Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

FaSRnet: a feature and semantics refinement network for human pose estimation*

Yuanhong ZHONG[†‡1], Qianfeng XU[1], Daidi ZHONG[†2], Xun YANG[3], Shanshan WANG[4]

[1]School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China

[2]Bioengineering College of Chongqing University, Chongqing University, Chongqing 400044, China

[3]School of Information Science and Technology, University of Science and Technology of China, Hefei 230039, China

[4]Institutes of Physical Science and Information Technology, Anhui University, Hefei 230039, China

[†]E-mail: zhongyh@cqu.edu.cn; daidi.zhong@cqu.edu.cn

Received Dec. 12, 2022; Revision accepted June 27, 2023; Crosschecked Mar. 20, 2024

Abstract: Due to factors such as motion blur, video out-of-focus, and occlusion, multi-frame human pose estimation is a challenging task. Exploiting temporal consistency between consecutive frames is an efficient approach for addressing this issue. Currently, most methods explore temporal consistency through refinements of the final heatmaps. The heatmaps contain the semantics information of key points, and can improve the detection quality to a certain extent. However, they are generated by features, and feature-level refinements are rarely considered. In this paper, we propose a human pose estimation framework with refinements at the feature and semantics levels. We align auxiliary features with the features of the current frame to reduce the loss caused by different feature distributions. An attention mechanism is then used to fuse auxiliary features with current features. In terms of semantics, we use the difference information between adjacent heatmaps as auxiliary features to refine the current heatmaps. The method is validated on the large-scale benchmark datasets PoseTrack2017 and PoseTrack2018, and the results demonstrate the effectiveness of our method.

Key words: Human pose estimation; Multi-frame refinement; Heatmap and offset estimation; Feature alignment; Multi-person
https://doi.org/10.1631/FITEE.2200639 **CLC number:** TP391

1 Introduction

Human pose estimation is a popular subject in computer vision studies, with the goal of detecting and marking the positions of human key points (e.g., head and wrists) in an image. It has numerous applications in diverse domains, such as video surveillance, autonomous driving, and motion analysis (Insafutdinov et al., 2017; Li et al., 2018; Zheng et al., 2019; Fang ZJ and López, 2020). Human pose estimation has been developed rapidly with the establishment of large datasets (Sapp and Taskar, 2013; Andriluka et al., 2014; Lin et al., 2014) and deep learning (Wang M et al., 2012; Chu et al., 2017; Martinez et al., 2017; Yang X et al., 2017, 2018; Liu et al., 2019). However, existing high-accuracy methods (Weinzaepfel et al., 2013; Fang HS et al., 2017; Xiao et al., 2018; Sun et al., 2019; Cao et al., 2021) perform poorly when directly applied to video data. Many high-precision methods were designed based on static images; when applied to video data, they often struggle to achieve a satisfactory performance due to motion blur, defocus, and occlusion. The frames lose a lot of spatial information, leading to inaccurate detection results.

Due to the continuity of motion, multi-frame joint prediction is commonly used to improve output

[‡] Corresponding author

[*] Project supported by the National Key Research and Development Program of China (Nos. 2021YFC2009200 and 2023YFC3606100) and the Special Project of Technological Innovation and Application Development of Chongqing, China (No. cstc2019jscx-msxmX0167)

ORCID: Yuanhong ZHONG, https://orcid.org/0000-0001-5689-1146

accuracy. For example, some methods (Weinzaepfel et al., 2013; Pfister et al., 2015) use optical flow to capture the motion information in successive frames and combine it with graphic information for prediction. However, the essence of optical flow is to calculate the motion of pixels in the image. When the background changes rapidly, the effect is poor, and the calculation is time-consuming. Other methods (Luo et al., 2018) use long and short-term memory (LSTM) networks for pose estimation, but their training and inference require high hardware configuration, which limits their applicability. Tracking-based methods can locate the same human body in adjacent frames, but they require feature extraction, similarity calculation, and data association modules, which increase the computation and running time of the overall network. Besides, pose tracking is susceptible to occlusion and motion blur, which will significantly affect the accuracy of the outputs. Heatmaps, as the final output of general pose estimation networks, effectively represent the spatial distribution of key points while preserving their semantics information. Recent methods (Bertasius et al., 2019; Liu et al., 2021) use image-based human pose estimation networks as backbones and directly associate temporal information at the semantics level, achieving improved results.

However, there are still some issues. For instance, in cases of rapid movement or deformation, the positions of key points may undergo significant changes. In such scenarios, the semantics information alone may not accurately estimate the pose. The quality of heatmaps is influenced by corresponding features, and directly aggregating heatmaps at the semantics level yields unsatisfactory results (Fig. 1b). Therefore, we argue that it is necessary to associate and fuse temporal information at the feature level to better address these problems.

