Alexandria Engineering Journal 128 (2025) 100–116

Contents lists available at ScienceDirect

Alexandria Engineering Journal

journal homepage: www.elsevier.com/locate/aej

Original article

Application of Swin-UNet and pose estimation-based athlete motion quality detection device in IoT systems

GuanLan Cai [a],[*], Guodong Zhang [b]

[a] Sports Department, Zhengzhou University of Aeronautics, zhengzhou, 450046, Henan, China
[b] School of Information Engineering, Liaoning Institute of Science and Engineering, Jinzhou, 121013, Liaoning, China

ARTICLE INFO

Keywords:
SwinUNetPose
Pose estimation
Swin Transformer
U-Net
Human segmentation module
IoT (Internet of Things)

ABSTRACT

With the rapid development of IoT technology, athlete motion quality detection devices based on human pose estimation have significant potential in sports training and health monitoring. This study proposes a motion quality detection device combining Swin-UNet and pose estimation to improve accuracy and robustness. The SwinUNetPose model integrates the Swin Transformer with U-Net to enhance multi-scale information capture and includes a human segmentation module to optimize pose estimation, particularly in complex backgrounds and occlusion scenarios. The experimental results demonstrate that SwinUNetPose achieves superior performance compared to existing methods, excelling in both accuracy (AP) and recall rate (AR) on the COCO dataset, while maintaining a competitive inference speed. The method demonstrates scalability and efficiency, making it suitable for real-time motion quality detection, particularly in multi-athlete scenarios. This study highlights SwinUNetPose's reliability in advancing sports health monitoring in IoT systems.

1. Introduction

With the rapid development of Internet of Things (IoT) technology, the widespread use of smart devices and sensors is transforming various industries, especially in the sports field. Real-time monitoring and analysis of athletes' movement quality have become essential means for improving athletic performance and preventing sports injuries. Traditional movement quality detection methods often rely on manual observation and experience-based analysis, which are inefficient and prone to human bias. With the advancement of deep learning technologies, especially pose estimation and computer vision, combining sensor and camera data for athlete motion analysis has become a growing trend. The performance of athletes in training and competition directly affects their competitive level. As sports science has developed, sports training has shifted from focusing solely on physical fitness to emphasizing the precision of sports techniques and movement efficiency. High-quality movements not only enhance performance but also effectively reduce the risk of injury, thereby extending an athlete's career. Therefore, how to scientifically, objectively, and accurately assess and improve athletes' movement quality has become a crucial issue in sports training and sports medicine. Traditional movement analysis primarily relies on the coach's experience and subjective judgment, making comprehensive and quantitative analysis difficult. With the continuous advancement of computer technology, sensor technology, and artificial intelligence, methods for assessing athletes' movement quality have entered a new era. Modern technologies enable precise data collection and analysis, providing quantitative feedback to help athletes optimize their techniques, prevent injuries, improve training outcomes, and implement personalized training programs. These detection methods are significant in improving athletic performance, preventing injuries, providing personalized feedback, and enhancing training efficiency. Through movement quality detection, athletes can identify and correct deficiencies in their movements, improve technical skills, and enhance movement efficiency. Additionally, these methods can detect potentially dangerous movements, reducing the risk of injury. Personalized feedback helps athletes adjust their training plans accordingly, maximizing training effectiveness. Recent research in the field of Internet of Things (IoT) applications has also explored solutions to improve the efficiency and performance of such systems. For example, Naouri et al. [1] discuss efficient fog node placement strategies for IoT applications, which are crucial for optimizing the deployment of real-time athlete motion quality detection systems. Khelloufi et al. [2] propose a multimodal latent-features-based service recommendation system, which provides valuable insights for offering personalized feedback in the development of IoT-based sports health monitoring systems.

With the continuous advancement of sports science and technology, the methods for detecting the quality of athletes' movements have gradually shifted from traditional subjective observation to more

* Corresponding author.
E-mail addresses: caiguanlan@zua.edu.cn (G. Cai), zgdliaoligongedu@126.com (G. Zhang).

https://doi.org/10.1016/j.aej.2025.05.017
Received 19 March 2025; Received in revised form 23 April 2025; Accepted 5 May 2025
Available online 26 May 2025

precise and quantitative techniques. In recent years, pose estimation, as an emerging method for motion quality detection, has received widespread attention due to its good balance between accuracy and flexibility. This method utilizes computer vision and deep learning technologies to analyze 2D or 3D video data and extract the key point positions of athletes in real-time, thereby providing a more accurate assessment of their motion quality. Compared with traditional motion capture technologies and sensor-based methods, pose estimation has distinct advantages, including no need for wearable devices, ease of operation, and lower costs. Therefore, pose estimation is not only suitable for daily training but also provides convenient technical support for real-time monitoring and large-scale performance evaluation of athletes. However, pose estimation methods still have certain limitations in practical applications. Firstly, video quality and shooting angles have a significant impact on detection accuracy. Low-quality videos and suboptimal shooting angles may lead to inaccurate joint position calibration, which in turn affects the accuracy of motion analysis results. Secondly, for some fast and complex movements, the accuracy of pose estimation may decline, especially in cases of low resolution or when the athlete is partially occluded. In such situations, the model's performance may be significantly compromised. Furthermore, current pose estimation models are still not fully capable of handling complex motion scenarios, particularly in multi-person sports or dynamic environments, where the application effectiveness still needs improvement. Despite these challenges, pose estimation remains a promising technology with great potential for application in detecting athletes' motion quality. In order to further enhance its accuracy, applicability, and robustness, further research and optimization are needed in terms of technology improvement and model refinement.

This paper aims to enhance the accuracy and robustness of pose estimation in athlete motion quality assessment. The powerful image processing capabilities of Swin-UNet [3] have demonstrated excellent performance in fine-grained image segmentation, providing an innovative approach, especially in medical image analysis. Building on this, we propose an innovative pose evaluation network architecture called SwinUNetPose, which combines the Swin Transformer [4] backbone and the human segmentation module, enabling high accuracy while maintaining efficient real-time performance. This architecture's dual focus on accuracy and inference speed is crucial for real-time sports monitoring, where rapid inference is essential without sacrificing precision. To balance the trade-off between accuracy and inference speed in motion analysis, the key innovation of SwinUNetPose lies in offering multiple model configurations SwinUNetPose-T, SwinUNetPose-S, SwinUNetPose-B, and SwinUNetPose-L allowing users to choose the most suitable model based on specific application needs. For example, SwinUNetPose-T provides a faster, more lightweight solution, particularly suitable for applications where inference speed is critical, while SwinUNetPose-L offers the highest accuracy for scenarios where precision is a top priority. This flexibility allows SwinUNetPose to adapt to applications of varying complexities, providing strong technical support for sports health monitoring. Additionally, the segmentation module in SwinUNetPose plays a critical role in enhancing the model's robustness, especially in complex scenarios with occlusions or significant background noise. By removing background interference, the segmentation module provides clearer inputs for the pose estimation task, ensuring high reliability for real-time motion quality detection across various dynamic sports scenarios. The architecture excels in fine-grained image segmentation, effectively extracting multi-scale features and successfully separating the human region from complex backgrounds. This ability helps overcome the challenges posed by cluttered backgrounds, providing clearer inputs for subsequent pose estimation. The pose estimation detection head in SwinUNetPose is responsible for locating human skeletal keypoints and accurately predicting joint positions using deep convolutional networks. This method ensures accurate skeletal pose prediction, which is crucial for motion analysis. The key innovation of this paper lies in the shared feature information between the segmentation head and the pose estimation detection head. The segmentation network removes background noise and refines human contours, providing clearer input for pose estimation, while the pose estimation detection head optimizes the localization of skeletal keypoints, further guiding the optimization of the segmentation network. The synergy between these two components improves the accuracy and robustness of pose detection. By sharing features, SwinUNetPose enables end-to-end training and optimization, thus improving the model's generalization ability and precision. This method performs excellently in complex motion scenarios, especially in the presence of occlusion and significant background noise, providing a more reliable and accurate motion quality detection solution. In conclusion, SwinUNetPose offers a highly efficient and reliable motion quality detection solution for IoT-based sports health monitoring systems. Its flexibility and robustness in handling complex scenarios and real-time performance make it a valuable tool with significant practical application potential in sports health monitoring.

To summarize, The main contributions of this paper are as follows:

- Proposed SwinUNetPose Model: The model combines the Swin Transformer and U-Net architectures for efficient bottom-up human pose estimation. SwinUNetPose takes advantage of the Transformer Block to gather crucial data from feature maps at different scales, thereby increasing the precision and resilience of pose estimation.
- Introduction of Segmentation Module to Enhance Pose Estimation: The paper proposes an innovative method that combines human segmentation and pose estimation tasks. The addition of the segmentation module greatly improves the accuracy of pose estimation, particularly in recall rate (AR), showing substantial performance improvement compared to models without the segmentation head.
- Ablation Study Analysis: The paper carries out a set of ablation experiments to evaluate the performance from multiple dimensions, including input resolution, segmentation module, pretraining data, and pretraining methods. The results demonstrate that these designs significantly enhance model performance, validating the effectiveness of these innovative designs in improving model outcomes and proving the model's comprehensiveness and practicality.
- Comprehensive Comparison with Existing Methods: This study systematically compares the SwinUNetPose model with several existing bottom-up 2D human pose estimation methods, showing that it surpasses many of them in performance while keeping inference latency low.
- Model Scalability and Efficiency: SwinUNetPose demonstrates excellent scalability, maintaining strong performance across models of various sizes. In larger models (such as SwinUNetPose-L), it achieves high precision while effectively controlling inference time, proving the method's efficiency in practical applications.

2. Related work

Human Pose Estimation (HPE) aims to recognize and locate the positions of various body joints, such as the elbows, shoulders, knees and head from images or videos. The core objective is to reconstruct the human skeletal structure and analyze human actions from 2D or 3D data. HPE is typically classified into two categories: 2D HPE and 3D HPE. 2D HPE focuses on detecting the joint positions from single or multiple 2D images and producing 2D coordinates for each joint. This method is widely applied in areas like human–computer interaction, sports analysis, and behavior monitoring due to its computational efficiency and fast processing speed. However, 2D methods lack depth information, which limits their performance in scenarios with large depth variations or occlusions. In contrast, 3D HPE predicts joint positions in three-dimensional space, offering more precise pose information, especially in cases involving occlusions or spatial overlaps. The method is particularly applicable to virtual reality (VR), augmented reality (AR), and robotics. On the other hand, 3D HPE is computationally more demanding and requires larger datasets for training, which poses challenges for real-time performance and resource allocation. Given the real-time and computational efficiency needs for the application in this study, and considering that 2D HPE satisfies the accuracy requirements for most tasks, we chose to adopt the 2D HPE methods.

2D HPE focused on detecting the locations of human keypoints from 2D images and generate the corresponding 2D coordinates for each keypoint. Early 2D HPE methods mainly depended on conventional image processing approaches, like feature extraction and template matching. However, these methods had limited effectiveness and lower accuracy, especially when dealing with poor image quality or complex backgrounds. With deep learning gaining prominence, especially with the use of Convolutional Neural Networks (CNNs) [5–7] has led to substantial advancements in 2D HPE. Deep learning-based methods, by utilizing complex network architectures for feature extraction and joint localization, have significantly improved the accuracy and robustness of HPE. The success of these methods is attributed to deep learning's powerful ability to automatically learn and extract essential features from images. Based on deep learning, 2D HPE methods can be classified into single-person and multi-person pose estimation [8,9]. The single-person approach focuses on HPE for one individual, while the multi-person approach deals with multiple individuals.

2.1. 2D single-person human pose estimation

2D single-person human pose estimation primarily processes the input single-person image to identify and locate the keypoints of their body. If the input is an image containing multiple people, a human detector [10,11] is typically used to detect each individual in the image. The detected bounding boxes are then used to crop each person separately, and the HPE network takes these cropped single-person images to localize the keypoints. 2D single-person pose estimation is generally classified into regression methods and heatmap-based methods, depending on the deep learning techniques used [12,13]. The regression approach uses an end-to-end learning framework to directly convert input images into human joint positions or model parameters. This method trains a neural network model and uses a regression approach to predict the coordinates of each joint, thereby determining its actual position in space. In contrast, heatmap-based methods [14–16] estimate joint locations indirectly by predicting the heatmap for each joint. In the heatmap, each pixel indicates the probability of that position being a joint and the network learns by reducing the difference between the predicted and ground truth heatmaps, continuously refining the model's predictions. Fig. 1 illustrates the basic framework of 2D single-person pose estimation methods. Currently, heatmap-based frameworks have become the mainstream approach in 2D HPE tasks because they better preserve spatial information and make the training process smoother. In summary, these two methods have their own advantages: regression methods are typically simpler and more direct, while heatmap-based methods perform more stably in complex scenarios, especially when dealing with occlusions and dense human poses. Currently, heatmap-based methods play a significant role in 2D HPE.

