electronics

MDPI

Article

KeypointNet: An Efficient Deep Learning Model with Multi-View Recognition Capability for Sitting Posture Recognition

Zheng Cao †, Xuan Wu †, Chunguo Wu, Shuyang Jiao, Yubin Xiao, Yu Zhang * and You Zhou *

Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun 130012, China; caozheng22@mails.jlu.edu.cn (Z.C.); wuuu22@mails.jlu.edu.cn (X.W.); wucg@jlu.edu.cn (C.W.); jiaosy21@mails.jlu.edu.cn (S.J.); xiaoyb21@mails.jlu.edu.cn (Y.X.)

* Correspondence: zy26@jlu.edu.cn (Y.Z.); zyou@jlu.edu.cn (Y.Z.)

† These authors contributed equally to this work.

Abstract: Numerous studies leverage pose estimation to extract human keypoint data and then classify sitting postures. However, employing neural networks for direct keypoint classification often yields suboptimal results. Alternatively, modeling keypoints into other data representations before classification introduces redundant information and substantially increases inference time. In addition, most existing methods perform well only under a single fixed viewpoint, limiting their applicability in complex real-world scenarios involving unseen viewpoints. To better address the first limitation, we propose KeypointNet, which employs a decoupled feature extraction strategy consisting of a Keypoint Feature Extraction module and a Multi-Scale Feature Extraction module. In addition, to enhance multi-view recognition capability, we propose the Multi-View Simulation (MVS) algorithm, which augments the viewpoint information by first rotating keypoints and then repositioning the camera. Simultaneously, we propose the multi-view sitting posture (MVSP) dataset, designed to simulate diverse real-world viewpoints. The experimental results demonstrate that KeypointNet outperforms the other state-of-the-art methods on both the proposed MVSP dataset and the other public datasets, while maintaining a lightweight and efficient design. Ablation studies demonstrate the effectiveness of MVS and all KeypointNet modules. Furthermore, additional experiments highlight the superior generalization, small-sample learning capability, and robustness to unseen viewpoints of KeypointNet.

Keywords: deep learning; computer vision; convolutional neural networks; sitting posture recognition

check for updates

Academic Editor: George A. Tsihrintzis

Received: 9 January 2025
Revised: 9 February 2025
Accepted: 11 February 2025
Published: 12 February 2025

Citation: Cao, Z.; Wu, X.; Wu, C.; Jiao, S.; Xiao, Y.; Zhang, Y.; Zhou, Y. KeypointNet: An Efficient Deep Learning Model with Multi-View Recognition Capability for Sitting Posture Recognition. *Electronics* **2025**, *14*, 718. https://doi.org/10.3390/electronics14040718

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, improper sitting posture has led to an increasing prevalence of musculoskeletal disorders. However, these conditions can be prevented through the proactive identification and correction of poor sitting habits [1–4]. A method for recognizing improper sitting posture can help improve people's health to some extent. Furthermore, recognizing incorrect posture can provide valuable references for broader research areas [5–12]. Pose estimation techniques utilized in the study of sitting posture recognition are not only applicable to ergonomic assessments but also find extensive use in various domains such as healthcare, sports, and human–computer interaction.

As a common computer vision application, pose estimation techniques [13–17] are frequently utilized to extract human keypoints. These keypoints are then used for sub-

sequent sitting posture recognition. Among them, how to handle keypoints is the core step [18]. Certain studies model keypoints as other data representations, such as skeleton images [19,20], 3D voxels [21], or spatiotemporal sequences [22], enabling the application of more advanced and powerful models for classification. However, modeling keypoints into other data representations inevitably introduces noise and redundant information, which increases inference time [21] and hinders networks from effectively extracting critical features. To avoid data redundancy in posture recognition, directly utilizing keypoint data is an effective approach. For instance, certain studies used methods such as random forest, support vector machines (SVMs), and multi-layer perceptrons (MLPs) [23–26]. However, these methods are relatively traditional and sufficient feature extraction capabilities [22]. Consequently, there is a need for an efficient model that can directly process keypoint data while maintaining high performance.

In addition to the limitation associated with handling keypoint data effectively, another significant limitation is the insufficient research on multi-view recognition. Real-world scenarios such as classrooms, car cameras, and libraries present diverse angles that can significantly impact the performance of recognition models. The variability in viewpoints can introduce substantial challenges, especially when the training data are limited to a narrow range of viewpoints. However, existing studies on multi-view sitting posture recognition are insufficient, especially when models are faced with unseen viewpoints.

To better address the first limitation, we propose KeypointNet, a model that can efficiently identify sitting posture directly from keypoint data. KeypointNet is mainly composed of a Keypoint Feature Extraction module (KFE) and Multi-Scale Feature Extraction module (MSFE). Specifically, KFE uses 1D convolution with a kernel size of 1 to extract within-point features from keypoint data and transfers the extraction of between-point features to the next stage. This approach decouples keypoint feature extraction into within-point and between-point extraction, enabling more efficient recognition. Subsequently, to obtain richer spatial information when extracting between-point features, the tailored MSFE is composed of a Local Feature Extraction module (LFE) and a Global Feature Extraction module (GFE). Specifically, LFE uses 1D convolutions and MLPs to extract features from our six proposed body-part regions of keypoints, while GFE extracts global features from all of the keypoint data. These two modules, designed to operate across multiple scales, enhance the network's ability to capture the interactions between keypoints. Moreover, to improve the model's multi-view recognition capability, we introduce a data augmentation method, named the Multi-View Simulation algorithm (MVS). Specifically, MVS first rotates the coordinates of the keypoints and then randomly generates a new set of camera pose information, combined with the collinearity equation [27] in photogrammetry, to construct the coordinates of the keypoints from a new viewpoint.

To simulate complex viewpoint scenarios in the real world, we build a multi-view sitting posture dataset MVSP, which exploits the high-performance MHFormer [14] to obtain 3D keypoint data. This dataset contains seven fixed viewpoints for training and one dynamic shooting viewpoint for testing. This data division design ensures the test set includes viewpoints that are not present in the training set, effectively evaluating the model's capability to recognize multi-view data.

To assess the effectiveness of KeypointNet, we conduct extensive experiments across diverse datasets. The experimental results demonstrate that KeypointNet outperforms other state-of-the-art methods on the MVSP, HASP [22], and MSRA [28] datasets, achieving the highest multi-view recognition accuracy and superior speed. Specifically, it achieved an impressive accuracy of 98.46% on the MVSP, 96.52% on HASP, and 99.48% on the MSRA. Ablation studies validate the effectiveness of MVS and each module of KeypointNet, and data sampling experiments validate that our model can learn the most from minimal

data, compared with LMSPNet [20] and ASPR [22]. Finally, we analyze the performance of KeypointNet in dealing with unseen viewpoints in the training set.

The key contributions of this work are as follows:

1. We propose KeypointNet, featuring a decoupled architecture with KFE for extracting within-point features and MSFE for capturing between-point features. This design enables efficient and direct sitting posture recognition from keypoint data.
2. To improve multi-view recognition performance, we propose the MVS algorithm, which augments the viewpoint information of the keypoints. In addition, we propose MVSP, a multi-view sitting posture dataset, specifically designed to investigate the performances associated with multi-view sitting posture recognition.
3. The experimental results demonstrate that KeypointNet achieves 98.46% accuracy on MVSP and leading accuracies on other datasets, while excelling in inference speed. Furthermore, additional experiments demonstrate the advantages of our method in various aspects.

2. Related Work

In this section, we first introduce existing sitting posture recognition methods, especially those based on computer vision. Subsequently, we discuss the PointNet series of models.

2.1. Sitting Posture Recognition Methods

In the field of computer vision-based sitting posture recognition [29–35], many studies have employed pose estimation to extract keypoints from images, thereby mitigating the influence of lighting and shading variations. Utilizing these keypoint data for recognition is a crucial step. Liu et al. [21] employed Gaussian voxel modeling for skeletal keypoints and utilized a 3D-CNN model for sitting posture recognition. Fang et al. [19] and Jiao et al. [20] both drew skeleton images based on keypoints and converted them into CNN classification tasks. Although advanced CNN models can be leveraged for sitting posture recognition, using skeleton images or 3D voxels introduces significant information redundancy and additional noise, complicating the extraction of important features [21]. Li et al. [22] utilized a spatiotemporal graph convolutional network (ST-GCN) to powerfully recognize keypoint data by stacking keypoints in the temporal dimension to form a spatiotemporal sequence. Despite its effectiveness, part of the information is devoted to representing temporal dynamics, and many of the data are repetitive, making it less efficient for recognizing static postures. To reduce data redundancy, directly recognizing keypoint data is a promising method. Chen et al. [36] used a transformer to handle temporal changes in keypoints for action counting. For posture recognition, many studies [23–26] used methods such as random forest, SVM, or MLP to process keypoint data directly. However, these methods lack powerful expression capabilities [22]. Regarding studies on multi-view sitting posture recognition, Xu et al. [37] used three different cameras to capture images from three viewpoints, obtaining richer viewpoint information. However, this method requires extensive equipment setup and does not address the multi-view recognition problem under single-camera constraints. Abdullah et al. [38] recognized the inadequacy of multi-view recognition for a single-camera setup and constructed datasets captured from multiple viewpoints. Nevertheless, these above studies did not explore the impact when training data for certain angles are missing, leaving room for further investigation.

To address the first limitation, we propose KeypointNet, which can directly and efficiently identify keypoint data. To further improve multi-view recognition, we propose the MVS algorithm for data augmentation and introduce the MVSP dataset for experimentation.

2.2. Point Cloud Processing Networks: PointNet Series

Given the significant similarity between 3D point cloud data and 3D keypoint data, we identify the great potential of PointNet Series, a method that efficiently and concisely processes 3D point cloud data. Since point cloud data are unordered and structurally irregular, PointNet [39] uses the Max operation to achieve invariance to the input order:

$$\mathrm{Max}(x_1, x_2, \ldots, x_n) \equiv \mathrm{Max}(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}). \quad (1)$$

Specifically, it first performs dimensionality expansion on the point cloud data to construct redundancy and prevent information loss, and then it uses the *Max* operation to reduce dimensionality, allowing the network to learn the unordered nature of the data. To improve local feature extraction, Qi et al. [40] proposed PointNet++, which introduces a hierarchical feature extraction paradigm to recursively capture local geometric structures. PointNet++ has laid the foundation for subsequent works and has inspired the development of more advanced models [41,42].

