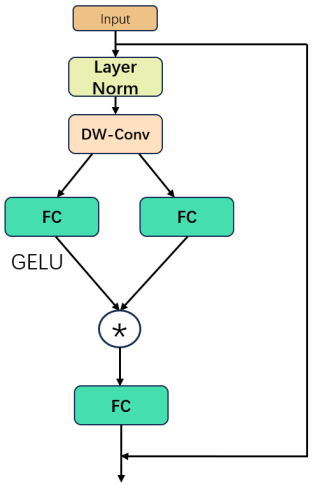

Figure 3. Star block

3. Experimental design and results

3.1. Details of the experiment

(1) Introduction to the dataset

The experiments use the COCO2017 human keypoint dataset, which contains labeled image data covering complex scenes, such as multiple people, occlusion, and diverse actions, with 17 keypoints. The dataset includes 58,945 images, of which 56,599 are in the training set and 2,346 are in the validation set.

(2) Evaluation metrics

Our method was systematically validated and evaluated on the COCO2017 dataset. This work employs Object Keypoint Similarity (OKS) as the core metric for evaluating model performance. OKS comprehensively and accurately reflects the performance of pose estimation models by calculating the normalized distance between predicted and ground-truth keypoints, while accounting for the annotation difficulty and scale variations across different keypoints. Its specific calculation formula is as follows:

$$OKS = \frac{\sum_i \exp(-\frac{d_i^2}{2s^2k_i^2})\delta(v_i>0)}{\sum_i \delta(v_i>0)} \tag{5}$$

Where d_i denotes the Euclidean distance between the detected keypoint and its true label, v_i denotes the visibility flag bit of the true keypoint on the surface, s denotes the scale of the target, and k_i denotes the normalization factor of the i^(th) human keypoint. This indicates that a keypoint is a positive example if its OKS is greater than a certain threshold. AP50 is the prediction accuracy at a 0.5 similarity threshold. AP50-95 is the average precision (AP) calculated at multiple OKS thresholds (from 0.5 to 0.95 in 0.05 increments).

(3) Experimental environment and settings

The experimental environment is as follows: the operating system is Windows 10, and the GPU is an NVIDIA Tesla T4 that utilizes parallel computing to accelerate model training and inference. The optimizer uses the stochastic gradient descent (SGD) algorithm. The initial learning rate is 0.01, the batch size is 32, and there are 100 iteration rounds.

3.2. Ablation experiments

To validate the effectiveness of each proposed improvement module, we conducted systematic ablation experiments on the COCO2017 human keypoint dataset. We compared four configurations: the baseline model, the model with the StarNet module, the model with the WConv module, and the model incorporating both StarNet and WConv modules. The experimental results are shown in **Table 1**.

Table 1. Ablation experiments on the COCO2017 dataset

Methods	Params/M	GFLOPs	AP50	AP50-95
YOLO11n-Pose	StarNet	WConv	Params/M	GFLOPs
√			2.91	7.7
√	√		2.27	6.1
√		√	2.86	7.2
√	√	√	2.23	6.3

Research indicates that using the StarNet backbone network alone reduces the number of parameters by 0.64 million, while lowering AP50 by 0.7% and computational load by 1.6 GFLOPs, demonstrating improved lightweight efficiency and computational effectiveness. Using WConv alone reduces parameters by 0.5M, lowers AP50 by 0.2%, and decreases computational load by 0.5GFLOPS. This minimizes redundant computations while maintaining or even enhancing feature representation capabilities at extremely low computational overhead. Integrating both modules into the YOLOv11n-Pose framework reduces parameters by 0.68M, lowers AP50 by 0.5%, and decreases computational load by 1.4GFLOPS. This demonstrates the superiority of the proposed SW-YOLO model, effectively reducing model parameters while enhancing its ability to capture keypoint details.

3.3. Comparison experiment

To visually validate the superiority of the SW-YOLO model, comparative training experiments were conducted. As shown in **Table 2**, compared to YOLO series pose estimation models, the proposed model achieves optimal performance in both parameter count and computational complexity. While its accuracy is slightly lower than that of YOLO series algorithms, this reduction is an unavoidable consequence of significantly decreasing the number of parameters.