2.1.1. Regression methods of human pose estimation

Previous research has focused on regression-based frameworks, which aim to directly regress human keypoints' coordinates from images, as shown in Fig. 1(a). For example, the DeepPose [17] method proposed by uses AlexNet as the backbone network to estimate keypoint locations. Due to its excellent performance, the research in HPE gradually shifted from traditional methods to deep learning, particularly CNN architectures. Subsequently, Sun et al. [18] proposed the "compositional pose regression" method,. This method based on the ResNet-50 [19] architecture and applies a reparameterized skeleton representation, which enables the simultaneous capture of body structure and pose, overcoming the shortcomings of conventional joint-based representations. Meanwhile, Li et al. [20] proposed a cascaded network based on Transformers, which uses the self-attention mechanism to capture the spatial relationships of joints and appearance features. They also introduced RLE to improve the optimization of joint location distribution, thereby improving prediction accuracy. In regression methods, extracting features that can encode rich pose information is crucial. In order to strengthen the model's capacity for generalization, multi-task learning (MTL) [21] has found extensive use in pose estimation. Li et al. [22] designed a heterogeneous framework that concurrently addresses joint position regression and sliding window-based body part recognition. Fan et al. [23] introduced a dual-source CNN approach, which further enhances regression accuracy through two tasks: joint detection and localization. These methods achieve more accurate pose estimation by sharing feature representations and combining the loss functions of different tasks.

2.1.2. Heatmap-based methods of human pose estimation

The heatmap-based method differs with traditional methods that directly regress joint coordinates. Instead, it estimates human pose by generating 2D heatmaps associated with each joint location. Specifically, the goal is to generate K heatmaps for the K keypoints, denoted as $\{H_1, H_2, \ldots, H_K\}$. The pixel intensity $H_i(x, y)$ in each heatmap signifies the probability of keypoint i being positioned at the coordinate (x, y). These heatmaps output the probability of the true location for each joint, which is generated by a 2D Gaussian kernel [24]. Fig. 1(b) illustrates its working principle. Compared to direct joint coordinate regression, heatmaps better maintain spatial location information, which improves the robustness of the training process. This helps mitigate the position accuracy issues that may arise in coordinate regression and makes the training smoother. During training, The model parameters are typically updated by minimizing the difference between the ground truth heatmaps and the predicted heatmaps, with MSE loss function being commonly employed. The method outputs joint locations while also capturing the spatial interactions between joints. Over the past decade, heatmap-based methods for HPE have received substantial interest, especially with the application of CNN architectures. Wei et al. introduced CPM [25], which refine keypoint predictions through multi-stage processing. Newell et al. proposed the "stacked hourglass" (SHG) [26] network, utilizing an encoder–decoder structure with bottom-up processing to capture multi-scale information. Building on SHG, Chu et al. designed Hourglass Residual Units (HRUs) [27] to enhance feature extraction across different scales, while Yang et al. introduced a Pyramid Residual Module (PRM) [28] to improve scale invariance. Sun et al. [29] presented High-Resolution Network (HRNet), which connects multi-resolution subnetworks in parallel to achieve more accurate keypoint heatmap predictions by fusing information across scales. Thanks to its outstanding performance, HRNet [29] and its variations [30,31] have gained widespread adoption in pose estimation and other related tasks.

2.2. 2D multi-person human pose estimation

2D multi-person HPE is more complex and challenging compared to 2D single-person HPE. It involves not only identifying the quantity of individuals, but also precisely detecting their keypoints and effectively grouping them in an image. To address these challenges, current approaches for 2D multi-person HPE are generally classified into two main categories. The method flow of top-down uses a two-stage process for HPE. First, all individuals in the image need to be detected, and their corresponding bounding boxes are obtained. Then, the image regions are cropped based on these bounding boxes. Subsequently, pose estimation is performed for each cropped region. Since

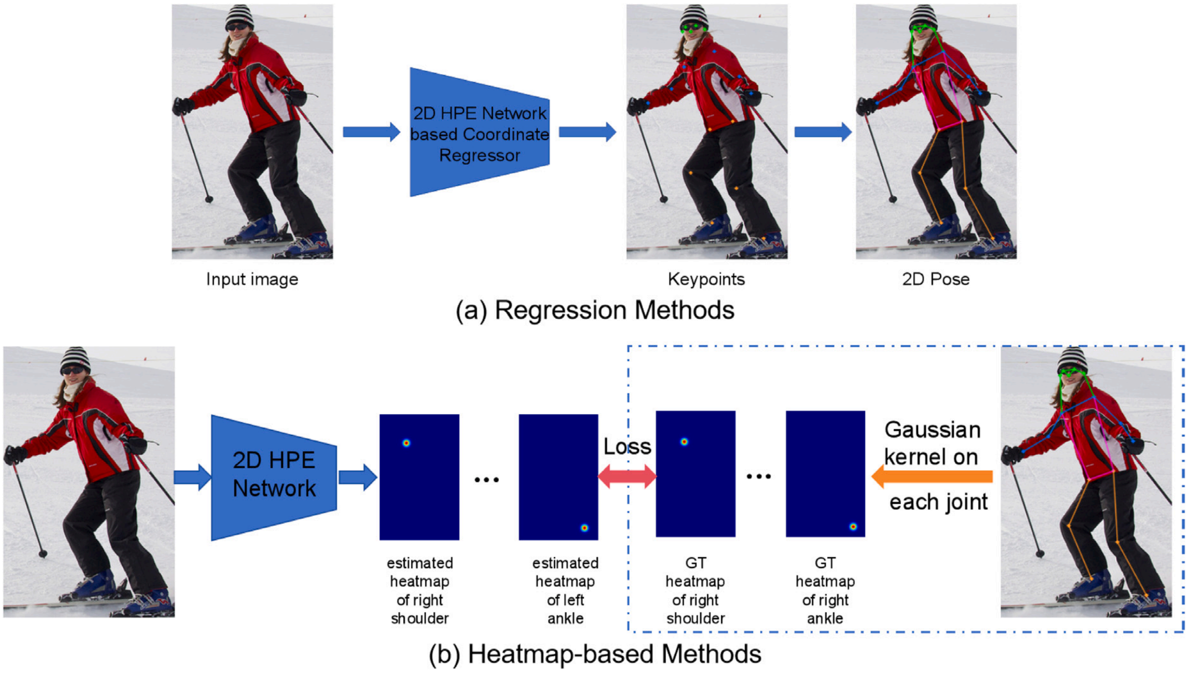

2D HPE Network based Coordinate Regressor
Input image
Keypoints
2D Pose
(a) Regression Methods
2D HPE Network
estimated heatmap of right shoulder
estimated heatmap of left ankle
Loss
GT heatmap of right shoulder
GT heatmap of right ankle
Gaussian kernel on each joint
(b) Heatmap-based Methods

Fig. 1. Single-person 2D human pose estimation framework. (a) The coordinate regression method directly predicts the joint coordinates through a network model. (b) Ground-truth heatmaps for each joint are generated by applying a Gaussian kernel to the location of each joint. Then, a deep learning model based on heatmaps is used to predict the heatmap for each joint.

this method processes each individual separately and estimates their pose independently, its computational cost increases as the quantity of individuals in the image grows. In contrast, The method flow of bottom-up adopts a one-stage framework for HPE which directly locating all keypoints in the image and grouping them appropriately. Since this method can simultaneously locate all the keypoints without the need for separate keypoint predictions for each individual, it is generally more efficient than the top-down approach, and its computational cost is less significantly influenced by the quantity of individuals in the image. The framework of multi-person HPE is shown Fig. 2.

2.2.1. *Top-down methods of human pose estimation*

The HPE framework of top-down methods is shown in Fig. 2(a), the top-down HPE framework consists of two main components. The first component employs a human detector to generate bounding boxes for individuals in the input image. The second component utilizes a 2D HPE network to predict the keypoint locations within each detected bounding box. Many studies have focused on optimizing and improving these modules in the HPE network to enhance model performance. For example, Xiao et al. [32] introduced a model that employs ResNet as the feature extractor and produces high-resolution heatmaps by deconvolution layers. By finding the maximum value in the heatmaps, the spatial locations of the joints are accurately localized. Fang et al. [33] proposed the AlphaPose framework, which predicts keypoints based on human body regions. The approach first processes the image within each person's bounding box using a multi-branch CNN, extracting features from different body parts. By combining the local features, the model generates comprehensive full-body pose information, and the joint locations are precisely determined by predicting the heatmaps for each joint. Wang et al. proposed Graph-PCNN [34], a two-stage HPE framework built on graph-based techniques. Firstly, a CNN model is used to extract features and generate joint heatmaps for coarse pose estimation. Secondly, a GCN model is introduced to refine the estimation by modeling spatial relationships between joints, improving the initial pose and correcting errors. Cai et al. introduced a multi-level network that integrates the RSN module with the PRM [35]. In this framework, the RSN module captures detailed local representations through an efficient feature fusion strategy, focusing on capturing the details of joint information. The PRM module, on the other hand, finds the optimal equilibrium between local and global features, optimizing their combination to boost the performance of pose estimation.

2.2.2. *Bottom-UP methods of human pose estimation*

The HPE framework of bottom-up methods is shown in Fig. 2(b), the bottom-up method directly detects the keypoints of each individual by processing the entire image, without relying on prior object detection boxes. First, convolutional neural networks (CNNs) or other deep learning models are utilized to create heatmaps corresponding to each joint, predicting the locations of the keypoints. Next, keypoint association techniques are employed to correctly assign the detected joints to each individual, especially in cases of multiple people or occlusions, using methods like graph optimization or graph convolutional networks to solve the association problem. Finally, the correctly associated keypoints are combined to form a complete pose, achieving multi-person pose estimation. Many studies have focused on optimizing and improving these modules in the HPE network to enhance model performance. For example, Pishchulin et al. proposed DeepCut [36], which is among the first two-stage bottom-up methods. This method first uses Fast R-CNN as the human detector to locate the bounding boxes of individuals in the image, and then utilizes a deep convolutional network to generate joint heatmaps, integrating both global and local features for keypoint detection. Next, a graph-cut optimization method is used to associate the joints, ensuring correct assignment of joints to each individual. Finally, a joint optimization strategy is employed to correct estimation errors, and multi-stage inference is applied to progressively refine the accuracy of pose prediction. In multi-person scenarios, DeepCut has a high computational cost. To overcome this difficulty, Cao et al. proposed OpenPose [37], which uses a two-branch network: One branch employs CPM to predict keypoint coordinates through heatmaps, while the other branch uses PAFs to capture the position and direction of limbs and properly associate keypoints with each individual. To further improve computational efficiency, Osokin et al. presented a simplified version of OpenPose called Lightweight OpenPose [38]. This approach uses a more lightweight backbone network, enabling real-time HPE even in CPU environments. In order to boost performance, HigherHRNet [39] introduced multi-stage supervision.

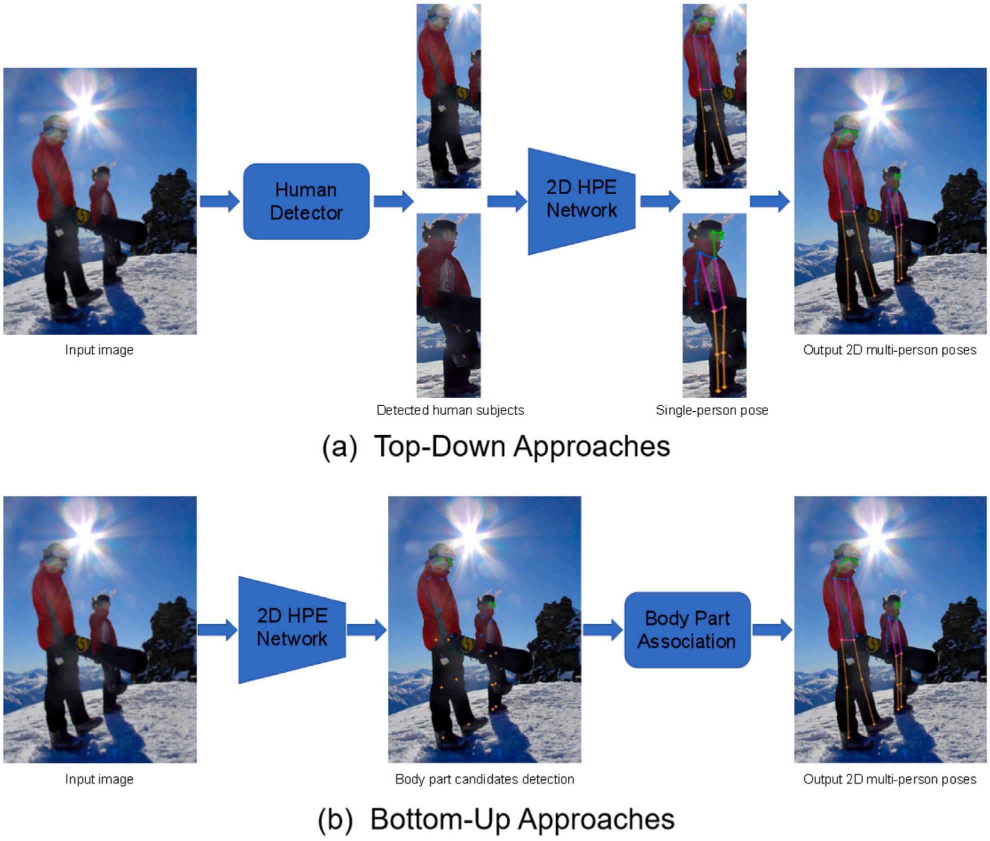


Fig. 2. Multi-person 2D HPE framework. (a) The method flow of top-down approaches for HPE. (b) The method flow of bottom-up approaches for HPE.