While PointNet and its variants have demonstrated remarkable success in handling unordered point cloud data, applying these models to keypoint data presents unique challenges. First, since keypoint data are typically ordered, PointNets' operation to handle point cloud data's unordered nature introduces unnecessary complexity in the network structure. Second, many designs in PointNets are tailored for handling thousands of points, whereas keypoint data usually contain only a few dozen points, limiting the effectiveness of PointNets in processing smaller data. To fully leverage the potential of PointNets for keypoint data, we design KeypointNet, which reduces redundant operations for handling unordered data and optimizes the network structure, achieving excellent results.

3. Method

In this section, we first introduce the Multi-View Simulation algorithm (MVS), which includes two steps: random keypoint rotation and random camera repositioning. Then, we introduce KeypointNet, which consists of a Keypoint Feature Extraction module (KFE) and a Multi-Scale Feature Extraction module (MSFE). Figure 1 provides an overview of the overall process of the proposed method.

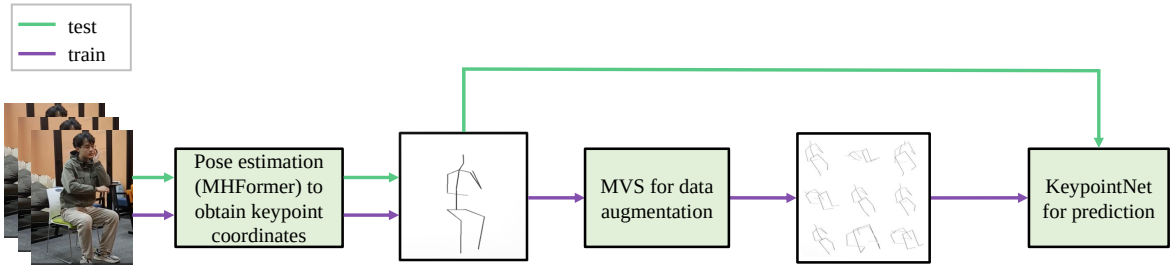


Figure 1. An overview of the proposed MVS and KeypointNet method for sitting posture recognition. Firstly, a video stream is captured by a camera, and images from the stream are extracted and processed using a pose estimation method to infer 3D keypoint data (MHFormer [14] is used in our method). During the training process, MVS is employed to augment the keypoint data, after which KeypointNet is used for recognition. During the testing process, the keypoint data obtained from the pose estimation method are directly recognized using KeypointNet.

3.1. Multi-View Simulation Algorithm

To enhance the multi-view recognition capability in sitting posture recognition, we propose the tailored MVS algorithm. Unlike prior studies [39–41], which typically use rotation matrices to transform the geometric orientation of keypoints without considering the interaction between the camera and the keypoints, we enhance the viewpoint transforma-

tion by incorporating camera factors into the rotation matrix. We decompose the viewpoint transformation into two independent parts, namely, random keypoint rotation and random camera repositioning. Specifically, random keypoint rotation changes the orientation of keypoints without altering the camera state, while random camera repositioning changes the camera pose while keeping the keypoint data unchanged. The illustration of MVS is presented in Figure 2.

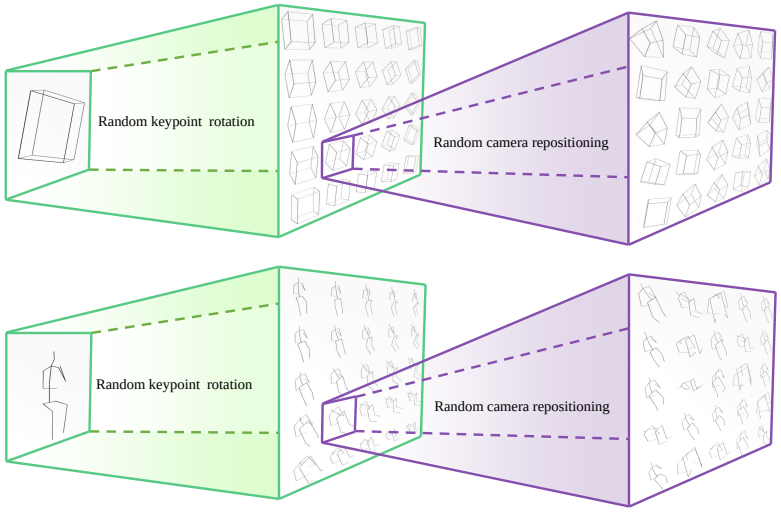


Figure 2. An illustration of MVS's viewpoint transformation process. The upper part uses cubes to visualize the spatial transformation more intuitively. Random keypoint rotation alters the orientation of the data without changing the relative positions between keypoints. In contrast, random camera repositioning introduces more diverse viewpoint transformations and changes the relative positions between keypoints. The lower part demonstrates a set of keypoint data to showcase the actual data augmentation effects.

Random keypoint rotation: The goal of random keypoint rotation is to alter the spatial orientation of keypoints while preserving their internal structure and relative positions. A rotation matrix is directly applied to the keypoints to randomly rotate them by a certain angle. This method rotates the orientation of keypoints. The computation is as follows:

$$P_r = R_X R_Y R_Z P, \tag{2}$$

where P_r denotes the rotated keypoints and $P = \{p^1, p^2, \ldots, p^{17}\}$ denotes the initial keypoint data. Each keypoint p^i has three coordinates x^i, y^i, and z^i. The number and order of the 17 keypoints, as shown in Figure 3, are based on the keypoint configuration used in the Human3.6M dataset [43]. R_X, R_Y, and R_Z denote the rotation matrices for the corresponding axes, respectively. The definitions of these rotation matrices are as follows:

$$\begin{cases} R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_1 & -\sin\theta_1 \\ 0 & \sin\theta_1 & \cos\theta_1 \end{bmatrix} \\ R_Y = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \\ -\sin\theta_2 & 0 & \cos\theta_2 \end{bmatrix} \\ R_Z = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{cases}, \tag{3}$$

where $\theta_1, \theta_2, \theta_3 \in [-\pi/3, \pi/3]$ denote the rotation angles of R_X, R_Y, and R_Z, respectively.

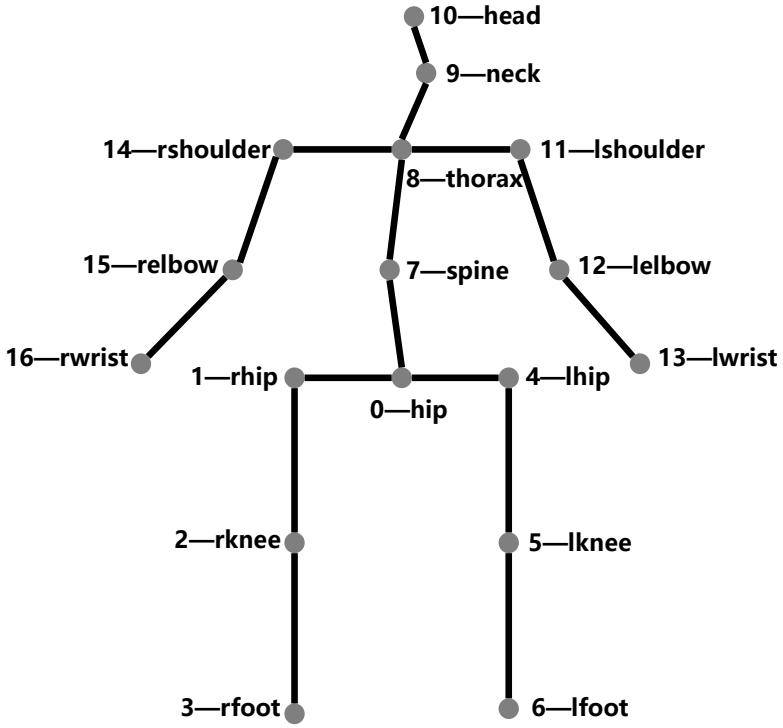


Figure 3. An illustration of the 17 human skeletal keypoints' distribution.

Random camera repositioning: The purpose of random camera repositioning is to alter the pose of the camera while keeping the keypoint data unchanged, thus enriching the viewpoint information of keypoints. We use the collinearity equation to compute the keypoint data from different viewpoints. Specifically, the collinearity equation establishes a relationship between real-world 3D coordinates, image-world 2D coordinates, and the camera pose. Given real-world 3D coordinates and the camera pose, the 2D coordinates in the image world can be calculated. Here, we assume that the initial keypoint data P represent real-world 3D coordinates. We randomly set a camera pose and use the collinearity equation to compute the corresponding image-world 2D coordinates, thus obtaining 2D keypoint data from a new viewpoint. Then, by appropriately constructing the depth information, we can derive the complete 3D keypoint coordinates.