Table 2. Comparison experiments on the COCO2017 dataset

Methods	Params/M	GFLOPs	AP50	AP50-95
Yolov5n-Pose	2.80	8.3	72.6	39.6
Yolov8n-Pose	3.30	9.2	75.6	42.7
Yolov10n-Pose	2.55	7.8	75.3	42.6
Yolov11n-Pose	2.91	7.7	74.2	42.0
Yolov12n-Pose	2.88	7.8	74.9	41.2
Ours (SW-YOLO)	2.23	6.3	73.7	40.8

4. Conclusion

To address the high computational complexity of current pose estimation algorithms and the challenges of deploying them on resource-constrained edge devices, this paper proposes a novel lightweight network architecture, SW-YOLO. Building upon the YOLOv11n-Pose baseline architecture, this model enhances multi-scale feature fusion and spatial structure modeling capabilities by incorporating the StarNet module. It employs WConv to dynamically adjust the receptive field and recalibrate features, significantly reducing model parameters while strengthening the capture of fine-grained details. Experiments demonstrate that SW-YOLO effectively improves pose estimation accuracy in complex scenarios involving occlusion and deformation while maintaining low computational complexity, achieving an optimized balance between precision and efficiency.

Disclosure statement

The author declares no conflict of interest.

References

[1] Ma N, Zhang X, Zheng HT, et al., 2018, ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design, in Ferrari V, Hebert M, Sminchisescu C, Weiss Y, (eds) Computer Vision – ECCV 2018. Lecture Notes in Computer Science, vol 11218, Springer, Cham.

[2] Zhang X, Zhou X, Lin M, et al., 2017, ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. arXiv. https://arxiv.org/abs/1707.01083

[3] Yu C, Xiao B, Gao C, et al., 2021, Lite-HRNet: A Lightweight High-Resolution Network, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[4] Kreiss S, Bertoni L, Alahi A, 2022, OpenPifPaf: Composite Fields for Semantic Keypoint Detection and Spatio-Temporal Association. IEEE Transactions on Intelligent Transportation Systems, 23(8): 13498–13511.

[5] Cao Z, Hidalgo G, Simon T, et al., 2021, OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(1): 172–186.

[6] Howard AG, Zhu M, Chen B, et al., 2017, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv. https://arxiv.org/abs/1704.04861

[7] Ma X, Dai X, Bai Y, et al., 2024, Rewrite the Stars, 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[8] Cammarasana S, 2025, Optimal Weighted Convolution for Classification and Denosing. arXiv. https://arxiv.org/abs/2505.24558

[9] Khanam R, Hussain M, 2024, YOLOv11: An Overview of the Key Architectural Enhancements. arXiv. https://arxiv.org/abs/2410.17725

[10] Ioffe S, Szegedy C, 2015, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, Proceedings of the 32nd International Conference on Machine Learning, PMLR 37: 448–456.

[11] Ba JL, 2016, Layer Normalization. arXiv. https://arxiv.org/abs/1607.06450

[12] Chollet F, 2016, Xception: Deep Learning with Depthwise Separable Convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1800–1807.

[13] Sandler M, Howard A, Zhu M, et al., 2018, MobileNetV2: Inverted Residuals and Linear Bottlenecks, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[14] Wang J, Sun K, Cheng T, et al., 2021, Deep High-Resolution Representation Learning for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(10): 3349–3364.

[15] Yuan F, Lin Z, Tian Z, et al., 2025, Bio-Inspired Hybrid Path Planning for Efficient and Smooth Robotic Navigation. Int J Intell Robot Appl.

[16] Liang B, Yuan F, Deng J, et al., 2025, Cs-pbft: A Comprehensive Scoring-Based Practical Byzantine Fault Tolerance Consensus Algorithm. J Supercomput, 81: 859.

[17] Yuan F, Huang X, Jiang H, et al., 2025, An xLSTM–XGBoost Ensemble Model for Forecasting Non-Stationary and Highly Volatile Gasoline Price. Computers, 14: 256.

[18] Zhang K, Yuan F, Jiang Y, et al., 2025, A Particle Swarm Optimization-Guided Ivy Algorithm for Global Optimization Problems. Biomimetics (Basel), 10(5): 342.