2.3. *Vision transformer methods of 2D human pose estimation*

Transformer-based methods have become more prevalent in various vision applications, mainly due to their effectiveness in capturing long-distance relationships and global context. In contrast to conventional CNN models, Transformer can effectively capture global features in images through its self-attention mechanism, boosting the model's capability to process complex visual contexts. Transformer has shown significant improvements in tasks like object localization, image classification, and pixel-level segmentation. Notably in tasks with long-range dependencies, such as capturing large-scale contextual information in images, Transformer performs better in capturing details and improving the performance of vision tasks. As a result, an increasing number of studies have started to combine Transformer with CNNs or use pure Transformer architectures for certain tasks, achieving outstanding results. In recent years, Transformer-based models have achieved remarkable advancements in the task of HPE. Many approaches have improved the model's ability to handle intricate situations by using Transformers as decoders in combination with CNN backbone networks. For instance, TransPose [40] combines the advantages of CNN and Transformers. First, CNN is used to extract low-level features from the input image. Then, the self-attention mechanism of the Transformer captures spatially distant dependencies within the image, effectively modeling the spatial relationships between joints. Finally, high-precision joint heatmaps are generated for accurate joint location prediction. TokenPose [41] segments the input image into patches and converts them into tokens, which are processed using the self-attention mechanism of Transformer. This allows it to model the spatial relationships between joints and capture intricate dependencies. By generating accurate joint heatmaps, it predicts each joint's location and associates joints using global information, resulting in precise multi-person pose estimation. Ultimately, TokenPose predicts the position of each joint via heatmaps and performs joint association using global information, achieving accurate multi-person pose estimation. While these techniques have demonstrated outstanding results on well-known pose estimation dataset, they remain dependent on CNNs for extracting features. Alternatively, HRFormer [31] innovatively combines high-resolution feature extraction with Transformer self-attention, leveraging HRNet for extracting high-resolution details and incorporating multi-scale feature fusion to boost the capture of intricate details. While modeling long-range dependencies between joints, HRFormer provides more precise joint position prediction, making it suitable for complex pose estimation tasks. Another Transformer-based model, ViTPose [42], uses a pure Transformer backbone for feature extraction and regresses keypoint heatmaps for pose estimation. ViTPose introduces multiple decoder designs and employs knowledge distillation and decomposition techniques to enhance its multi-task learning ability, resulting in stronger scalability and transferability. These Transformer-based pose estimation methods demonstrate significant advantages in performance over traditional CNN-based approaches. They excel at capturing long-range dependencies and global information, leading to improvements in model accuracy and robustness, especially when dealing with complex pose estimation tasks. Although existing methods mostly estimate keypoints by directly predicting or detecting human bodies, few studies have explored combining segmentation and pose estimation. To overcome this challenge, the paper proposes a multi-task learning approach that integrates instance segmentation and pose estimation, with mutual supervision enhancing the accuracy and generalization of the model.

2.4. *Segment methods of 2D human pose estimation*

Traditional pose estimation methods typically rely on a single pose prediction module, which often overlooks the impact of background interference in complex scenarios. While these methods perform well in simple environments, they tend to make errors in multi-person scenes or cluttered backgrounds, especially in applications with real-time requirements. To address these challenges, there has been widespread attention in recent years on multi-task learning methods that combine semantic segmentation and pose estimation. These methods first perform human segmentation to accurately extract human regions and then use these regions for pose estimation, thereby improving the accuracy of pose prediction and achieving better robustness in complex environments. For instance, Xia et al. [43] proposed an algorithm that jointly addresses human pose estimation and semantic part segmentation. This method combines pose estimation and part segmentation tasks, utilizing their complementarity to optimize overall performance in multi-person images. Pose estimation provides object-level shape priors for part segmentation, while part-level segmentation constrains the variation of pose locations, thus promoting mutual improvement. However, despite improving accuracy, the bottleneck in inference speed still limits its real-time application. Furthermore, optimization methods based on Mask R-CNN, such as Cai et al. [44], introduce MobileNet as the backbone network to optimize the inference speed of Mask R-CNN while maintaining high accuracy. The method also proposes using pixel segmentation results to assist in detecting human key points, further improving pose estimation accuracy, especially in complex scenes. While the optimization improves inference speed and reduces false detection rates, the use of a simpler model structure to predict segmentation components and optimize pose estimation still has its limitations, especially when dealing with complex backgrounds. Given the limitations of the aforementioned methods, this study draws inspiration from the success of Swin-UNet in medical image analysis and proposes an innovative approach. By combining the Swin Transformer with a human segmentation module, we are able to significantly improve pose estimation accuracy and robustness while maintaining efficient inference speed. The powerful image processing capabilities and flexible multi-scale feature extraction of Swin-UNet allow it to perform well in complex environments, significantly enhancing performance in both pose estimation and semantic segmentation tasks. Based on this, we introduce SwinUNetPose, a new algorithm that employs feature sharing and multi-task joint learning. By sharing information between segmentation and pose estimation tasks, SwinUNetPose optimizes both accuracy and robustness while ensuring real-time performance. This approach overcomes the limitations of traditional methods, providing a more efficient and robust solution for multi-person pose estimation, particularly in complex backgrounds and occluded scenarios, with improved adaptability and precision.

3. Method

3.1. SwinUNetPose network architecture

The SwinUNetPose model proposed in this paper combines the Swin Transformer and U-Net architecture, aiming to enhance the accuracy and robustness of human pose estimation (HPE) through multi-task mutual supervision learning. The architecture includes an encoder, neck layer, decoder, patch expanding layer, head layer, and skip connections, fully leveraging the benefits of the Transformer in feature learning. This design enables efficient multi-task learning, leading to notable improvements in both HPE and image segmentation tasks. The encoder relies on the Swin Transformer backbone, first dividing the input image into patches and generating patch tokens suitable for Transformer processing. These patch tokens pass through the Swin Transformer blocks, using the self-attention mechanism to capture global and local features. Meanwhile, the Patch Merging layers progressively downsample the feature map's resolution while increasing its feature dimension, effectively doubling the feature map's size compared to the original. After several stages of downsampling and feature dimension enhancement, the encoder produces deep feature representations with rich multi-scale information (see Fig. 3). The decoder adopts a symmetric design based on the U-Net architecture, using stacked Swin Transformer blocks for feature recovery and Patch Expanding layers for upsampling, progressively increasing the resolution of the feature map for effective image reconstruction. Each Patch Expanding layer first doubles the feature dimension and then expands the resolution by 2 times. This step-by-step upsampling enables the decoder to transform low-resolution feature maps into high-resolution ones, ultimately restoring them to the input image size ($W \times H$). This process compensates for the spatial information lost during downsampling and recovers fine-grained details of the image. The decoder's skip connections facilitate the effective fusion of multi-scale representations obtained in the encoder, ensuring spatial integrity and further enhancing the model's expressiveness and segmentation accuracy. The skip connections not only help the decoder recover spatial information lost during downsampling but also allow the model to effectively combine multi-scale features, enhancing its ability to restore fine details. At the final stage of the model, the upsampled features are processed by a segmentation head module to generate pixel-level segmentation maps. This process converts the restored high-resolution feature maps into actual pixel-level segmentation results, providing precise outputs for both HPE and image segmentation tasks. The SwinUNetPose model combines the precise spatial recovery abilities of U-Net with the strong feature extraction capabilities of Swin Transformer, effectively overcoming the limitations of CNNs in human pose estimation (HPE) and image segmentation. The encoder extracts multi-scale features, while the decoder gradually recovers the image resolution and compensates for the lost spatial information through Patch Expanding layers. The skip connections effectively fuse features from different scales, ultimately generating accurate pixel-level segmentation results. The model demonstrates strong performance in both HPE and image segmentation tasks, highlighting the advantages of combining U-Net and Transformer structures.

3.2. Swin Transformer block

To overcome the issues of high computational and memory requirements in traditional Transformers for high-resolution image processing, the Swin Transformer network proposes a window-based Transformer structure called the Swin Transformer Block [3]. By utilizing a windowed self-attention mechanism and a shifting window operation, this design effectively captures both local and global features while reducing computational complexity. The Swin Transformer block consists of multiple key components designed to efficiently capture both local and global features. Fig. 4 shows the architecture of Swin Transformer block, each block containing LN layer, MHSA module, MLP and residual connection. These two blocks apply different variants of the self-attention mechanism.

Here is the computation process for each consecutive Swin Transformer block:

1. **W-MSA**: The input feature z_{l-1} is first normalized using LayerNorm, followed by passing it through the W-MSA module. This module performs self-attention within local windows and returns the weighted features. These features are then added to the original input z_{l-1} via a residual connection, resulting in the output $\hat{z}_l$.

$$\hat{z}_l = \text{W-MSA}(\text{LN}(z_{l-1})) + z_{l-1}, \tag{1}$$

2. **MLP**: The output $\hat{z}_l$ is passed through an MLP layer for a nonlinear transformation, producing the feature z_l. A residual connection adds this back to $\hat{z}_l$ to produce the final output of this stage.

$$z_l = \text{MLP}(\text{LN}(\hat{z}_l)) + \hat{z}_l, \tag{2}$$

3. **SW-MSA**: The output from the MLP z_l undergoes a shifting operation and is passed through the SW-MSA module. This stage computes cross-window self-attention by shifting the windows to

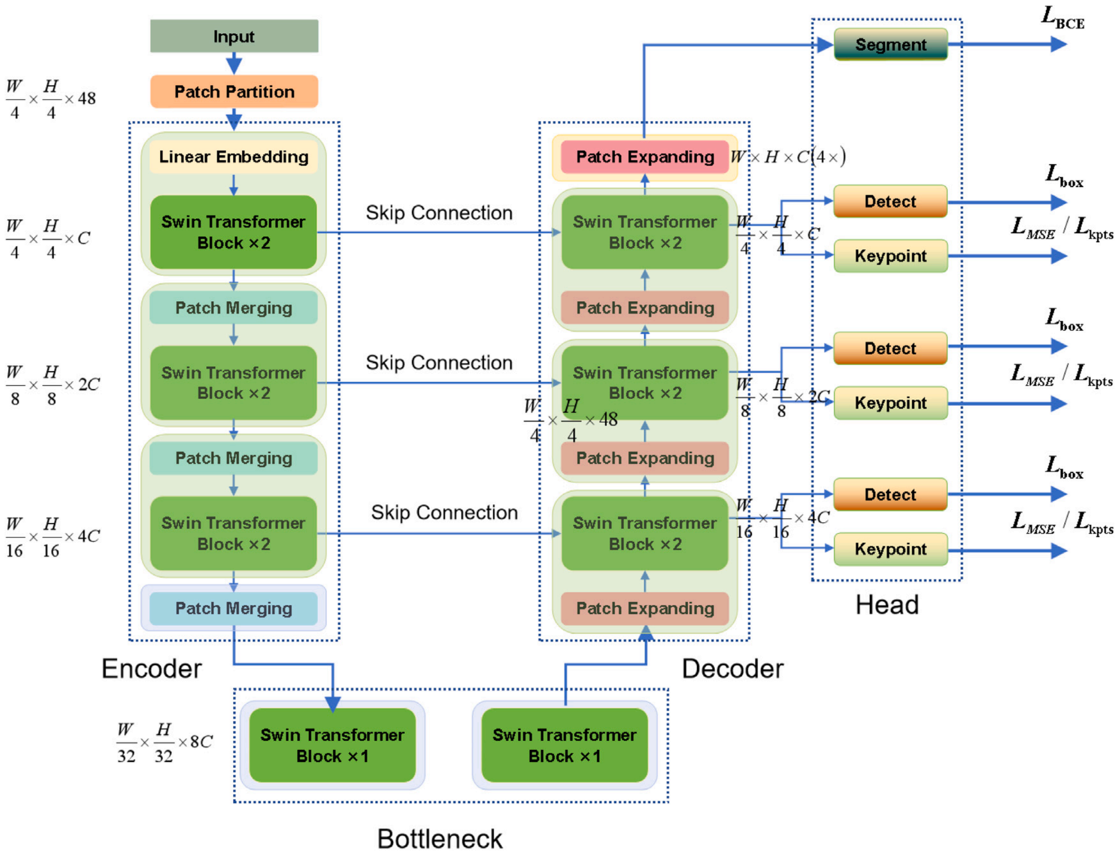

Input
Patch Partition
$\frac{W}{4}\times\frac{H}{4}\times 48$
Linear Embedding
Swin Transformer Block ×2
$\frac{W}{4}\times\frac{H}{4}\times C$
Patch Merging
$\frac{W}{8}\times\frac{H}{8}\times 2C$
$\frac{W}{16}\times\frac{H}{16}\times 4C$
Skip Connection
Patch Expanding
$W\times H\times C(4\times)$
$\frac{W}{32}\times\frac{H}{32}\times 8C$
Swin Transformer Block ×1
Encoder
Decoder
Bottleneck
Segment
Detect
Keypoint
Head
L_{BCE}
L_{box}
L_{MSE} / L_{kpts}

Fig. 3. The architecture of SwinUNetPose network.