To reasonably satisfy the solving conditions of the collinearity equation, we need to randomly generate a set of camera pose data within a reasonable scale. Specifically, the scale Sca of P is defined as the maximum distance across the three coordinate dimensions and is computed as follows:

$$Sca = \max(X_{max} - X_{min}, Y_{max} - Y_{min}, Z_{max} - Z_{min}). \tag{4}$$

The camera pose parameters are then set as:

$$\begin{cases} Xs \in [Xmean - 2 \cdot Sca, Xmean + 2 \cdot Sca] \\ Ys \in [Ymean - 2 \cdot Sca, Ymean + 2 \cdot Sca] \\ Zs \in [Scale, 5 \cdot Sca] \\ f = 2 \cdot Sca \\ \varphi, \omega, \kappa \in \left[-\frac{\pi}{16}, \frac{\pi}{16}\right] \end{cases}, \tag{5}$$

where Xs, Ys, and Zs denote the coordinates of the camera, f denotes focal length, and φ, ω, and κ denote the angles between the camera orientation and the coordinate axis. Once the camera pose parameters are established, the next step is to compute the 3D coordinates of the keypoints from a new viewpoint using the collinearity equation. Each time the MVS algorithm is used for random camera repositioning, a new set of parameters Xs, Ys, Zs, φ, ω, and κ must be randomly sampled within the data ranges specified in Equation (5). Let $P_a = \{p_a^1, p_a^2, \ldots, p_a^{17}\}$ denote 17 keypoints in the new viewpoint, where each keypoint $p_a^i = \{x_a^i, y_a^i, z_a^i\}$ contains the 3D coordinates. The coordinates x_a^i, y_a^i, z_a^i from a new viewpoint are computed as follows:

$$\begin{cases} x_a^i = -f\frac{a1(x^i-Xs)+b1(y^i-Ys)+c1(z^i-Zs)}{a3(x^i-Xs)+b3(y^i-Ys)+c3(z^i-Zs)} \\ y_a^i = -f\frac{a2(x^i-Xs)+b2(y^i-Ys)+c2(z^i-Zs)}{a3(x^i-Xs)+b3(y^i-Ys)+c3(z^i-Zs)} \\ z_a^i = a3(x^i-Xs)+b3(y^i-Ys)+c3(z^i-Zs) \end{cases}, \tag{6}$$

$$\begin{bmatrix} a1 & a2 & a3 \\ b1 & b2 & b3 \\ c1 & c2 & c3 \end{bmatrix} = \begin{bmatrix} \cos\varphi\cos\kappa - \sin\varphi\sin\omega\sin\kappa & -\cos\varphi\sin\kappa - \sin\varphi\sin\omega\cos\kappa & -\sin\varphi\cos\omega \\ \cos\omega\sin\kappa & \cos\omega\cos\kappa & -\sin\omega \\ \sin\varphi\cos\kappa + \cos\varphi\sin\omega\sin\kappa & -\sin\varphi\sin\kappa + \cos\varphi\sin\omega\cos\kappa & \cos\varphi\cos\omega \end{bmatrix}. \tag{7}$$

In this equation, x_a^i and y_a^i can be directly computed using the collinearity equation. Since the collinearity equation does not provide depth information (z_a^i), we use the original keypoint coordinates x^i, y^i, z^i and perform a coordinate transformation and rotation to obtain z_a^i. Thus, after applying random rotation to the initial keypoints P, we obtain the rotated keypoints P_r. By applying random camera repositioning to P_r, we obtain the new keypoint set P_a, representing the keypoints from a new viewpoint.

Normalization of keypoint data: Finally, to unify the keypoint data space, we first compute the centroid P_{cent} of the keypoints P_a. The centroid is computed as the average position of all keypoints:

$$P_{cent} = \frac{1}{17}\sum_{i=1}^{17} P_a^i. \tag{8}$$

Then, the P_a is normalized as follows:

$$P_{norm} = \frac{P_a - P_{cent}}{Sca}. \tag{9}$$

3.2. KeypointNet

Many prior studies [19–21] used keypoints to model skeleton images or 3D voxels, which results in significant information redundancy. Unlike these prior studies, our aim is to design a model that can efficiently utilize keypoint data while avoiding modeling keypoints into other data representations. Therefore, we propose KeypointNet, which unlocks PointNets' potential for keypoint data. The overall structure of KeypointNet is shown in Figure 4. The initial keypoint data are first subjected to unified extraction of within-point features through KFE, and then MSFE is responsible for further extraction of between-point features. Among them, MSFE conducts multi-scale spatial feature extraction via two modules: the local feature extraction module (LFE) and global feature extraction module (GFE).

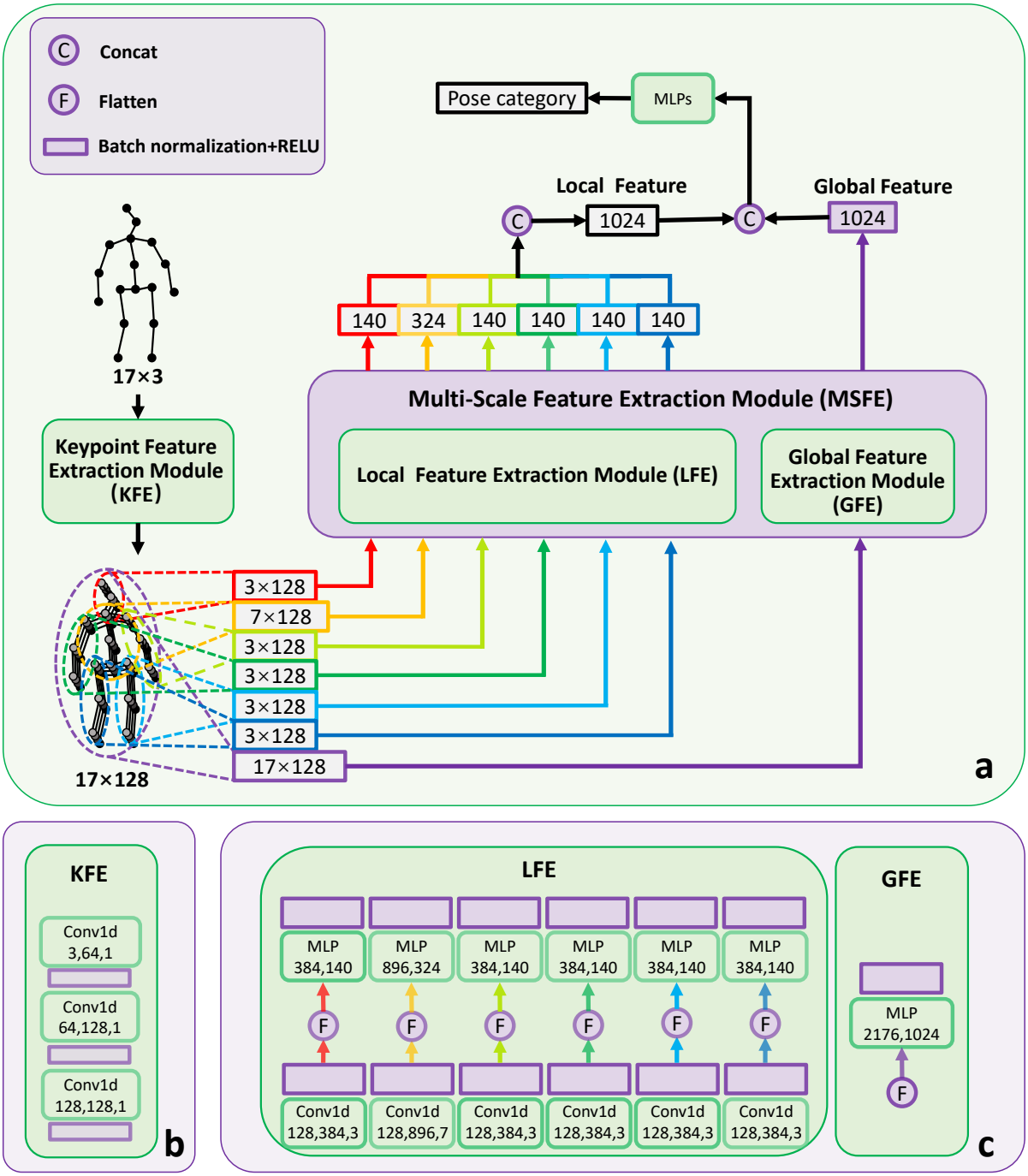


Figure 4. Framework of proposed KeypointNet. (**a**) Overview of the entire KeypointNet architecture. (**b**) Detailed structure of the KFE module within KeypointNet. (**c**) Detailed structure of the MSFE module within KeypointNet.

3.2.1. Keypoint Feature Extraction Module

In PointNet++ [40], the point cloud data directly undergo Multi-Scale Feature Extraction. This approach works well for large point cloud data, where each local region has its own learning pattern. However, given that keypoints are relatively few in number, to efficiently process keypoint features, we propose KFE, which exploits three 1D convolution layers for unified preliminary feature extraction, following the design of PointNet [39] for extracting point cloud features.

The 1D convolution with a kernel size of 1 means that the convolution operation's receptive field is limited to a single point. This allows KFE to extract features for each point

independently, preventing interference between different points. This design effectively decouples feature extraction into within-point and between-point stages. Specifically, KFE extracts within-point features with three layers of 1D convolutions, increasing the feature dimension from 3 to 128. The output feature *KFeature* is computed as follows:

$$KFeature = \text{Relu}(\text{Bn}_i(\text{Conv1d}_i(P_{norm}))), i \in \{1, 2, 3\}, \tag{10}$$

where Conv1d_i represents a 1D convolution operation, which is applied three times corresponding to i. These 1D convolutions are designed to extract features from the normalized keypoint data P_{norm}, where $P_{norm} \in \mathbb{R}^{3\times17}$ denotes the input information for 17 keypoints, and the units and normalization method of P_{norm} are detailed in Equation (9). Unlike PointNet++, which handles the unordered nature of point cloud data by upsampling the dimensions and using a max pooling operation, we eliminate the max pooling operation because keypoint data are ordered. Therefore, we omit operations related to handling disorder.

3.2.2. Multi-Scale Feature Extraction Module

After extracting initial features for each keypoint, our proposed MSFE extracts complex relationships between keypoints and captures features at different scales.

Local feature extraction module: PointNet++ uses a multi-stage hierarchical feature extraction approach to optimize local feature extraction, which uses iterative farthest point sampling [40] to select evenly distributed points from thousands of point cloud data points as representatives and divides the entire point cloud into corresponding groups based on the representative points to construct different local point clouds. Subsequently, hierarchical convolution operations are performed to extract local features. This approach enables the network to capture fine-grained local features from point cloud data, but it is overly complex when handling the relatively small number of human keypoints. Thus, we simplify this design and use spatial relationships between body parts to group nearby keypoints into local regions, as shown in Table 1.

In the proposed LFE, we first use six 1D convolutions to continue extracting specific within-point features of our six proposed local regions, enabling a more region-specific feature extraction process. Then, we use six parallel MLP blocks to aggregate the between-point features of our six proposed local regions, respectively. The structural design of LFE has been proven to be effective in Section 4.3. Ultimately, the extracted local feature, denoted as *LocalFeature*, is composed of six smaller features corresponding to our different proposed body parts: $HeadF, BodyF, LArmF, RArmF, LLegF, RLegF$. For example, the head feature $HeadF$ is computed as follows:

$$HeadF = \text{Bn}(\text{Relu}(\text{Conv1d}(\text{Concat}(KFeature_8, KFeature_9, KFeature_{10})))), \tag{11}$$

$$HeadF = \text{Bn}(\text{Relu}(\text{MLP}(\text{Flatten}(HeadF)))), \tag{12}$$

where $KFeature_8, KFeature_9$, and $KFeature_{10} \in \mathbb{R}^{128}$ denote the features for the 8th, 9th, and 10th keypoints corresponding to the thorax, neck, and head. The corresponding local features for all keypoints are shown in Table 1. The LFE processes and outputs the local features, which are subsequently fused with the global features.

Table 1. Correspondence between six local features and keypoints.

Local Features	Corresponding Keypoints
$HeadF$	8, 9, 10
$BodyF$	0, 1, 4, 7, 8, 11, 14
$LArmF$	11, 12, 13

Table 1. *Cont.*

Local Features	Corresponding Keypoints
RArmF	14, 15, 16
LLegF	4, 5, 6
RLegF	1, 2, 3

Global feature extraction module: In addition to local feature extraction, we introduce GFE to capture the global features of the keypoints from a different scale. Specifically, we use an MLP to capture the relationships between keypoints, which is computed as follows:

$$GlobalFeature = \text{Relu}(\text{Bn}(\text{MLP}(\text{Flatten}(KFeature)))), \tag{13}$$

where $GlobalFeature \in \mathbb{R}^{1024}$ denotes the global feature vector extracted by GFE. The flattened *KFeature* is passed through the MLP for extracting.

Finally, after obtaining *GlobalFeature* through GFE and *LocalFeature* through MSFE, we concat these features using MLP to enable autonomous extracting of different feature representations. The resulting pose classification output *Pose* is computed as follows:

$$Pose = \text{Bn}(\text{Relu}(\text{MLP}(\text{Concat}(LocalFeature, GlobalFeature)))), \tag{14}$$

$$Pose = \text{Relu}(\text{MLP}(Pose)). \tag{15}$$

4. Experimental Results

In this section, we first introduce the datasets used to assess the effectiveness of KeypointNet and MVS, along with the data processing methods and experimental setup. Then, we compare KeypointNet with other advanced methods across multiple datasets. Following this, we conduct ablation studies to demonstrate the effectiveness of KFE, MSFE, and MVS. We also perform data sampling experiments to investigate the performance of various models with limited samples. Finally, we conduct experiments on viewpoint limitedness to provide a specific analysis of the performance in multi-view sitting posture recognition for KeypointNet.

4.1. Data and Experimental Setup

Multi-view sitting posture dataset (MVSP): To explore the model's multi-view recognition capability, we construct the multi-view sitting posture (MVSP) dataset, which includes 50,176 keypoints (extracted from images using MHFormer) from seven fixed viewpoints and one dynamic shooting viewpoint. For the subsequent experiments, we use the keypoints captured from the fixed viewpoints as the training set, while the keypoints from the dynamic shooting viewpoint serve as the test set. This partitioning method can better evaluate the performance of the model when facing diverse viewpoints, verifying the multi-view recognition capability. As shown in Figure 5, the fixed viewpoints include front, front-left, front-right, left side, right side, top-left, and top-right angles, while the dynamic shooting viewpoint encompasses a wider variety of potential viewpoints. The seven actions in this dataset are as follows: upright sitting, leaning down, leaning backward, leaning left, leaning right, chin on hand, and crossing legs. Throughout all data collection processes, participants are instructed to vary their posture as much as possible within the same posture category. This increases the complexity of the dataset, presenting a greater challenge for the generalization capability of the sitting pose recognition model.

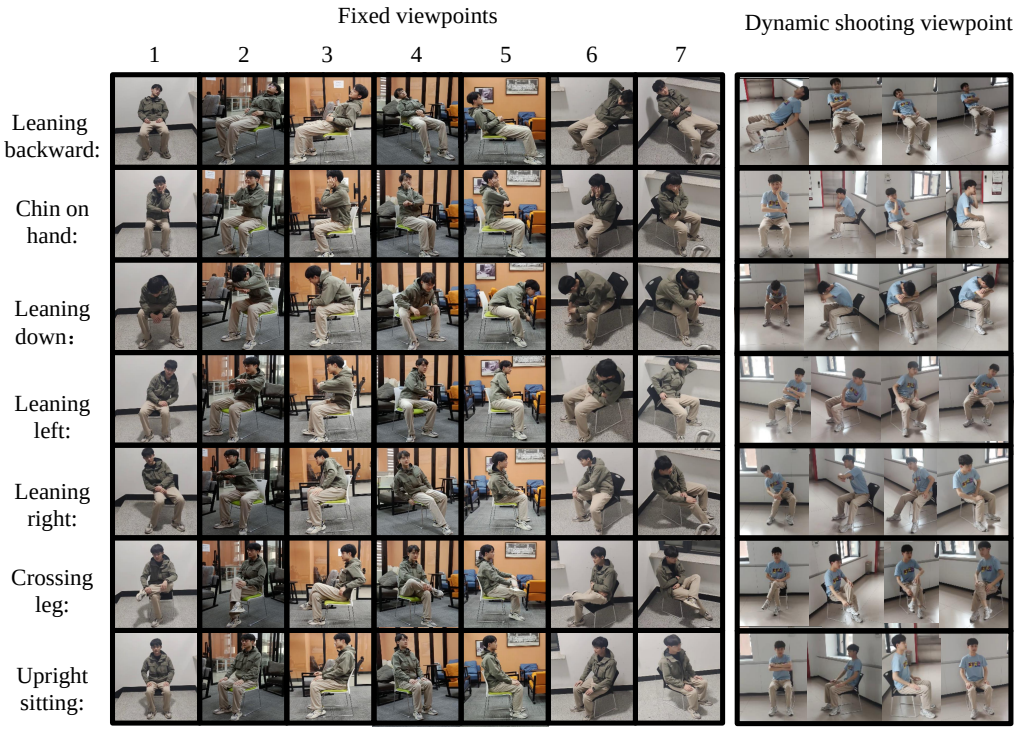


Figure 5. An illustration of the 7 postures in the MVSP dataset from different viewpoints.

Human Abnormal Sitting Posture dataset (HASP) [22]: The HASP dataset contains 700 video clips, 600 of which are used for training. Each clip consists of 80 frames that transition from normal to abnormal sitting postures, all captured from a frontal viewpoint. The dataset defines six types of abnormal sitting postures. To adapt the dataset for our experiments, we manually select 8244 images from these clips, categorizing them into seven classes by including a "normal sitting posture" class in addition to the original six abnormal postures.

MSRA Hands Dataset (MSRA) [28]: MSRA includes 76,500 depth images with 17 hand gestures, each annotated with 3D keypoint coordinates. The dataset's significant viewpoint variations offer an excellent benchmark for assessing the model's multi-view recognition capability.

Data processing methods: In the subsequent experiments, the comparison models are primarily divided into four categories, with details described in Section 4.2. To ensure that the keypoint data meet the input requirements of the other three strategies, we adopt the following data processing method.

To ensure the skeleton images constructed from keypoints remain within a reasonable range in the frame, we apply the following transformation to the keypoint data:

$$P = 0.7 \cdot Size \frac{P - P_{\text{cent}}}{Scale} + 0.5 \cdot Size, \tag{16}$$

where $Size$ denotes the image size (set to 400), and the definitions of P_{cent} and $Scale$ are given in Section 3.1. To preserve depth information in the skeleton images, we use grayscale intensity to represent depth, with the darkest points (deepest depth) assigned a value of 100 and the lightest points (shallowest depth) assigned a value of 255. The grayscale intensity of the background is 0. The resulting skeleton images are shown in Figure 6.

To adapt the keypoint data for input into PointNets, we construct the point cloud data of 1024 points from 17 keypoints. Each segment between corresponding keypoints is considered a limb according to the natural skeletal connections of the human body. These limbs are then modeled as cylindrical surfaces, with points on these surfaces undergoing

slight random perturbations to ensure a more uniform distribution across the cylinders. The resulting point cloud data are illustrated in Figure 6.

Finally, to enable ST-GCNs to process individual keyframe data, we repeat the single-frame keypoint data across multiple frames, treating them as a sequence of static action. The resulting spatiotemporal sequence data are illustrated in Figure 6.

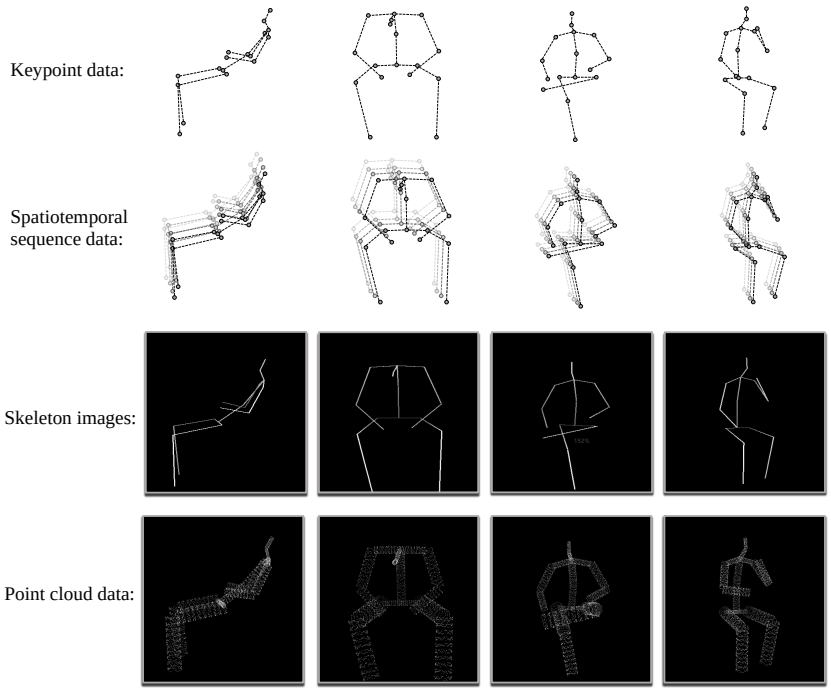


Figure 6. Illustrations of keypoint data, skeleton images, point cloud data, and spatiotemporal sequence data.