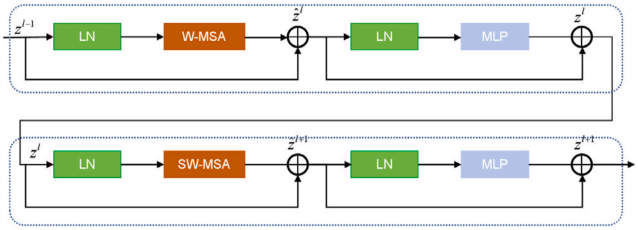
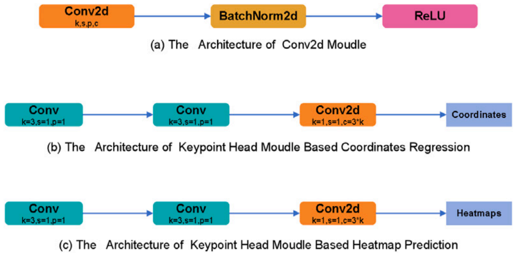


Fig. 4. Swin Transformer block.



(a) The Architecture of Conv2d Moudle

(b) The Architecture of Keypoint Head Moudle Based Coordinates Regression

(c) The Architecture of Keypoint Head Moudle Based Heatmap Prediction

Fig. 5. The architecture of keypoint prediction head.

cover a broader area. The result is added to the original input z_l producing the output $\hat{z}_{l+1}$.

$$\hat{z}_{l+1} = \text{SW-MSA}(\text{LN}(z_l)) + z_l, \tag{3}$$

4. **MLP**: Finally, the output from SW-MSA $\hat{z}_{l+1}$ is passed through another MLP layer for further transformation. A residual connection again adds the result to $\hat{z}_{l+1}$, yielding the final output z_{l+1}.

$$z_{l+1} = \text{MLP}(\text{LN}(\hat{z}_{l+1})) + \hat{z}_{l+1}, \tag{4}$$

The W-MSA and SW-MSA modules both rely on the attention mechanism, where the self-attention operation is defined by the following formula:

$$\text{Attention}(Q, K, V) = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d}} + B\right)V, \tag{5}$$

In this formula, Q, K, and V represent the query, key, and value matrices, respectively. d is the dimension of the query or key, and B is the bias matrix, which helps adjust the attention calculation.

The Swin Transformer Block introduces a novel approach combining local window-based self-attention with the strategy of shifted windows, enabling the model to effectively retain global context while mitigating the computational burden. This design significantly enhances the efficiency of the Swin Transformer, especially when handling high-resolution images, and facilitates its exceptional performance across a variety of vision tasks. By stacking multiple Swin Transformer Blocks, the model progressively learns hierarchical multi-scale features, transitioning from local to global representations. This process allows the network to capture both fine-grained details and the overarching global structure of the image, leading to a robust understanding of complex visual information.

3.3. Heads module

In this paper, the SwinUNetPose network architecture is proposed, where the head design improves the accuracy of keypoint prediction through mutual supervision and coordination across multiple tasks. To achieve this, the network integrates dedicated heads for each task, including detection head, segmentation head, and keypoint localization head, ensuring that each task is optimized in a complementary and synergistic manner to improve overall performance.

The detection and segmentation heads are inspired by the design of YOLOv10 [45], a highly efficient target detection algorithm that employs multi-task outputs for regressing the target box location, classification, and confidence. In this model, the detection head follows the YOLOv10 design, using convolutional layers to predict the coordinates, category, and confidence level for each bounding box. To adapt this for HPE, the detection head localizes the target and ensures that the bounding box is accurately extracted in scenarios involving multiple people. This approach leverages the proven efficiency and accuracy of YOLOv10 in object detection tasks. Similarly, the segmentation head adopts a similar design concept to YOLOv10, generating pixel-level classification outputs for image segmentation tasks. By predicting at the pixel level, the segmentation head effectively segments the image into regions and accurately labels human body regions, ensuring high-resolution, precise segmentation results.

The keypoint prediction head, which targets the human body pose estimation task, Fig. 5 illustrates consecutive of the regression method and the heatmap prediction method. Each method has its own strengths, explained as follows:

1. **Coordinates Regression Method**: This approach directly predicts the coordinates of keypoints through a regression network. The network treats the 2D coordinates of each keypoint as targets and refines the predicted positions through training. Its primary advantage is high computational efficiency since it directly outputs the predicted coordinates, which makes it ideal for real-time applications. Typically, this method is implemented using fully connected layers or convolutional neural networks (CNNs).
2. **Heatmap Prediction Method**: In this method, the network generates a 2D heatmap for each joint, where each heatmap corresponds to a specific joint, and each pixel signifies how likely it is to be the joint's exact location. The network learns by reducing the discrepancy between the predicted heatmap and the true keypoint locations, typically employing mean squared error loss. A major benefit of the heatmap method is its effectiveness in preserving spatial correlations among joints, which improves resilience to occlusion and multi-person settings, leading to more accurate localization.

In summary, the SwinUNetPose head design addresses the challenges of HPE and image segmentation by combining detection, segmentation, and keypoint prediction heads, all while leveraging YOLOv10 and a multi-task learning strategy. The design of each head ensures that tasks complement and enhance one another, greatly improving the model's performance and its potential for real-world applications.

3.3.1. Coordinates regression head module for human pose estimation

The regression method in HPE is typically implemented by directly predicting the coordinates of keypoints. The core idea is to regress the 2D coordinates of each keypoint through a network model. In regression methods, the objective of this approach is to minimize the error between predicted and ground truth keypoint positions, typically using Mean Squared Error (MSE) or L1 loss. A key advantage of this method is its high computational efficiency, as it directly outputs coordinate predictions without relying on intermediate heatmaps, making it well-suited for real-time applications. Fig. 5(b) shows the architecture of the coordinates regression method, which includes two convolutional modules and a 1×1 convolutional layer. The process is as follows:

1. **Feature Extraction**: The network first extracts deep features from the input image using two convolutional modules. Each convolutional module consists of convolutional layers, batch normalization, and activation functions, which together allow the network to capture both local and global features from the image effectively. After processing through these modules, the feature map size is $H \times W \times C$, where H and W represent the height and width of the feature map, and C is the number of channels.
2. **Keypoint Prediction**: A 1×1 convolutional layer then predicts keypoints within each bounding box. This layer independently determines both the coordinates and visibility of keypoints. Given that the network predicts K keypoints, the output feature map from the convolutional modules has dimensions $H \times W \times C$. After the 1×1 convolutional layer, the output becomes $H \times W \times (3K)$, where:
   - The first $2K$ channels correspond to the 2D coordinates (x, y) of each keypoint.
   - The remaining K channels are used to predict the visibility flag for each keypoint, indicating whether the keypoint is visible.

This structure allows the model to simultaneously perform two tasks: regressing the position of each keypoint and classifying the visibility of each keypoint. Specifically, the regression task predicts the accurate location of the keypoints, while the visibility prediction task determines whether the keypoint is occluded or invisible. Overall, this regression algorithm structure is simple and efficient, directly outputting the coordinates and visibility information of keypoints, making it ideal for real-time HPE. Furthermore, the regression method does not rely on complex intermediate heatmaps or probability maps, improving computational efficiency and maintaining robustness in complex environments.

The OKS (Object Keypoint Similarity) metric is essential for assessing the accuracy of keypoint detection. Traditional detection methods often rely on L1 loss for keypoint prediction, but this method does not always effectively optimize OKS and fails to fully consider the differences in object scale or keypoint type, which limits its effectiveness. In regression methods, OKS, as a more precise evaluation metric, can be directly optimized instead of serving as a simple surrogate loss function. The core advantage of this approach lies in calculating the loss by comparing the predicted keypoint positions with the true locations, which greatly enhances the accuracy of pose estimation. Unlike traditional IoU loss, OKS loss is extended to handle keypoints, directly computing the loss based on the similarity between the predicted and true positions of each keypoint. Moreover, OKS loss is scale-invariant, meaning it can adapt to objects of varying sizes. In addition, OKS assigns different weights to different types of keypoints. Specifically, Keypoints on the head incur a higher penalty for the same error, while keypoints on the body are penalized to a lesser extent. This weighting mechanism makes OKS loss more suitable for HPE, especially in multi-person scenarios, where keypoints on the head and torso play a more significant role in determining the overall accuracy of pose estimation. Furthermore, OKS loss effectively avoids the vanishing gradient problem that occurs with traditional IoU loss when there is no overlap, making OKS loss more stable and efficient during training. Similar to dIoU loss, OKS loss is more robust, particularly in handling complex backgrounds or occlusion scenarios. In practical applications, each bounding box stores the corresponding pose information. When a ground truth box aligns with an anchor, the model predicts the keypoint positions relative to the anchor's center. The OKS loss is computed for each keypoint individually and then combined to derive the final OKS or keypoint IoU loss. This enables the network to optimize both the regression of keypoint positions and the prediction of keypoint visibility, ultimately improving the overall pose estimation accuracy. Their calculation formulas are given by Eqs. (6) and (7).

$$L_{\text{kpts}}(s,i,j,k) = 1 - \sum_{n=1}^{K} \text{OKS} = 1 - \frac{\sum_{n=1}^{K} \exp\left(\frac{d_k^2}{2s^2 w_k^2}\right)\delta(v_k > 0)}{\sum_{n=1}^{K}\delta(v_k > 0)}, \tag{6}$$

where d_k is the Euclidean distance between the predicted keypoint position and the ground truth location for the kk-th keypoint. w_k is the weight associated with the kth keypoint, and ss represents the object scale. The function $\delta(v_k)$ indicates whether the kth keypoint is visible (if $v_k > 0$). This formula helps to compute the keypoint loss based on the predicted positions and the ground truth, factoring in visibility and weight.