Experimental setup: For all the experiments in this section, we select the AdamW optimizer with a learning rate set to 0.0001, a weight decay rate set to 0.001, a batch size of 256, and a cross entropy loss function for 50 epochs of training. We use the learning rate scheduler to make the learning rate become 0.9 times the original every five epochs. For the MVS algorithm, the parameters Xs, Ys, Zs, φ, ω, and κ are randomly sampled within the data range specified in Equation (5), ensuring diverse viewpoint augmentation for robust multi-view recognition. The hardware platform used in our experiment is as follows: CPU: Intel (R) Xeon (R) Gold 6254; GPU: NVIDIA GeForce RTX 3090; CUDA Version: 11.4; RAM: 629 G.

4.2. Comparison with Other Methods

To evaluate the effectiveness of our method, we conduct experiments on three datasets, namely, MVSP, HASP, and MSRA. The selected comparison baselines are divided into four categories, namely, skeleton image + CNNs, point cloud + PointNets, spatiotemporal sequence + ST-GCNs, and direct keypoint processing. The following are the reasons why we chose these strategies:

1. Skeleton image + CNNs and spatiotemporal sequence+ST-GCNs: they are currently the most researched and effective methods, which can show the advantages of our method compared with other studies.
2. Point cloud + PointNets: because KeypointNet is improved based on PointNets, we choose PointNets to clearly demonstrate the enhancements and optimizations introduced in our approach.

3. Direct keypoint processing: Because our method directly processes keypoint, we choose other methods that also directly process keypoints to demonstrate our advantages. MLP is a representative example of such methods.

A detailed description of the comparison methods is shown in Table 2. The experimental results of these sitting posture recognition methods across different datasets are presented in Table 3.

Table 2. Descriptions of the comparison methods used in the experiments.

Strategy Description	Method	Method Description
Skeleton image + CNNs draw the keypoint data into a skeleton image and then use the CNN method for recognition, thereby modeling it into an image classification task.	ResNet (2016) [44]	A very classic and universal method in the field of CNN.
	EfficientNetV2 (2022) [45]	A more efficient and advanced CNN method.
	MobileVit (2022) [46]	A CNN method based on transformer architecture that is suitable for low data complexity.
	LMSPNet (2023) [20]	A lightweight advanced CNN method for posture recognition.
Point cloud + PointNets encapsulate keypoint data into point cloud data and then use PointNets for recognition and convert it into a point cloud classification task.	PointNet (2017) [39]	The founder of the PointNet series.
	PointNet++ (2017) [40]	On the basis of PointNet, a paradigm for multi-scale feature capture is introduced.
	PointMLP (2022) [41]	A more advanced method in the PointNet series.
Spatiotemporal sequence + ST-GCNs overlap the keypoint data in the time dimension, convert them into spatiotemporal sequence data, and then use ST-GCNs for posture recognition.	AGCN (2021) [22]	A classic method of ST-GCN.
	ASPR (2023) [47]	An advanced method for abnormal sitting posture recognition using ST-GCN.
Direct keypoint processing learn keypoint data directly without modeling the data representations.	MLP-S (2024) [30]	A smaller MLP.
	MLP-L	We modified MLP-S by expanding the embedding dimension of each layer by 5 times.

Table 3. Comparison of accuracy and speed with existing methods.

Strategy	Method	MVSP	HASP	MSRA	Inference Time (ms)	Params	FLOPS
Skeleton image + CNNs	ResNet	93.80%	37.88%	96.59%	4.65	11.8 M	1.82 G
	EfficientNetV2	93.03%	51.48%	98.73%	15.04	21.5 M	2.9 G
	MobilVit	94.06%	36.94%	95.49%	6.39	1.2 M	157.3 M
	LMSPNet	94.24%	70.52%	98.62%	7.36	3.9 M	149.4 M
Point cloud + PointNets	PointNet	70.45%	87.99%	96.83%	1.41	3.5 M	450.4 M
	PointNet++	62.54%	90.40%	99.03%	146.55	1.7 M	4.03 G
	PointMLP	69.76%	91.24%	99.14%	74.61	12.6 M	15.73 G
Spatiotemporal sequence + ST-GCNs	AGCN	93.80%	83.88%	97.64%	9.04	0.87 M	425.7 M
	ASPR	95.36%	95.66%	98.79%	7.81	0.65 M	428.8 M
Direct keypoint processing	MLP-S	92.03%	79.29%	93.18%	0.27	0.18 M	0.18 M
	MLP-L	93.83%	73.64%	93.58%	0.34	5.25 M	5.26 M
	KeypointNet (ours)	98.46%	96.52%	99.48%	1.21	5.4 M	5.0 M

For the MVSP dataset experiment, which involves data from multiple viewpoints, we apply MVS for data augmentation across all methods. The results demonstrate that our method clearly outperforms other methods, achieving an accuracy of 98.46%. Notably, the three PointNets models perform poorly on MVSP. A plausible reason for this is the significant variation in actions within the same class.

In the HASP dataset, where the viewpoint remains unchanged, no data augmentation is applied to any of the methods. Our model achieves an accuracy of 96.52% and outperforms other methods. It is worth noting that methods in the skeleton image + CNNs category, including ResNet, MobileVit, LMSPNet, and EfficientNetV2, perform poorly and cannot recognize effectively. We believe that this may be due to the small differences between many categories in HASP, making it challenging for these models to detect fine changes in image data. In addition, the accuracy of the two MLPs is also low, which may be due to the lack of sufficient flexibility and robustness of pure MLPs.

In the MSRA dataset, where data are from multiple viewpoints, we also apply MVS for data augmentation across all methods. Our model achieves the highest accuracy of 99.48%. This demonstrates that our model can handle human keypoint data effectively and also excels at hand keypoint data, showing excellent generalization ability. In addition, the accuracy values of the two methods relying solely on MLP are also low, which further demonstrates that KeypointNet is a better model for directly processing keypoint data.

To further evaluate the Multi-Scale Feature Extraction capabilities of KeypointNet, we conduct comparative experiments with hybrid models that combine CNNs and PointNets on MVSP. These hybrid models integrate features at different scales by concatenating them at the final MLP classification head. We select PointNet, KeypointNet, ResNet, and LMSPNet for these hybrid combinations. The experimental results are summarized in Table 4. As shown in the table, the accuracy of the hybrid models is slightly lower than the highest accuracy achieved by the individual models. This indicates that combining features from different scales does not yield more valuable information and instead introduces interference to the models that originally performed better. This further demonstrates the advantage of KeypointNet in Multi-Scale Feature Extraction.

Table 4. Comparison of different hybrid models.

PointNets	CNNs	Accuracy
PointNet	ResNet	92.87%
KeypointNet	ResNet	97.65%
PointNet	LMSPNet	94.03%
KeypointNet	LMSPNet	98.12%

Moreover, our model achieves a fast inference time of 1.21 ms and a low computational cost of 5.0 MFLOPS, trailing only behind MLP-S and MLP-L in terms of efficiency. Additionally, it boasts a compact parameter size, making it highly storage-efficient. It is worth noting that we adopt a Multi-Scale Feature Extraction strategy similar to PointNet++ but optimize it, resulting in significantly better inference speed and computational complexity than PointNet++. Specifically, our inference speed is approximately one hundred times faster than that of PointNet++. Additionally, to quantitatively demonstrate the advantages of directly using keypoint data for recognition, we compare it with the skeleton image-based strategy, aiming to illustrate the noise and redundancy issues inherent in such strategies. We primarily compare the two strategies in terms of data volume, the extent of data utilization, and processing time required. The results are shown in Table 5. It is evident that the amount of data for skeleton images is significantly larger compared to keypoints, with the difference potentially being several thousand times. Secondly, keypoint data utilize 100% of the output from 3D pose estimation, whereas skeleton images cannot achieve such high

data utilization rates. If the proportion of non-background pixels is calculated, our processing method yields a result of approximately 1.5%. Even with an optimistic estimate for other processing methods, this figure would not exceed 10%, indicating that a substantial amount of information is wasted. Redundant background information can be considered a form of noise. Additionally, in terms of processing time, processing skeleton images takes much longer than normalizing keypoints and can even exceed the time required for subsequent CNN model inference. This undoubtedly introduces redundancy.

Table 5. Quantitative comparison of direct keypoint processing and skeleton image-based methods.

Strategy	Data Volume	Extent of Data Utilization	Processing Time Required
Direct keypoint processing	17×3	100%	0.21 ms
Skeleton image-based	$400 \times 400 \times 3$	<10%	28.8 ms

4.3. Ablation Studies

To assess the effectiveness of our proposed network structure, we conduct ablation studies on the MVSP dataset. Five different structure combinations are tested, and the results are shown in Table 6. Combination 1 is a direct imitation of the PointNet++ structure, where our six proposed local regions are processed using 1D convolutions+MLPs for initial extraction, followed by MLP for classification. Compared to Combination 1, Combination 2 first processes the input through KFE before applying 1D convolutions for local feature extraction. Combination 3 abandons the 1D convolution for local feature extraction, leaving only MLP. Combinations 4 and 5 introduce GFE based on Combinations 3 and 2, respectively. Comparing Combination 1 with Combination 2 shows that performing initial feature extraction using KFE before extracting local features is more effective, with the accuracy improving from 97.48% to 98.07%. Comparing Combination 2 with Combination 3 demonstrates that using 1D convolutions before MLPs for local feature extraction is a superior choice as it increases accuracy by 0.83%. Furthermore, by examining Combinations 4 and 5, it becomes clear that GFE further optimizes the extraction and aggregation of keypoint features, with accuracy improving from 97.24% and 98.07% to 98.25% and 98.41%, respectively. This also illustrates that using 1D convolutions is a better choice for extracting local features.

Table 6. The ablation study on the network module

Combinations	Keypoint Feature Extraction Module	Local Feature Extraction Module	Global Feature Extraction Module	Accuracy
1		Conv1d + MLP		97.48%
2	✓	Conv1d + MLP		98.07%
3	✓	MLP		97.24%
4	✓	MLP	✓	98.25%
5	✓	Conv1d + MLP	✓	98.46%

✓ represents the presence of this structure in the network.

4.4. The Effectiveness of MVS

To evaluate the actual effectiveness and generalizability of MVS, we conduct experiments on the MVSP dataset. Considering that CNNs and ST-GCNs perform relatively well on MVSP, we select LMSPNet and ASPR as representative methods to demonstrate the generalizability of MVS. In this experiment, we divide the fixed viewpoint data into seven parts according to the viewpoint and select a part of them as the training set each time, forming seven training sets. The test set always consists of data from dynamic shooting viewpoints. This division method can assess the performance of MVS under varying

degrees of viewpoint deficiency. The experimental results are shown in Figure 7. It can be observed that, first, KeypointNet consistently outperforms the other two methods, both with and without the application of MVS. This highlights the superior performance of KeypointNet across different conditions. Second, after applying MVS, all three methods show improvements in accuracy, demonstrating the broad applicability of our proprietary MVS algorithm. It is not only suitable for KeypointNet but also effective for other types of models, such as CNNs or ST-GCNs. Third, MVS demonstrates high-level performance in dealing with varying degrees of viewpoint deficiency.

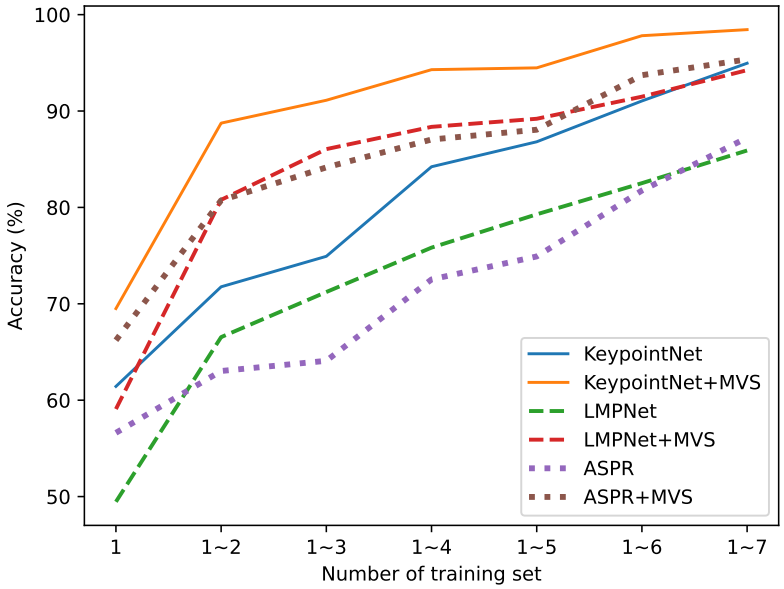


Figure 7. The performance of the MVS algorithm.

To further compare our MVS algorithm with standard data augmentation techniques, we utilized KeypointNet as the network model and evaluated it against two common methods: random Gaussian noise and random affine transformations. For the experiments, the Gaussian noise was set with a standard deviation of 0.1, and the random affine transformations were configured with the following parameters: scale range = (0.8, 1.2), rotation range = $(-30, 30)$, translate range = $(-10, 10)$, and shear range = $(-10, 10)$. The experimental results are shown in Figure 8. Overall, the MVS algorithm significantly outperforms the other two data augmentation methods. This is particularly evident when the number of viewpoints is small, where MVS enhances the multi-view representation capability of the data, achieving an accuracy exceeding 85%. In contrast, the accuracies of the other two methods remain below 75% under these conditions. Additionally, random affine transformations show relatively better performance when the number of viewpoints is limited. However, as the number of viewpoints increases, the disturbances introduced by affine transformations outweigh their benefits, leading to a decline in model performance. On the other hand, the performance of Gaussian noise remains relatively stable across different numbers of viewpoints.

To further investigate the robustness of the MVS algorithm, we used KeypointNet as the network model and simulated two types of perturbations: random occlusion and random body size variation. Specifically, random occlusion randomly adds black squares with side lengths ranging from 10 to 100 pixels in the original images. For random body size variation, with a 50% probability, the lengths of the arms and legs are scaled to between 70% and 130% of their original lengths while maintaining the pose. Similarly, with a 50% probability, the shoulder width is scaled to between 70% and 130% of its original

width, and with a 50% probability, the hip width is scaled to between 70% and 130% of its original width. The experimental results are shown in Figure 9. The experimental results indicate that the two different types of perturbations have relatively small impacts on the accuracy of MVS, demonstrating the notable robustness of the MVS algorithm. Among the different types of perturbations, random occlusion has a relatively minor impact. This can be attributed to the inherent robustness of 3D pose estimation algorithms, which are designed to handle occlusion issues effectively. In contrast, random body size variation has a relatively larger impact, although it still remains within an acceptable range. Furthermore, we observe that as the number of viewpoints and the quantity of data in the training set increase, the effects of these perturbations becomes less pronounced. For example, when the training set includes only one viewpoint, there is approximately a 6% difference in accuracy compared to the baseline. However, when the training set encompasses seven viewpoints, this accuracy gap narrows significantly to less than 3%. These findings suggest that the model acquires enhanced robustness by learning from a more diverse and comprehensive dataset, thereby improving its ability to generalize under various conditions.

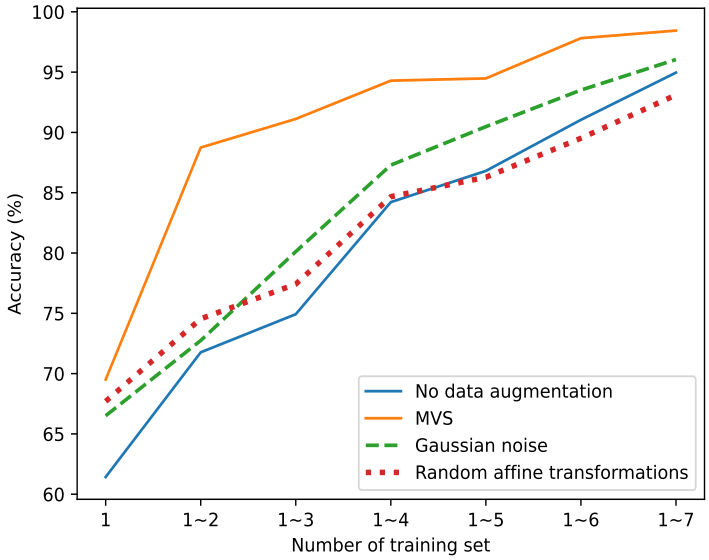


Figure 8. Comparison of MVS with other data augmentation techniques.

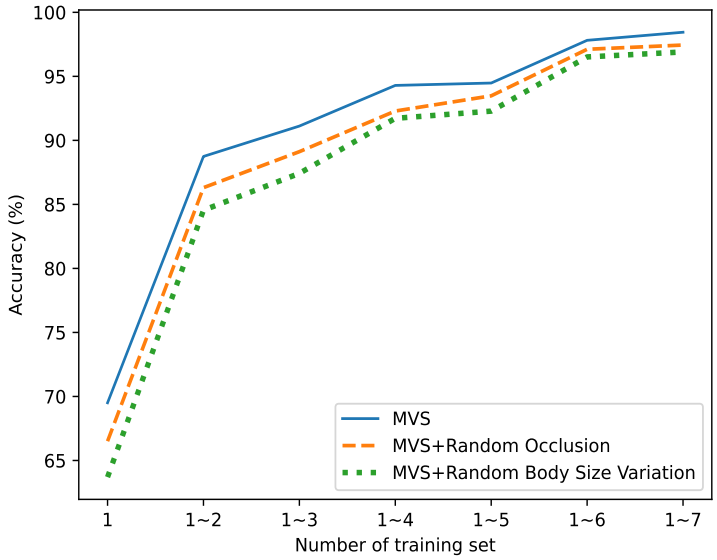


Figure 9. Result of Multi-View Simulation algorithm robustness experiments.

4.5. Experiments on Viewpoint Limitedness

To further evaluate KeypointNet's performance with limited training viewpoints, we conduct further experiments on the MVSP dataset. Specifically, we divide the data from each of the original seven fixed viewpoints of MVSP into training and testing sets in a 4:1 ratio. The training set starts with data from a single viewpoint and is progressively expanded by adding data from additional viewpoints, eventually incorporating all seven viewpoints. The test set comprises 20% of the data from all seven viewpoints. The reason why we do not continue to use the dynamic shooting viewpoint as the test set is that the viewpoint situation of the dynamic shooting viewpoint is too complex, which is not conducive to specific analysis. The dataset partitioning method and the experimental results are shown in Figure 10. It can be observed that as the number of viewpoints in the training set increases, the model's performance on the test set gradually improves. Notably, the absence of training data for certain viewpoints results in the network's inability to effectively recognize those viewpoints, which means the model's performance is degraded when dealing with low-quality data or extreme viewpoint variations. For example, when the training set contains viewpoints 1, 2, and 3, the KeypointNet achieves high-level recognition of viewpoints 1, 2, and 3 but exhibits accuracy below 90% for viewpoints 5, 6, and 7. This indicates that, despite KeypointNet achieving relatively superior multi-view recognition performance, achieving high accuracy in classifying viewpoints that are absent from the training set remains a challenging task. When relying solely on RGB images and 3D pose estimation methods, pure visual techniques cannot easily achieve perfect multi-view recognition. We believe that part of the reason for this limitation is that RGB images lack depth information, and 3D pose estimation methods have certain errors when performing depth predictions, leading to incorrect 3D keypoint coordinates. Additionally, the absence of camera pose information makes it impossible to align data from different viewpoints into a consistent representation. As a result, the network tends to perform targeted learning on the data present in the training set, lacking sufficient generalization capability.