Each keypoint is associated with a confidence parameter that indicates its presence on the target. A higher confidence reflects a greater probability of the keypoint being visible. The visibility flags serve as ground truth, aiding in the model's training to determine both the existence and visibility of keypoints. Thus, the confidence parameter not only denotes the presence of a keypoint but also its likelihood of being visible.

$$L_{\text{kpts\_conf}}(s,i,j,k) = \sum_{n=1}^{K} \text{BCE}(\delta(v_k > 0), p_{\text{kpts}}^k), \tag{7}$$

where, p_{kpts}^k represents predicted confidence of the kth keypoint. The loss at a given location (i,j) is considered valid for an anchor at scale ss when the ground truth bounding box is matched to that anchor. The total loss is then computed by summing the individual losses across all scales, anchors, and locations, as described in Eq. (8).

$$L_{\text{total}} = \sum_{s,i,j,k} (\lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{box}} L_{\text{box}} + \lambda_{\text{kpts}} L_{\text{kpts}} + \lambda_{\text{kpts\_conf}} L_{\text{kpts\_conf}}), \tag{8}$$

3.3.2. Heatmap prediction head module for human pose estimation

The heatmap-based keypoint prediction approach is commonly applied in HPE, particularly in top-down and bottom-up frameworks. Its fundamental concept involves creating a heatmap for every keypoint to indicate its position. Each pixel value in the heatmap indicates the confidence of that position being the keypoint, reflecting the likelihood of the position being the keypoint. Unlike regression-based methods, heatmap methods predict keypoint locations indirectly by generating a probability distribution for each keypoint's position, rather than directly regressing the coordinates. The advantage of this approach is that it effectively handles challenges such as keypoint location ambiguity, occlusion, and resolution mismatches. Fig. 5(c) illustrates the architecture of heatmap prediction method. The proposed heatmap prediction head includes two convolutional modules, followed by a 1 × 1 convolution layer and a sigmoid function, which together calculate the confidence for each pixel position. The entire process can be divided into several key stages:

1. **Feature Extraction**: First, The network obtains deep features from the input image by passing it through two convolutional modules. Each convolutional module is made up of a convolutional layer followed by a batch normalization layer and an activation function, such as ReLU. These components efficiently capture both fine-grained and broader contextual features from the image, enabling the network to recognize key spatial patterns. After passing through these two convolutional modules, the resulting feature map has a size of $H \times W \times C$, where H and W represent the feature map's height and width, and C denotes the number of channels.
2. **Keypoint Heatmap Prediction**: Next, The network employs a 1 × 1 convolutional layer to produce a heatmap corresponding to each keypoint type. The 1 × 1 convolutional layer's role is to map the features of each position to a corresponding confidence value that represents the probability of that position being a keypoint, rather than directly outputting a feature map. The key step here is that the 1 × 1 convolution processes each pixel independently, thus generating a heatmap for each keypoint type. Suppose the network needs to predict K keypoints, then after the two convolutional modules, the output feature map has dimensions of $H \times W \times C$. Once the feature map goes through the 1 × 1 convolutional layer, the output map is resized to $H \times W \times K$, with each channel representing the heatmap for a particular keypoint.
3. **Confidence Prediction**: To further process the probability distribution in each heatmap, the network uses a sigmoid function to calculate the confidence of each pixel position. The output of the sigmoid function ranges from [0,1], representing the confidence that the position is the keypoint. Therefore, the network not only outputs the keypoint location but also provides a confidence score for each position, indicating the likelihood of that position being the keypoint.

In HPE, since the task requires predicting heatmaps, the training data typically provides the actual locations of the keypoints. Therefore, it is necessary to generate the labels used for training from the task's annotation data. Specifically, for each type of keypoint, we generate K heatmaps $\{h_1, \dots, h_K\}$, where the ith heatmap h_i corresponds to the heatmap created based on the coordinates of the ith keypoint of the human body. In practical scenarios, since an image may contain multiple individuals, the ith heatmap h_i contains the heatmaps $\{h_{i1}, \dots, h_{iP}\}$ generated by the ith keypoint's coordinates for all P individuals. To handle multiple individuals, a common approach is to take the average or the maximum of these P heatmaps to combine the heatmaps of multiple targets. In our work, we adopt the maximum method, which is expressed by the following formula:

$$h_i(x,y) = \max_{j=1}^{P} h_{ij}(x,y) = \max_{j=1}^{P} \exp\left(\frac{-(x-x_{ij})^2 + (y-y_{ij})^2}{2\sigma^2}\right), \tag{9}$$

where (x_{ij}, y_{ij}) refers to the true position of the ith keypoint for the jth individual. The Gaussian kernel size is adjusted according to the image resolution and keypoint characteristics. A larger value of σ results in a smoother heatmap, while a smaller σ creates a more concentrated heatmap. In this study, σ is set to 6.

The training process centers on calculating the loss function. As the heatmap method aims to minimize the difference between the predicted and true heatmaps, the loss function is generally based on pixel-wise mean squared error (MSE). For each keypoint's heatmap, the network's output heatmap is compared pixel by pixel to the ground truth heatmap, and the error is computed. This error is quantified by the loss function, which guides the network in refining its predictions. The formula for the loss function is:

$$L_{\text{MSE}} = \frac{1}{K/m/n} \sum_{i=1}^{K} \sum_{y=1}^{m} \sum_{x=1}^{n} \left[h_i(x,y) - \hat{h}_i(x,y)\right]^2 \tag{10}$$

where m and n represent the heatmap dimensions, and K denotes the count of keypoints.

4. Experiment

4.1. Experimental environment

An experimental platform was set up to validate the effectiveness of the proposed method. The platform's hardware includes NVIDIA Tesla

Table 1
Configuration and experimental environment.

Environmental parameter	Value
GPU	NVIDIA Tesla V100 ×8
CPU	Intel Core i7-11800H
Operating system	Ubuntu 20.04 LTS
RAM	256 GB
Deep learning framework	PyTorch 1.13.1
Programming language	Python3.9

V100 GPU ×8, an Intel Core i7-11800H CPU, 256 GB RAM, offering substantial computational power. The software environment consists of Ubuntu 20.04 LTS, the programming language is Python 3.9, and the deep learning framework used is PyTorch 1.13.1, ensuring the efficiency and stability of the experiments. This platform provided the necessary support for validating the method effectively. The detailed experimental environment configuration is presented in Table 1.

4.2. Datasets

In the field of pose estimation, there are several commonly used evaluation datasets, each with different focuses in terms of tasks, scene complexity, and annotation content. MS COCO [46], MPII [47], AIC [48], COCO-W [49], OCHuman [50], and Interhand2.6M [51] are important datasets widely used in HPE and related research. Specifically, MS COCO is a widely recognized multi-task benchmark dataset in computer vision. It is suitable for complex pose estimation tasks, supporting occlusion, multi-scale, and dense scenes. Although it has limitations in 3D information and fine-grained action annotations, the COCO-W subset supplements the details of hands and feet, enhancing multi-task pose estimation capabilities. The MPII dataset focuses on complex daily activities and sports actions, providing annotations for 16 keypoints, making it suitable for handling dynamic blurring and occlusion scenes, with high annotation accuracy. The AIC dataset is designed for large-scale, densely packed multi-person scenes, with massive data and fine-grained occlusion annotations, making it suitable for pose estimation under complex lighting conditions, although it lacks hand and face annotations. The OCHuman dataset focuses on severe occlusion scenarios and provides fine-grained occlusion annotations, making it an important benchmark for evaluating occlusion-resistant algorithms. Interhand2.6M provides high-precision hand pose data, suitable for applications like VR hand gesture interaction and surgical robot control. Considering that our research mainly focuses on athlete motion quality monitoring with relatively simple scenes and employs multi-task learning to enhance algorithm robustness, requiring segmentation data for human instances, MS COCO is chosen as the ideal evaluation benchmark for our algorithm research due to its rich scene variety, diverse pose variations, and high-precision annotations.

MS COCO is a well-known benchmark dataset in computer vision, widely used for various tasks, including object detection, instance segmentation, and human pose estimation (HPE). It serves as a standard multi-task dataset for evaluating and advancing vision-based models. The dataset contains over 200K images, with annotations for approximately 250K human instances, each labeled with 17 keypoints, covering key body parts like the head, torso, and limbs. MS COCO supports complex pose estimation tasks, handling images with occlusion, multi-scale, and dense scenes, making it suitable for real-world pose estimation challenges. Key features of MS COCO include:

- **Scene Diversity**: The dataset includes images from both indoor and outdoor environments, featuring various lighting conditions, occlusions, and diverse backgrounds, testing pose estimation models in complex settings.
- **High-Precision Annotations**: The keypoint annotations have an error margin of less than 5 pixels, ensuring high-quality data, ideal for training precise pose estimation models.
- **Multi-Person Pose Annotations**: MS COCO supports multi-person pose estimation, enabling the algorithm to handle and differentiate multiple instances within the same scene.
- **Evaluation Standards**: The dataset uses the OKS (Object Keypoint Similarity) metric and the mAP (mean Average Precision) standard to evaluate the performance of pose estimation models, offering comprehensive benchmarks for model accuracy.

Overall, MS COCO's broad application scenarios and high-quality annotations make it a standard dataset in the field of HPE, especially for multi-task learning and pose recognition in complex environments.

4.3. Performance evaluation metrics

In this study, to thoroughly evaluate the model's accuracy and computational complexity, a series of widely used pose evaluation metrics were employed. In terms of detection performance, the core metrics include Average Recall (AR) and Average Precision (AP), which are used to measure the model's detection effectiveness. We use model parameters and FLOPs as evaluation criteria in terms of computational complexity. With these metrics, we can comprehensively analyze and assess the model's performance in terms of detection effectiveness and resource consumption. When calculating AR and AP, we first review the calculation of the OKS (Object Keypoint Similarity) metric in pose evaluation, which is crucial for understanding the computation of AP and AR.

OKS (Object Keypoint Similarity) is a metric used to evaluate the performance of HPE models, particularly suited for scenarios involving multiple complexities such as occlusion and multi-person scenes. It evaluates how closely the predicted keypoints align with the ground-truth keypoints by factoring in not only considering the Euclidean distance but also incorporating factors like object scale and normalization terms, thereby improving evaluation accuracy. The formula for OKS is defined as:

$$\mathrm{OKS} = \frac{\sum_{i=1}^{K} \exp\left(-\frac{d_i^2}{2s^2\sigma_i^2}\right) \cdot \delta(v_i > 0)}{\sum_{i=1}^{K} \delta(v_i > 0)}, \tag{11}$$

In multi-person pose estimation, there may be multiple people in a single image. In this case, the OKS formula for a particular person in the image can be written as follows:

$$\mathrm{OKS}_p = \frac{\sum_{i=1}^{K} \exp\left(-\frac{d_{pi}^2}{2s_p^2\sigma_i^2}\right) \cdot \delta(v_{pi} > 0)}{\sum_{i=1}^{K} \delta(v_{pi} > 0)}, \tag{12}$$

In the formula:

- p: Individual person in the image
- d_{pi}: Euclidean distance between the ith detected keypoint and the ith ground truth keypoint of the p person
- σ_i: Normalization factor for the ith keypoint, compensating for scale variations across body parts, computed from annotation errors in training data. The normalization factors for the 17 keypoints in the COCO dataset are computed based on 5000 samples. The values for these keypoints are as follows: Eyes: 0.025, Nose: 0.026, Ears: 0.035, Wrists: 0.062, Elbows: 0.072, Shoulders: 0.079, Knees: 0.087, Ankles: 0.089, and Hips: 0.107. These factors are used to account for the scale differences of various body parts during pose estimation.
- s_p: s_p represents the scale factor of the p person in the Ground Truth (GT). The value is determined by the area of the bounding box around the person, $s_p = \sqrt{wh}$ calculated as the product of the width w and height h of the bounding box
- v_{pi}: Visibility flag for the keypoint:
  - $v_{pi} = 0$: Keypoint is invisible
  - $v_{pi} = 1$: Keypoint is annotated but invisible

  - $v_{pi} = 2$: Keypoint is visible and annotated
- $\delta(v_{pi} > 0)$: Indicator function that evaluates the keypoint iff $v_{pi} > 0$, otherwise ignored

AP (Average Precision) is a key metric for evaluating the effectiveness of detection algorithms, widely utilized in object detection and pose estimation. AP evaluates model performance by calculating precision at different thresholds. In object detection, AP is typically defined by calculating various IoU (Intersection over Union) thresholds, whereas in pose estimation, AP is derived using OKS (Object Keypoint Similarity). For single-person pose estimation, AP is calculated by computing the OKS similarity. After setting a threshold T, the average precision of all persons in the image is computed. The formula is as follows:

$$AP = \frac{\sum_p \delta(\mathrm{oks}_p > T)}{\sum_p 1}, \tag{13}$$

where p represents the person in the image, oks_p is the OKS value for person p, and $\delta(\mathrm{oks}_p > T)$ serves as an indicator function that outputs 1 if oks_p is greater than the threshold T, and outputs 0 otherwise. For multi-person HPE, the calculation of AP depends on the detection method used. If the top-down method is employed, the calculation of AP is similar to that of single-person HPE. If the bottom-up method is used, first all keypoints are detected, then they are grouped to form complete persons. The OKS between each Ground Truth person and the predicted persons is calculated, and an $M \times N$ matrix is obtained, in which each row represents the OKS values between a GT person and all predicted persons. The maximum OKS value from each row is taken as the OKS for that GT person. Finally, AP is calculated as:

$$AP = \frac{\sum_m \sum_p \delta(\mathrm{oks}_p > T)}{\sum_m \sum_p 1}, \tag{14}$$

where m represents each Ground Truth person, oks_p is the OKS value for each person, and $\delta(\mathrm{oks}_m > T)$ is the indicator function. In the calculation of AP (Average Precision), the AP is calculated as $OKS = 0.50 : 0.05 : 0.95$, where OKS starts from 0.50 and increases by 0.05 until it reaches 0.95, averaging over all OKS values in this range. Specifically, AP_{50} corresponds to OKS = 0.50, and AP_{75} corresponds to OKS = 0.75. In datasets, targets are categorized based on their size into small, medium, and large objects. Small objects have an area of $area < 32^2$, large objects have an area of $area > 96^2$, and medium-sized objects fall between these two categories. The AP values are calculated separately for each size category: AP_S for small targets, AP_M for medium targets, and AP_L for large targets. These distinctions enable a more granular assessment of how the model performs on various scales of objects these distinctions enable a more granular assessment of how the model performs on various scales of objects. By using OKS as a precision evaluation standard, AP can comprehensively assess the model's performance in different test scenarios.

In pose estimation, AR (Average Recall) is an important metric used to measure the recall rate of detection results. It primarily reflects the missed detections in the model's output. AR is calculated by determining the ratio of true positive detections to all true positive samples in the dataset. Specifically, the AR formula is:

$$\mathrm{Recall} = \frac{TP}{TP + FN}, \tag{15}$$

where TP (True Positives) represents the correctly detected positive samples, and FN (False Negatives) represents the missed positive samples. AR measures the proportion of true positive samples detected by the model out of the total number of true positive samples. During evaluation, AR can be calculated for different OKS thresholds, such as AR_{50}, AR_{75}, AR_S, AR_M and AR_L corresponding to different OKS values (e.g., 0.50 and 0.70) and for small, medium, and large objects. These AR metrics help to evaluate the model's recall ability under various conditions, particularly for different target scales, providing insight into the model's performance in multiple scenarios.

Additionally, the relationship between AR and AP indicates that while AR focuses on the model's ability to recall correct targets, AP mainly evaluates the model's precision. Together, AR and AP provide a comprehensive assessment of the model's performance.

4.4. Implementation details

SwinUNetPose adopts a standard top-down architecture for human pose estimation (HPE), using SwinUNetPose to estimate keypoints for each instance. To build the model, we use Swin-T, Swin-S, Swin-B, and Swin-L [4] as the backbone networks, naming these models SwinUNetPose-T, SwinUNetPose-S, SwinUNetPose-B, and SwinUNetPose-L. All backbone networks are initialized with MAE [52] pre-trained weights to ensure that the model starts with good performance. During training, we train the models on 8 V100 GPUs based on the MMPose codebase, following common practices from HigherHRNet [39], with an input resolution of 640 × 640. The training uses the AdamW [53] optimizer with an initial learning rate of 1e−3, and the models are trained for 300 epochs. During training, each mini-batch randomly samples 128 images with a 500-step linear warm-up. The learning rate is reduced by a factor of 10 at the 150th and 250th epochs to ensure better convergence in the later stages of training. In addition, we perform layer-wise learning rate decay and stochastic drop path ratio tuning for each model, and the optimal training settings are obtained through experiments. The training loss uses the same loss function and hyperparameters as YOLOx, Four hyperparameters were selected $\lambda_{cls} = 0.5$, $\lambda_{box} = 0.75$, $\lambda_{kpts} = 12.0$, and $\lambda_{kpts_conf} = 1.0$. as shown in Table 2. These settings and training strategies ensure efficient performance of SwinUNetPose in HPE tasks. GPU latency is tested on an NVIDIA Tesla V100 using ONNXRuntime and TensorRT with half-precision floating-point (FP16) format.

4.5. Ablation studies of SwinUNetPose and analysis

4.5.1. The influence of different heads module

Based on the experimental results, we conducted an ablation study of the backbone and head modules of SwinUNetPose on the MS COCO validation set, comparing the performance of different network architectures and head modules. Two head modules were used in the experiment: keypoint regression head and heatmap prediction head. When compared with SimpleBaseline (which uses the ResNet backbone), the results show that when using the keypoint regression head, the performance of SwinUNetPose remains stable, whereas SimpleBaseline, particularly with ResNet-50 and ResNet-101, shows a drop of about 6 in AP. This indicates that SwinUNetPose, using a vision transformer backbone, can maintain strong performance with the keypoint regression head, with only a small decrease in AP. Looking at metrics such as AP, AP_{50}, AR, and AR_{50}, it is evident that as the model size increases, SwinUNetPose consistently improves in performance, with the heatmap prediction head showing better results. For example, SwinUNetPose-L achieves the highest AP of 74.2 with the heatmap prediction head, compared to 73.7 with the keypoint regression head. Additionally, the heatmap prediction head shows a noticeable improvement in overall performance, especially in large models, indicating that it handles pose estimation tasks more effectively. Furthermore, the experiments also demonstrate that SwinUNetPose, using a vision transformer backbone, excels at encoding linearly separable features, reducing the need for complex decoders, which significantly improves performance in pose estimation tasks. Overall, SwinUNetPose shows a significant performance advantage over the traditional ResNet architecture when using larger vision transformer networks and the heatmap prediction head, with stable performance across different model sizes, confirming its superior performance and excellent scalability in pose estimation tasks (see Table 3).

Table 2
Hyperparametric configuration for SwinUNetPose models.

Model	Image size	Batch size	Optimizer	Learning rate	Weight decay	Layer wise decay	Drop path rate
SwinUNetPose-T	640 × 640	64	AdamW	1e−3	0.1	0.85	0.10
SwinUNetPose-S	640 × 640	64	AdamW	1e−3	0.1	0.80	0.10
SwinUNetPose-B	640 × 640	64	AdamW	1e−3	0.1	0.75	0.30
SwinUNetPose-L	640 × 640	64	AdamW	1e−3	0.1	0.85	0.50

Table 3
Ablation study of the backbone and head in SwinUNetPose on the MS COCO val set.

Method	ResNet-50		ResNet-101		SwinUNetPose-T		SwinUNetPose-S		SwinUNetPose-B		SwinUNetPose-L	
Head	Coordinates	Heatmap	Coordinates	Heatmap	Coordinates	Heatmap	Coordinates	Heatmap	Coordinates	Heatmap	Coordinates	Heatmap
AP	57.3	63.4	58.5	64.2	70.5	70.9	71.6	72.7	73.2	73.8	73.7	74.2
AP_{50}	84.5	87.2	85.2	87.7	90.1	90.5	91.1	91.9	91.7	92.3	92.4	92.8
AR	67.1	72.8	70.9	73.2	74.2	75.1	78.4	78.7	79.5	80.7	80.8	81.1
AR_{50}	92.3	92.7	92.5	92.9	93.1	93.4	93.5	93.9	94.0	94.2	94.1	94.3

Table 4
The performance of SwinUNetPose-S with different pre-training dataset on the MS COCO val set.

Pre-training dataset	Dataset volume	AP	AP_{50}	AP_{75}
ImageNet-1k	1M	72.6	91.9	80.9
MS COCO	150 K	71.6	91.2	79.3
MS COCO+AIC	500 K	72.7	91.9	81.0

Table 5
The performance of SwinUNetPose-S with different pre-training methods on the MS COCO val set.

	Random	DeiT	MoCov3	MAE
AP	70.3	70.7	70.5	72.6

4.5.2. The influence of pre-training data

To evaluate the role of ImageNet data in pose estimation tasks, we pre-trained the Transformer backbone on different datasets, including ImageNet-1k, MS COCO, and a combination of MS COCO and AIC. The model was pre-trained for 300 epochs on each of these datasets. Since the annotation types of the COCO and AIC data combination differ, we shared the backbone network during training and used different prediction methods for each dataset according to the task. After pre-training, the backbone model was fine-tuned for 200 epochs on the MS COCO dataset with pose annotations. The experimental results are summarized in Table 4.

The results show that when pre-training with the combination of MS COCO and AIC data, the performance of SwinUNetPose is comparable to that achieved with ImageNet-1k, while the dataset size is only half of ImageNet-1k. This indicates that pre-training on downstream task data leads to higher data efficiency, validating the flexibility of SwinUNetPose in choosing pre-training data. However, when only MS COCO data is used for pre-training, the AP value drops by 1.1, likely due to the smaller size of the MS COCO dataset, which has only a third of the instances of the MS COCO and AIC combination. Overall, these results further demonstrate that when the pre-training and fine-tuning data come from the same type of task, SwinUNetPose can more efficiently leverage data during the pre-training stage.

4.5.3. The influence of pre-training methods

According to the results in Table 5, we evaluated the performance of SwinUNetPose-S using different pre-training methods on the MS COCO validation set. The experiment compared four pre-training methods: random initialization, DeiT [54], MoCov3 [55], and MAE [52]. The results show that with random initialization, DeiT, and MoCov3, the model achieved AP values of 70.3, 70.7, and 70.5, respectively, with similar performance and slight variations. However, when pre-trained with MAE, the model achieved the highest AP of 72.6, indicating a significant performance improvement with masked image pre-training.

Table 6
The performance of SwinUNetPose-S with different input resolutions on the MS COCO val set.

	480 × 480	640 × 640	960 × 960	1280 × 1280
AP	68.1	72.7	73.6	74.1
AR	77.2	78.7	79.7	80.1

These results suggest that although supervised pre-training methods (like DeiT) and contrastive self-supervised pre-training methods (like MoCov3) perform well in pose estimation tasks, masked image pre-training (like MAE) adapts better to the task, significantly improving model accuracy. Therefore, masked pre-training provides stronger feature learning capability, demonstrating its superiority for downstream tasks.

4.5.4. The influence of input resolution

To investigate the impact of different input resolutions on SwinUNetPose, we designed a series of comparison experiments, using different input sizes during model training, and summarized the results in Table 6. Four different input resolutions were used in the experiment: 480 × 480, 640 × 640, 960 × 960, and 1280 × 1280. As the input resolution increased, both AP and AR metrics showed significant improvement. Specifically, with the 480 × 480 resolution, the model achieved AP of 68.1 and AR of 77.2; whereas, when the input resolution increased to 1280 × 1280, AP reached 74.1 and AR reached 80.1, showing a notable enhancement. These results indicate that as the input resolution increases, SwinUNetPose-S is able to capture more image details, thus improving pose estimation accuracy and recall. However, this also comes with higher computational costs, suggesting that a balance needs to be struck between performance and computational efficiency depending on the specific application scenario. Overall, higher input resolutions help improve model performance, especially in tasks requiring higher accuracy.

4.5.5. The influence of segment head

In this paper, we introduce a segmentation module into the SwinUNetPose architecture to simultaneously perform human segmentation and enhance the robustness of pose estimation. To evaluate the impact of the segmentation module on the pose estimation task, we conducted a comprehensive comparative analysis, comparing the performance of SwinUNetPose with and without the segmentation module. The experimental results are shown in Table 7, and the results indicate that adding the segmentation head significantly improves both the Average Precision (AP) and Average Recall (AR) metrics. For example, in SwinUNetPose-T, when using the segmentation head, AP increased from 69.2 to 70.9, and AR increased from 71.7 to 75.1. Similarly, for SwinUNetPose-S, SwinUNetPose-B, and SwinUNetPose-L, the performance also improved after using the segmentation head, especially

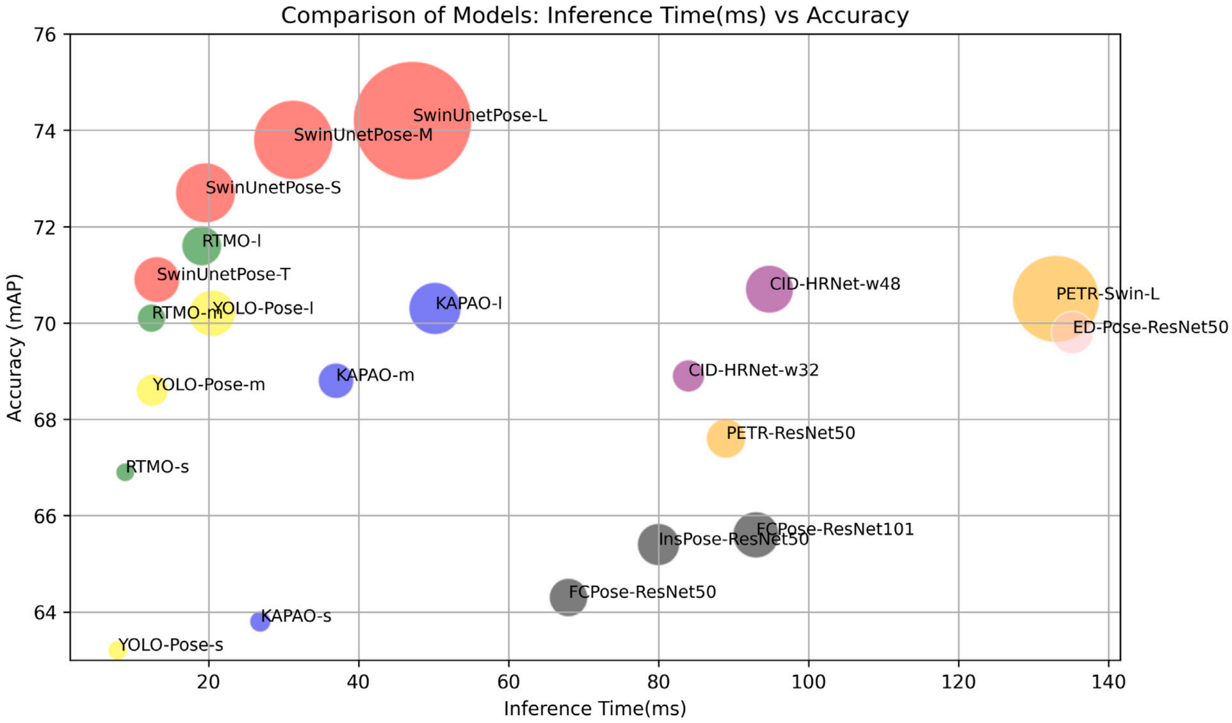


Fig. 6. Comparison of models: Inference time (ms) vs. Accuracy.