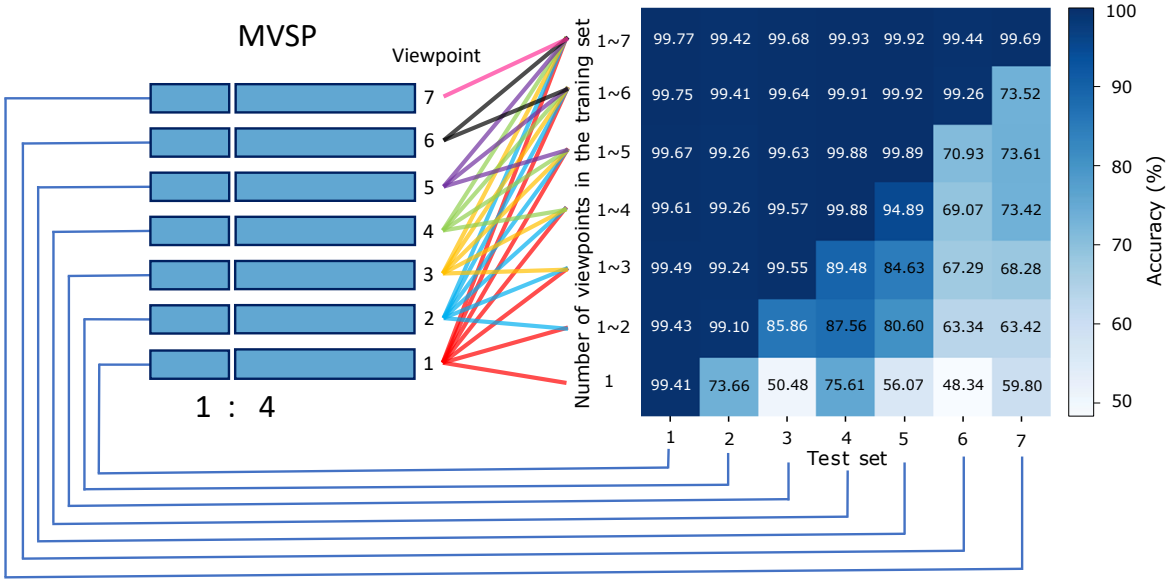


Figure 10. The results of the experiment on viewpoint limitedness.

4.6. Data Sampling Rate Experiment

To explore the model's performance when faced with reduced data volume and to evaluate whether the volume of the proposed dataset is sufficient as Liu et al. [21] mentioned, we conduct experiments on MVSP using a dataset sampling coefficient s. This coefficient ranges from 1 to 0.001 and controls the proportion of training data sampled from

each viewpoint. We evaluate KeypointNet, LMSPNet, and ASPR under these conditions for comparison. The experimental results are shown in Figure 11. Compared to other methods, KeypointNet shows a slower decrease in accuracy as the amount of training data decreases. When the data volume is reduced to only 10%, the accuracy of all three methods experiences only a slight decline. Even when the data volume is reduced to 1%, KeypointNet still maintains an accuracy of approximately 95%, whereas the accuracy of LMSPNet and ASPR drops below 85%. A significant drop in KeypointNet's accuracy only occurs when the data volume decreases to 0.6%. When $s = 0.001$, the accuracy of all models drastically decreases. KeypointNet performs relatively the best, still achieving over 60% accuracy, while LMSPNet's accuracy drops to around 50% and ASPR's accuracy falls below 40%. We believe the poor performance when dealing with low-quality data is predictable because the amount of training data is less than 40 samples. Considering that there are seven actions and seven viewpoints, this means that each viewpoint has fewer than one sample per action, significantly increasing the difficulty for the model to learn effectively.

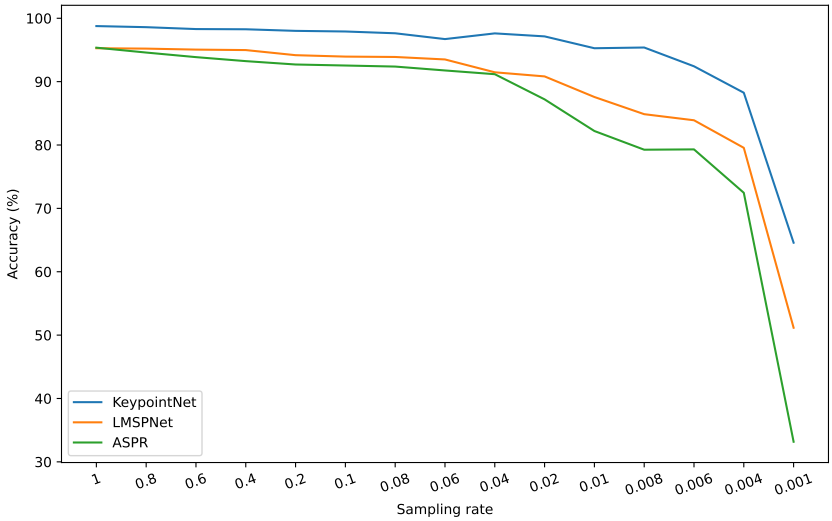


Figure 11. The results of the data sampling rate experiment.

This indicates that our method extracts more essential knowledge from the data compared to CNN methods or ST-GCN methods, rather than relying solely on large data volume to achieve higher accuracy. Additionally, the experiment demonstrates that the volume of MVSP is sufficient to some extent because none of the models experience a significant drop in accuracy when the data volume is reduced to 10%.

5. Conclusions and Discussion

In this paper, our primary objectives are to better address two limitations. The first is the complexity of multi-view recognition and the inadequacy of the number of viewpoints of the data. The second is the lack of an efficient model for directly processing keypoints. To tackle these issues, we propose KeypointNet and MVS. Specifically, KeypointNet is an efficient network for processing human keypoint data. KeypointNet comprises KFE and MSFE. By decoupling feature extraction, the first module focuses on within-point keypoint features, while the second one captures spatial relationships and between-point features. In addition, MVS is a data augmentation technique designed based on photogrammetry. It is used to simulate keypoints from additional viewpoints through computation, thereby enhancing the multi-view recognition capability. The experimental results on both our proposed MVSP and other public datasets demonstrate the effectiveness of KeypointNet and MVS.

However, multi-view recognition remains a challenge. Future work will explore using deep learning to process data from various viewpoints, including building comprehensive datasets and enabling networks to learn from error-prone keypoint data to construct better data for other viewpoints. Additionally, considering the significant body of research focused on temporal variations and action recognition, extending KeypointNet to better handle temporal variations in keypoint data is a promising direction. Future work could explore integrating advanced temporal modeling methods, such as transformers or ST-GCNs, to enhance KeypointNet's capability in handling sequential data effectively.

Author Contributions: Conceptualization, Z.C.; methodology, Z.C. and X.W.; software, Z.C.; validation, C.W., Y.X., and S.J.; formal analysis, Z.C. and Y.Z. (Yu Zhang); investigation, Z.C.; resources, Y.Z. (Yu Zhang).; data curation, Z.C.; writing—original draft preparation, Z.C.; writing—review and editing, X.W.; visualization, Z.C.; supervision, Y.Z. (You Zhou); project administration, Z.C.; funding acquisition, Y.Z. (You Zhou). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Jilin Provincial Department of Science and Technology Project under Grant 20230201083GX, Grant 20240101369JC, and Grant 20240302086GX and in part by the Nanyang Associate Professorship.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

Acknowledgments: We would like to thank the Computing Center of Jilin Province and the High Performance Computing Center of Jilin University for their technical support.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cabral, A.M.; Carreira Moreira, R.d.F.; de Barros, F.C.; Sato, T.d.O. Is physical capacity associated with the occurrence of musculoskeletal symptoms among office workers? A cross-sectional study. *Int. Arch. Occup. Environ. Health* **2019**, *92*, 1159–1172. [CrossRef] [PubMed]
2. Gebreyesus, T.; Nigussie, K.; Gashaw, M.; Janakiraman, B. The prevalence and risk factors of work-related musculoskeletal disorders among adults in Ethiopia: A study protocol for extending a systematic review with meta-analysis of observational studies. *Syst. Rev.* **2020**, *9*, 136. [CrossRef]
3. Hu, Y.; Huang, T.; Zhang, H.; Lin, H.; Zhang, Y.; Ke, L.; Cao, W.; Hu, K.; Ding, Y.; Wang, X.; et al. Ultrasensitive and Wearable Carbon Hybrid Fiber Devices as Robust Intelligent Sensors. *ACS Appl. Mater. Interfaces* **2021**, *13*, 23905–23914. [CrossRef] [PubMed]
4. Buckland, A.J.; Beaubrun, B.M.; Isaacs, E.; Moon, J.; Zhou, P.; Horn, S.; Poorman, G.; Tishelman, J.C.; Day, L.M.; Errico, T.J.; et al. Psoas morphology differs between supine and sitting magnetic resonance imaging lumbar spine: Implications for lateral lumbar interbody fusion. *Asian Spine J.* **2018**, *12*, 29–36. [CrossRef]
5. Urgo, M.; Berardinucci, F.; Zheng, P.; Wang, L. AI-Based Pose Estimation of Human Operators in Manufacturing Environments. In *Lecture Notes in Mechanical Engineering*; Springer: Cham, Switzerland, 2024; pp. 3–38.
6. Latreche, A.; Kelaiaia, R.; Chemori, A. AI-based Human Tracking for Remote Rehabilitation Progress Monitoring. In *AIJR Abstracts*; AIJR Publisher: Balrampur, India, 2024; pp. 7–9.
7. Hoang, D.C.; Nguyen, A.N.; Nguyen, T.U.; Hoang, N.A.; Vu, V.D.; Vu, D.Q.; Ngo, P.Q.; Phan, K.T.; Tran, D.T.; Nguyen, V.T.; et al. Attention-based hand pose estimation with voting and dual modalities. *Eng. Appl. Artif. Intell.* **2025**, *139*, 109526. [CrossRef]
8. Biswas, S.; Nandy, A.; Naskar, A.K.; Saw, R. MediaPipe with LSTM Architecture for Real-Time Hand Gesture Recognization. In *Computer Vision and Image Processing, Proceedings of the 8th International Conference, CVIP 2023, Jammu, India, 3–5 November 2023*; Springer: Cham, Switzerland, 2024; Volume 2010, pp. 422–431. [CrossRef]
9. Zhang, W.; Li, Y.; Cai, S.; Wang, Z.; Cheng, X.; Somjit, N.; Sun, D.; Chen, F. Combined MediaPipe and YOLOv5 range of motion assessment system for spinal diseases and frozen shoulder. *Sci. Rep.* **2024**, *14*, 15879. [CrossRef] [PubMed]