Table 7
The performance of SwinUNetPose with segment head on the MS COCO val set.

Method	SwinUNetPose-T		SwinUNetPose-S		SwinUNetPose-B		SwinUNetPose-L	
Segment head		✓		✓		✓		✓
AP	69.2	70.9	70.6	72.7	73.2	73.8	73.7	74.2
AR	71.7	75.1	78.4	78.7	79.5	80.7	80.8	81.1

Table 8
Performance comparison of state-of-the-art one-stage methods on the MS COCO val dataset. inference times in the table were measured on a single NVIDIA Tesla V100 GPU. Non-italicized entries are cited from the PETR [56] paper, while italicized timings are sourced from the RTMO [57] paper. Bold-formatted values represent measurements obtained with TensorRT under FP16 precision acceleration. Underlined entries indicate inference times directly measured using PyTorch due to ONNX export compatibility issues in the corresponding models.

Method	Backbone	Params	Time (ms)	*AP*	AP_{50}	AP_{75}	AP_M	AP_L	*AR*
DirectPose [58]	ResNet-50	–	74	62.2	86.4	68.2	56.7	69.8	–
DirectPose [58]	ResNet-101	–	–	63.3	86.7	69.4	57.8	71.2	–
InsPose [59]	ResNet-50	50.2M	80	65.4	88.9	71.7	60.2	72.7	–
InsPose [59]	ResNet-101	–	100	66.3	89.2	73.0	61.2	73.9	–
FCPose [60]	ResNet-50	41.7M	68	64.3	87.3	71.0	61.6	70.5	–
FCPose [60]	ResNet-101	60.5M	93	65.6	87.9	72.6	62.1	72.3	–
PETR [56]	ResNet-50	43.7M	89	67.6	89.8	75.3	61.6	76.0	–
PETR [56]	Swin-L	213.8M	133	70.5	91.5	78.7	65.2	78.0	–
CID [61]	HRNet-w32	29.4M	84.0	68.9	89.9	76.9	63.2	77.7	74.6
CID [61]	HRNet-w48	65.4M	94.8	70.7	90.4	77.9	66.3	77.8	76.4
ED-Pose [62]	ResNet-50	50.6M	135.2	69.8	90.2	77.2	64.3	77.4	–
ED-Pose [62]	Swin-L	218.0M	265.6	72.7	92.3	80.9	67.6	80.0	–
YOLO-Pose-s [63]	CSPDarknet	10.8M	*7.9*	63.2	87.8	69.5	57.6	72.6	67.6
YOLO-Pose-m [63]	CSPDarknet	29.3M	*12.5*	68.6	90.7	75.8	63.4	77.1	72.8
YOLO-Pose-l [63]	CSPDarknet	61.3M	*20.5*	70.2	91.1	77.8	65.3	78.2	74.3
KAPAO-s [64]	CSPNet	12.6M	*26.9*	63.8	88.4	70.4	58.6	71.7	71.2
KAPAO-m [64]	CSPNet	35.8M	*37.0*	68.8	90.5	76.5	64.3	76.0	76.3
KAPAO-l [64]	CSPNet	77.0M	*50.2*	70.3	91.2	77.8	66.3	76.8	77.7
RTMO-s [57]	CSPDarknet	9.9M	*8.9*	66.9	88.8	73.6	61.1	75.7	70.9
RTMO-m [57]	CSPDarknet	22.6M	*12.4*	70.1	90.6	77.1	65.1	78.1	74.2
RTMO-l [57]	CSPDarknet	44.8M	*19.1*	71.6	91.1	79.0	66.8	79.1	75.6
SwinUNetPose-T	Swin-T	59.2M	**13.1**	70.9	90.5	77.8	66.5	79.2	78.3
SwinUNetPose-S	Swin-S	101.4M	**19.6**	72.7	91.9	81.0	67.7	80.4	79.7
SwinUNetPose-B	Swin-B	177.7M	**31.3**	73.8	92.3	81.2	68.1	81.3	80.7
SwinUNetPose-L	Swin-L	396.1M	**47.2**	74.2	92.8	81.5	68.3	81.5	81.1

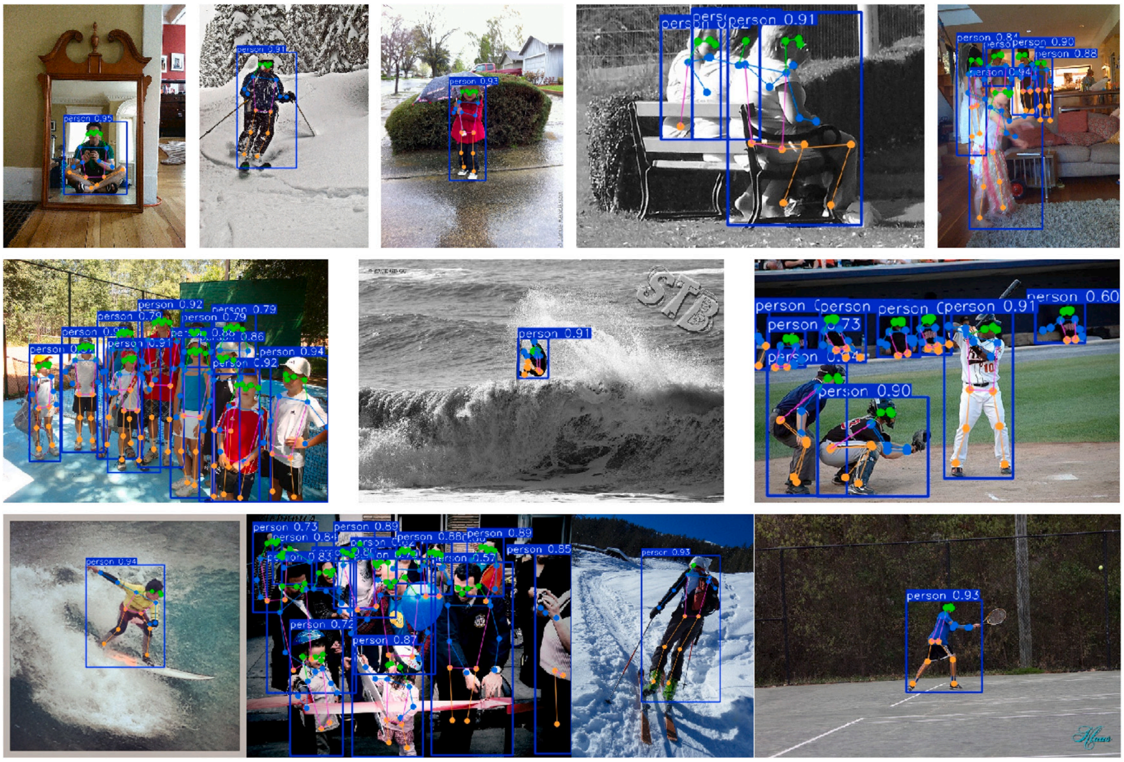


Fig. 7. Visualization results of our approach.

in the AR metric, indicating that the segmentation head plays a critical role in enhancing the recall rate for pose estimation tasks.

These results suggest that the segmentation head helps more accurately localize the human body, thereby improving the accuracy and robustness of pose estimation. Particularly in larger models such as SwinUNetPose-B and SwinUNetPose-L, the performance improvement is more significant with the segmentation module. This further validates the positive impact of simultaneously performing human segmentation on pose estimation performance and highlights the importance of the segmentation head in enhancing model accuracy and robustness. In the SwinUNetPose model, the segmentation module is tightly integrated with the keypoint prediction head, working together by sharing feature information to achieve mutual benefits. The segmentation module first separates the human region from the background, removing background interference to provide cleaner input for the keypoint prediction head. This process helps the model more accurately identify and localize human regions in complex backgrounds, providing strong support for the pose estimation task. Conversely, the pose estimation task guides the segmentation network by predicting human keypoints, focusing on accurate keypoint localization to further refine the segmentation results. This mutual supervision mechanism is supported by a joint loss function, optimizing both segmentation and pose estimation tasks. Specifically, the segmentation loss and keypoint prediction loss are combined into a joint loss function, allowing the segmentation task to not only provide regularization but also improve segmentation performance through the pose estimation task. In particular, the pose estimation task improves segmentation accuracy by ensuring keypoint consistency. Through this integrated strategy, SwinUNetPose effectively optimizes both tasks, leading to a significant enhancement in overall performance.

4.6. Comparison with different methods

The proposed SwinUNetPose is a bottom-up 2D HPE algorithm. To comprehensively evaluate its performance, we conducted a systematic comparison with several existing bottom-up 2D pose estimation algorithms. Specifically, we selected DirectPose [58], FCPose [60], InsPose [59], PETR [56], ED-Pose [62], CID [61], KAPAO [64], YOLO-Pose [63], and RTMO [57] as comparison models. We measured the AP and inference latency of each model on the COCO val2017 dataset to comprehensively assess the detection performance and computational efficiency of different algorithms. The experimental results are shown in Table 8:

Based on the experimental results in Table 8, it can be seen that the SwinUNetPose series models strike a good balance between accuracy and inference speed, demonstrating excellent performance. Specifically, SwinUNetPose-S and SwinUNetPose-B perform outstandingly in metrics such as AP, AP_{50}, AP_{75}, and AR, showing a clear advantage over many existing one-stage models (e.g., YOLO-Pose and KAPAO series). For example, SwinUNetPose-S outperforms KAPAO-s (63.8) and YOLO-Pose-s (63.2) in AP (72.7), while maintaining a low inference latency (19.6 ms), reflecting its excellent performance in both accuracy and efficiency.

When compared to other Transformer-based models, SwinUNetPose-L also exhibits significant advantages, achieving an AP of 74.2, which is notably higher than ED-Pose (72.7) and RTMO-l (71.6), while maintaining an inference speed of 47.2 ms, demonstrating both faster speed and lower computational overhead. Compared to advanced models like HRNet and RTMO, SwinUNetPose excels in AP and AR, particularly showcasing good scalability across different model sizes, maintaining high inference efficiency even with large models. In summary, SwinUNetPose excels in accuracy, robustness, and computational efficiency, particularly striking a balance across models of different sizes. Compared to existing one-stage pose estimation algorithms, SwinUNetPose surpasses many CNN- and Transformer-based methods in AP and AR, while maintaining low inference latency. Fig. 6 further illustrates the relationship between the number of parameters, inference time, and accuracy of SwinUNetPose, highlighting the model's competitive advantages across multiple dimensions.

In summary, SwinUNetPose combines the powerful feature extraction capabilities of the Swin Transformer with the spatial recovery advantages of U-Net, significantly improving the accuracy, robustness, and computational efficiency of pose estimation. The model's excellent scalability ensures it maintains high performance across different application scales. These experiments validate SwinUNetPose's potential as an efficient and competitive pose estimation algorithm, particularly in practical applications such as athlete motion quality detection.

4.7. Visualization results

To verify the effectiveness of the proposed SwinUNetPose algorithm in real-world applications, we conducted a detailed visualization analysis of the detection results. As shown in Fig. 7, SwinUNetPose demonstrates significant advantages in handling challenging situations. In complex scenarios, including occlusion, blurriness, scale variation, appearance differences, unusual body postures, complex backgrounds, and crowded scenes, the model consistently provides reliable pose estimation results. The visualization results clearly show that the model performs reliably even in scenarios involving multiple athletes' poses, different background settings, and small targets. For example, in multi-person scenes, despite occlusions and overlaps between individuals, SwinUNetPose accurately identifies the keypoints of each person and performs correct pose estimation. Additionally, the model demonstrates strong robustness in capturing fast movements and complex actions, such as hitting a ball or skiing in sports, providing fast and precise pose predictions.