10. Husna, R.; Brata, K.C.; Anggraini, I.T.; Funabiki, N.; Rahmadani, A.A.; Fan, C.P. An Investigation of Hand Gestures for Controlling Video Games in a Rehabilitation Exergame System. *Computers* **2025**, *14*, 25. [CrossRef]
11. Ganai, U.J.; Ratne, A.; Bhushan, B.; Venkatesh, K.S. Early detection of autism spectrum disorder: Gait deviations and machine learning. *Sci. Rep.* **2025**, *15*, 873. [CrossRef] [PubMed]
12. Syed, I.S.; Matheswaran, K.; Li, C.Q. Integrating Mediapipe Module for Learn From Demonstration on Cobots. In Proceedings of the 2024 International Symposium on Flexble Automation, ISFA 2024, Seattle, WA, USA, 21–24 July 2024.
13. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 172–186. [CrossRef]
14. Li, W.; Liu, H.; Tang, H.; Wang, P.; Van Gool, L. MHFormer: Multi-Hypothesis Transformer for 3D Human Pose Estimation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 13137–13146. [CrossRef]
15. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep High-Resolution Representation Learning for Human Pose Estimation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)), Long Beach, CA, USA, 15–20 June 2019; pp. 5686–5696. [CrossRef]
16. Cheng, B.; Xiao, B.; Wang, J.; Shi, H.; Huang, T.S.; Zhang, L. HigherHRNet: Scale-Aware Representation Learning for Bottom-Up Human Pose Estimation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 5385–5394. [CrossRef]
17. Bazarevsky, V.; Grishchenko, I.; Raveendran, K.; Zhu, T.; Zhang, F.; Grundmann, M. BlazePose: On-device Real-time Body Pose tracking. *arXiv* **2020**, arXiv:2006.10204.
18. Nadeem, M.; Elbasi, E.; Zreikat, A.I.; Sharsheer, M. Sitting Posture Recognition Systems: Comprehensive Literature Review and Analysis. *Appl. Sci.* **2024**, *14*, 8557. [CrossRef]
19. Fang, Y.; Shi, S.; Fang, J.; Yin, W. SPRNet: Sitting Posture Recognition Using improved Vision Transformer. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022. [CrossRef]
20. Jiao, S.; Xiao, Y.; Wu, X.; Liang, Y.; Liang, Y.; Zhou, Y. LMSPNet: Improved Lightweight Network for Multi-Person Sitting Posture Recognition. In Proceedings of the 2023 IEEE 3rd International Conference on Computer Communication and Artificial Intelligence (CCAI), Taiyuan, China, 26–28 May 2023; pp. 289–295.
21. Liu, J.; Wang, Y.; Liu, Y.; Xiang, S.; Pan, C. 3D PostureNet: A unified framework for skeleton-based posture recognition. *Pattern Recognit. Lett.* **2020**, *140*, 143–149. [CrossRef]
22. Li, L.; Yang, G.; Li, Y.; Zhu, D.; He, L. Abnormal sitting posture recognition based on multi-scale spatiotemporal features of skeleton graph. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106374. [CrossRef]
23. Dong, C.; Leu, M.C.; Yin, Z. American Sign Language Alphabet Recognition Using Microsoft Kinect. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston, MA, USA, 7–12 June 2015.
24. Elforaici, M.E.A.; Chaaraoui, I.; Bouachir, W.; Ouakrim, Y.; Mezghani, N. Posture recognition using an RGB-D camera: Exploring 3D body modeling and deep learning approaches. In Proceedings of the 2018 IEEE Life Sciences Conference (LSC), Montreal, QC, Canada, 28–30 October 2018; pp. 69–72.
25. Le, T.L.; Nguyen, M.Q.; Nguyen, T.T.M. Human posture recognition using human skeleton provided by Kinect. In Proceedings of the 2013 International Conference on Computing, Management and Telecommunications (COMMANTEL), Ho Chi Minh City, Vietnam, 21–24 January 2013; pp. 340–345.
26. Lin, B.S.; Lee, I.J.; Fahn, C.S.; Lee, Y.F.; Chou, W.J.; Wu, M.L. Depth-Camera Based Energy Expenditure Estimation System for Physical Activity Using Posture Classification Algorithm. *Sensors* **2021**, *21*, 4216. [CrossRef] [PubMed]
27. Schenk, T. *Introduction to Photogrammetry*; Department of Civil and Environmental Engineering and Geodetic Science, The Ohio State University: Columbus, OH, USA, 2005; pp. 60–61.
28. Sun, X.; Wei, Y.; Liang, S.; Tang, X.; Sun, J. Cascaded Hand Pose Regression. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 824–832.
29. Bustamante, A.; Belmonte, L.M.; Morales, R.; Pereira, A.; Fernandez-Caballero, A. Bridging the Appearance Domain Gap in Elderly Posture Recognition with YOLOv9. *Appl. Sci.* **2024**, *14*, 9695. [CrossRef]
30. Zhao, S.; Su, Y. Sitting Posture Recognition Based on the Computer's Camera. In Proceedings of the 2024 2ND Asia Conference on Computer Vision, Image Processing and Pattern Recognition, CVIPPR 2024, Xiamen, China, 26–28 April 2024. [CrossRef]
31. Ji, H.; Xie, J.; Sun, P. A Lightweight IoT Device-friendly Anomaly Sitting Posture Detector for Protecting Adolescent Bone Development. In Proceedings of the International Conferences on Internet of Things (iThings)/Green Computing and Communications (GreenCom)/Cyber, Physical and Social Computing (CPSCom)/Smart Data, SmartData and Congress on Cybermatics (Cybermatics), Danzhou, China, 17–21 December 2024; pp. 233–238. [CrossRef]
32. Markova, V.; Markov, M.; Petrova, Z.; Filkova, S. Assessing the Impact of Prolonged Sitting and Poor Posture on Lower Back Pain: A Photogrammetric and Machine Learning Approach. *Computers* **2024**, *13*, 231. [CrossRef]

33. Carneros-Prado, D.; Cabanero-Gomez, L.; Johnson, E.; Gonzalez, I.; Fontecha, J.; Hervas, R. A Comparison Between Multilayer Perceptrons and Kolmogorov-Arnold Networks for Multi-Task Classification in Sitting Posture Recognition. *IEEE Access* **2024**, *12*, 180198–180209. [CrossRef]
34. Ye, Y.; Shi, S.; Zhao, T.; Qiu, K.; Lan, T. Patches Channel Attention for Human Sitting Posture Recognition. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2023, Heraklion, Greece, 26–29 September 2023; pp. 358–370.
35. Atvar, A.; Cinbis, N.I. Classification of Human Poses and Orientations with Deep Learning. In Proceedings of the 2018 26TH Signal Processing and Communications Applications Conference (SIU), Izmir, Turkey, 2–5 May 2018.
36. Chen, H.; Zendehdel, N.; Leu, M.C.; Moniruzzaman, M.; Yin, Z.; Hajmohammadi, S. Repetitive Action Counting Through Joint Angle Analysis and Video Transformer Techniques. In Proceedings of the 2024 International Symposium on Flexible Automation, ISFA 2024, Seattle, WA, USA, 21–24 July 2024.
37. Xu, M.; Guo, L.; Wu, H.C. Robust Abnormal Human-Posture Recognition Using OpenPose and Multiview Cross-Information. *IEEE Sens. J.* **2023**, *23*, 12370–12379. [CrossRef]
38. Abdullah, S.; Ahmed, S.; Choi, C.; Cho, S.H. Distance and Angle Insensitive Radar-Based Multi-Human Posture Recognition Using Deep Learning. *Sensors* **2024**, *24*, 7250. [CrossRef] [PubMed]
39. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85. [CrossRef]
40. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet plus plus: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
41. Ma, X.; Qin, C.; You, H.; Ran, H.; Fu, Y. Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework. *arXiv* **2022**, arXiv:2202.07123.
42. Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.A.A.K.; Elhoseiny, M.; Ghanem, B. PointNeXt: Revisiting PointNet plus plus with Improved Training and Scaling Strategies. In Proceedings of the Advances in Neural Information Processing Systems 35 (NEURIPS 2022), New Orleans, LA, USA, 28 November–9 December 2022.
43. Ionescu, C.; Papava, D.; Olaru, V.; Sminchisescu, C. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 1325–1339. [CrossRef] [PubMed]
44. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
45. Tan, M.; Le, V.Q. EfficientNetV2: Smaller Models and Faster Training. In Proceedings of the International Conference on Machine Learning (ICML), Virtual, 18–24 July 2021; Volume 139, pp. 7102–7110.
46. Mehta, S.; Rastegari, M. MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer. *arXiv* **2021**, arXiv:2110.02178.
47. Peng, Z.; Liu, H.; Jia, Y.; Hou, J. Attention-driven Graph Clustering Network. In Proceedings of the Processdings of the 29TH ACM International Conference on Multimedia, MM 2021, Virtual, 20–24 October 2021; pp. 935–943. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.