5. Conclusion and discussion

In this paper, we propose SwinUNetPose, an innovative human pose estimation (HPE) model that combines the Swin Transformer with the U-Net architecture. This hybrid architecture leverages the multi-scale capabilities of the Transformer module, significantly enhancing the accuracy and robustness of pose estimation, especially in complex environments. By processing input images at different resolutions and capturing key information across multiple scales, SwinUNetPose significantly improves pose estimation performance. Additionally, we introduce a segmentation module, which simultaneously performs human segmentation alongside pose estimation, further enhancing the model's robustness. This integration not only improves recall rate (AR) but also makes the model more adaptable to challenging scenarios. The effectiveness of these innovations is validated through a series of ablation experiments. The experimental results show that increasing input resolution significantly improves both accuracy (AP) and recall rate (AR). The inclusion of the segmentation module also optimizes the model's performance in complex scenes, leading to more stable and accurate predictions. Notably, the segmentation head plays a key role in improving recall rate, which directly enhances the accuracy of pose estimation. Compared to existing bottom-up 2D HPE methods, SwinUNetPose outperforms them in terms of accuracy, recall rate, and inference efficiency, making it a powerful solution for athlete motion quality detection.

However, while the model excels in pose estimation tasks, there are some limitations. As the model scale increases, inference time also increases, especially for larger models (e.g., SwinUNetPose-L). This increased latency may limit its application in real-time or near-real-time tasks. Future research could explore model compression and quantization techniques to optimize inference efficiency and improve response speed without sacrificing accuracy. Moreover, while SwinUNetPose performs excellently in athlete pose estimation, its robustness may be affected when dealing with multiple athletes. Further work could focus on optimizing the segmentation module or incorporating context modeling techniques to improve the model's adaptability in these challenging scenarios. In the context of Internet of Things (IoT) applications, it is important to consider the trade-off between inference speed and accuracy. While SwinUNetPose shows promising results, the increased computational cost of larger models may limit its deployment in resource-constrained IoT devices. Therefore, optimizing the model for faster inference is crucial for real-time applications, particularly in sports and health monitoring systems where low-latency performance is essential.

Additionally, in real-world scenarios, especially when processing multiple athletes in real-time, the model's scalability and energy efficiency become critical factors. The impact of SwinUNetPose on energy consumption and latency needs further exploration to assess its suitability for deployment on edge devices. Deploying the model on devices with limited computational resources presents challenges such as ensuring efficient performance, minimizing power consumption, and maintaining low inference latency. Future work will address these practical considerations, ensuring that SwinUNetPose remains adaptable and efficient in real-world applications.

In conclusion, SwinUNetPose provides an efficient and accurate solution for human pose estimation tasks, especially in the context of athlete motion quality detection. With further optimization in inference speed and robustness, SwinUNetPose is expected to become a widely used tool in sports, health monitoring, and other IoT-based applications. Future research will focus on improving model efficiency, enhancing real-time performance, and ensuring its viability for deployment in complex, resource-constrained environments.

CRediT authorship contribution statement

GuanLan Cai: Writing – original draft, Funding acquisition, Data curation, Conceptualization. **Guodong Zhang:** Writing – review & editing, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data set sources have been identified in the text. The code can be obtained by asking the corresponding author.

References

[1] A. Naouri, N.A. Nouri, A. Khelloufi, A.B. Sada, H. Ning, S. Dhelim, Efficient fog node placement using nature-inspired metaheuristic for IoT applications, Clust. Comput. 27 (6) (2024) 8225–8241.

[2] A. Khelloufi, H. Ning, A. Naouri, A.B. Sada, A. Qammar, A. Khalil, L. Mao, S. Dhelim, A multimodal latent-features-based service recommendation system for the social Internet of Things, IEEE Trans. Comput. Soc. Syst. (2024).

[3] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, M. Wang, Swin-unet: Unet-like pure transformer for medical image segmentation, in: European Conference on Computer Vision, Springer, 2022, pp. 205–218.

[4] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022.

[5] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, M. Shah, Deep learning-based human pose estimation: A survey, ACM Comput. Surv. 56 (1) (2023) 1–37.

[6] X. Xing, B. Wang, X. Ning, G. Wang, P. Tiwari, Short-term OD flow prediction for urban rail transit control: A multi-graph spatiotemporal fusion approach, Inf. Fusion 118 (2025) 102950, http://dx.doi.org/10.1016/j.inffus.2025.102950, URL: https://www.sciencedirect.com/science/article/pii/S1566253525000235.

[7] P. Zhang, X. Yu, X. Bai, J. Zheng, X. Ning, E.R. Hancock, Fully decoupled end-to-end person search: An approach without conflicting objectives, Int. J. Comput. Vis. (2025) 1–22.

[8] Y. Xu, J. Zhang, Q. Zhang, D. Tao, Vitpose: Simple vision transformer baselines for human pose estimation, Adv. Neural Inf. Process. Syst. 35 (2022) 38571–38584.

[9] E. Ning, C. Wang, H. Zhang, X. Ning, P. Tiwari, Occluded person re-identification with deep learning: A survey and perspectives, Expert Syst. Appl. 239 (2024) 122419, http://dx.doi.org/10.1016/j.eswa.2023.122419, URL: https://www.sciencedirect.com/science/article/pii/S0957417423029214.

[10] 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 (6) (2016) 1137–1149.

[11] A.S. Micilotta, E.-J. Ong, R. Bowden, Real-time upper body detection and 3D pose estimation in monoscopic images, in: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part III 9, Springer, 2006, pp. 139–150.

[12] C. Wang, X. Ning, W. Li, X. Bai, X. Gao, 3D person re-identification based on global semantic guidance and local feature aggregation, IEEE Trans. Circuits Syst. Video Technol. (2023) http://dx.doi.org/10.1109/TCSVT.2023.3328712, 1–1.

[13] T. Zhanglu, Y. Hui, W. Fuhao, Similarity search algorithm based on DWT, Stat. Inf. Forum 38 (1) (2023) 3–15, URL: https://link.cnki.net/urlid/61.1421.C.20221021.1721.010.

[14] D. Wang, Stacked Dense-Hourglass Networks for Human Pose Estimation (Ph.D. thesis), University of Illinois at Urbana-Champaign, 2018.

[15] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, C. Bregler, Efficient object localization using convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 648–656.

[16] X. Chen, A.L. Yuille, Articulated pose estimation by a graphical model with image dependent pairwise relations, Adv. Neural Inf. Process. Syst. 27 (2014).

[17] A. Toshev, C. Szegedy, Deeppose: Human pose estimation via deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1653–1660.

[18] X. Sun, J. Shang, S. Liang, Y. Wei, Compositional human pose regression, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2602–2611.

[19] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[20] K. Li, S. Wang, X. Zhang, Y. Xu, W. Xu, Z. Tu, Pose recognition with cascade transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1944–1953.

[21] S. Ruder, An overview of multi-task learning in deep neural networks, 2017, arXiv preprint arXiv:1706.05098.

[22] S. Li, Z.-Q. Liu, A.B. Chan, Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 482–489.

[23] X. Fan, K. Zheng, Y. Lin, S. Wang, Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1347–1355.

[24] J.J. Tompson, A. Jain, Y. LeCun, C. Bregler, Joint training of a convolutional network and a graphical model for human pose estimation, Adv. Neural Inf. Process. Syst. 27 (2014).

[25] S.-E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, Convolutional pose machines, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4724–4732.

[26] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation european conference on computer vision (ECCV), 2016.

[27] X. Chu, W. Yang, W. Ouyang, C. Ma, A.L. Yuille, X. Wang, Multi-context attention for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1831–1840.

[28] W. Yang, S. Li, W. Ouyang, H. Li, X. Wang, Learning feature pyramids for human pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1281–1290.

[29] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5693–5703.

[30] C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang, J. Wang, Lite-hrnet: A lightweight high-resolution network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10440–10450.

[31] Y. Yuan, R. Fu, L. Huang, W. Lin, C. Zhang, X. Chen, J. Wang, Hrformer: High-resolution vision transformer for dense predict, Adv. Neural Inf. Process. Syst. 34 (2021) 7281–7293.

[32] B. Xiao, H. Wu, Y. Wei, Simple baselines for human pose estimation and tracking, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 466–481.

[33] H.-S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, Y. Xiu, Y.-L. Li, C. Lu, Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time, IEEE Trans. Pattern Anal. Mach. Intell. 45 (6) (2022) 7157–7173.

[34] J. Wang, X. Long, Y. Gao, E. Ding, S. Wen, Graph-pcnn: Two stage human pose estimation with graph pose refinement, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16, Springer, 2020, pp. 492–508.

[35] Y. Cai, Z. Wang, Z. Luo, B. Yin, A. Du, H. Wang, X. Zhang, X. Zhou, E. Zhou, J. Sun, Learning delicate local representations for multi-person pose estimation, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, Springer, 2020, pp. 455–472.

[36] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, B. Schiele, Deepercut: A deeper, stronger, and faster multi-person pose estimation model, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14, Springer, 2016, pp. 34–50.

[37] S. Qiao, Y. Wang, J. Li, Real-time human gesture grading based on OpenPose, in: 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI, IEEE, 2017, pp. 1–6.

[38] D. Osokin, Real-time 2d multi-person pose estimation on cpu: Lightweight openpose, 2018, arXiv preprint arXiv:1811.12004.

[39] B. Cheng, B. Xiao, J. Wang, H. Shi, T.S. Huang, L. Zhang, Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5386–5395.

[40] S. Yang, Z. Quan, M. Nie, W. Yang, Transpose: Keypoint localization via transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 11802–11812.

[41] Y. Li, S. Zhang, Z. Wang, S. Yang, W. Yang, S.-T. Xia, E. Zhou, Tokenpose: Learning keypoint tokens for human pose estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 11313–11322.

[42] Y. Xu, J. Zhang, Q. Zhang, D. Tao, ViTPose++: Vision transformer foundation model for generic body pose estimation, 2022, arXiv preprint arXiv:2212.04246.

[43] F. Xia, P. Wang, X. Chen, A.L. Yuille, Joint multi-person pose estimation and semantic part segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6769–6778.

[44] H. Cai, Y. Gao, et al., Optimization of human pose detection based on mask RCNN, in: 2021 2nd International Symposium on Computer Engineering and Intelligent Communications, ISCEIC, IEEE, 2021, pp. 273–277.

[45] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G. Ding, Yolov10: Real-time end-to-end object detection, 2024, arXiv preprint arXiv:2405.14458.

[46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.

[47] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2D human pose estimation: New benchmark and state of the art analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3686–3693.

[48] J. Wu, H. Zheng, B. Zhao, Y. Li, B. Yan, R. Liang, W. Wang, S. Zhou, G. Lin, Y. Fu, et al., Ai challenger: A large-scale dataset for going deeper in image understanding, 2017, arXiv preprint arXiv:1711.06475.

[49] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu, C. Qian, W. Ouyang, P. Luo, Whole-body human pose estimation in the wild, in: European Conference on Computer Vision, Springer, 2020, pp. 196–214.

[50] S.-H. Zhang, R. Li, X. Dong, P. Rosin, Z. Cai, X. Han, D. Yang, H. Huang, S.-M. Hu, Pose2seg: Detection free human instance segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 889–898.

[51] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, K.M. Lee, Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16, Springer, 2020, pp. 548–564.

[52] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16000–16009.

[53] S.J. Reddi, S. Kale, S. Kumar, On the convergence of adam and beyond, 2019, arXiv preprint arXiv:1904.09237.

[54] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: International Conference on Machine Learning, PMLR, 2021, pp. 10347–10357.

[55] X. Chen, S. Xie, K. He, An empirical study of training self-supervised vision transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 9640–9649.

[56] D. Shi, X. Wei, L. Li, Y. Ren, W. Tan, End-to-end multi-person pose estimation with transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11069–11078.

[57] P. Lu, T. Jiang, Y. Li, X. Li, K. Chen, W. Yang, RTMO: towards high-performance one-stage real-time multi-person pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 1491–1500.

[58] Z. Tian, H. Chen, C. Shen, Directpose: Direct end-to-end multi-person pose estimation, 2019, arXiv preprint arXiv:1911.07451.

[59] D. Shi, X. Wei, X. Yu, W. Tan, Y. Ren, S. Pu, Inspose: instance-aware networks for single-stage multi-person pose estimation, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 3079–3087.

[60] W. Mao, Z. Tian, X. Wang, C. Shen, Fcpose: Fully convolutional multi-person pose estimation with dynamic instance-aware convolutions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9034–9043.

[61] D. Wang, S. Zhang, Contextual instance decoupling for robust multi-person pose estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11060–11068.

[62] J. Yang, A. Zeng, S. Liu, F. Li, R. Zhang, L. Zhang, Explicit box detection unifies end-to-end multi-person pose estimation, 2023, arXiv preprint arXiv:2302.01593.

[63] D. Maji, S. Nagori, M. Mathew, D. Poddar, Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2637–2646.

[64] W. McNally, K. Vats, A. Wong, J. McPhee, Rethinking keypoint representations: Modeling keypoints and poses as objects for multi-person human pose estimation, in: European Conference on Computer Vision, Springer, 2022, pp. 37–54.