Recent studies have found that multiple low-quality predictions can be fused to generate a high-quality prediction. This process is similar to video super-resolution (Wang XT et al., 2019; Tian et al., 2020). In this process, there is usually a frame alignment operation to align the auxiliary features with the reference features. In the video sequence, when the human body is moving, the spatial position of features containing important information in the auxiliary frame is often inconsistent with that of the current frame. When convolution is used to aggregate the same position information for multi-frame features, the contribution of the auxiliary frame to the current frame will be reduced. When the auxiliary frame is aligned with the reference frame, the position of the effective feature is closer to that of the current frame, so the aggregation can achieve better results.

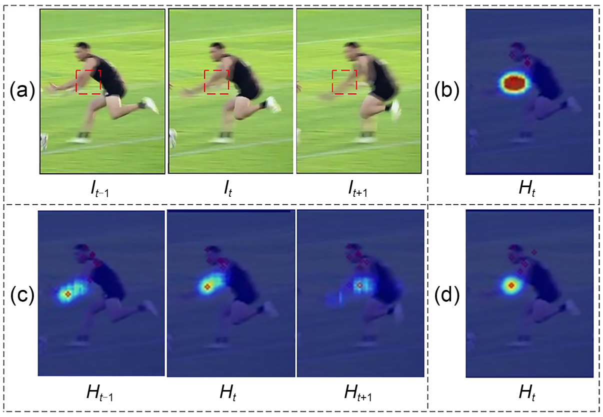

Fig. 1 Comparison of image- and video-based methods: (a) video sequence of motion blur, where the goal is to estimate the left hand marked by the red box in I_t; (b) rough heatmap generated by aggregation; (c) three rough heatmaps generated by the image-based method (Sun et al., 2019); (d) heatmap aggregated by our method (References to color refer to the online version of this figure)

Inspired by these methods, we propose a temporally consistent refinement architecture called the feature and semantics refinement network (FaSRnet). The backbone network is used to obtain feature and heatmap information from the previous frame, current frame, and next frame. In the feature refinement stage, the features of the previous and next frames are aligned with those of the current frame separately, resulting in aligned features. Next, the correlation coefficients between adjacent features are calculated using an attention mechanism, and weighted feature fusion is used to generate the feature-refined current feature, which is used to refine the heatmap of the current frame. The difference information between adjacent heatmaps can be used to refine the semantics features since these differences often reflect the positional motion information of key points in the video. In the semantics refinement stage, a subtraction is first conducted for the adjacent heatmaps, and motion change information is extracted. Then, the differential features are inputted, along with the aggregated adjacent heatmaps, into a fusion refinement module based on deformable convolution, to generate the final result. Our method refines the results at the feature and semantics levels simultaneously, decreases the impact of features extracted by the backbone network, and simplifies the overall network structure, yielding better results (Fig. 1d). Results from experiments on the widely used large-scale benchmark datasets PoseTrack2017 and PoseTrack2018 demonstrate the effectiveness of our method.

Our contributions can be summarized as follows:

1. We propose a feature and semantics refinement network (FaSRnet) for human pose estimation, which uses temporal information to refine the output at the semantics and feature levels, and is intuitive.

2. We design two components: feature refinement module (FRM) and semantics refinement module (SRM). FRM is used to effectively aggregate key point information in adjacent features; SRM uses difference information from adjacent heatmaps to refine the current heatmaps.

3. We have conducted extensive experiments on datasets PoseTrack2017 and PoseTrack2018 and achieved competitive results.

2 Related works

In this section, we briefly introduce two topics relevant to our work: video-based human pose estimation and semantics refinement.

2.1 Video-based human pose estimation

Video-based human pose estimation methods usually take advantage of temporal dependency among video frames (Pfister et al., 2015; Song et al., 2017; Girdhar et al., 2018; Shao et al., 2023). Methods based on optical flow estimate human posture by calculating optical flow between consecutive frames (Pfister et al., 2015). Pfister et al. (2015) used dense optical flow aligned heatmaps predictions from neighboring frames. Some approaches explore the structure of networks that exploit temporal information. Optical flow based methods are computationally expensive and time-consuming, and in some cases show performance degradation, such as occlusion and camera motion. Luo et al. (2018) built a recurrent architecture with LSTM to capture temporal geometric consistency and dependency among video frames for pose estimation. Recurrent neural networks (RNNs) can memorize well the information of a single person. With LSTM it is difficult to extract features that can improve the support frame, and the method has high time complexity. Girdhar et al. (2018) extended a three-dimensional (3D) convolutional neural network (CNN) architecture to integrate temporal information from adjacent video frames. This allows the 3D model to propagate useful information from contiguous frames. Liu et al. (2022) proposed to strengthen the correlation between multiple time frames by maximizing mutual information to better estimate the human body posture. Dang et al. (2022a) used the temporal correlations between joints to propose a plug-and-play kinematic modeling module (KMM) based on a domain-cross attention mechanism. KMM explicitly models the temporal relationships among different joints across video frames.

2.2 Semantics refinement

Since the outputs of the human pose estimation network are heatmaps containing semantics information, some methods began using the corresponding semantics information to improve the accuracy of the results. Cao et al. (2021) calculated the similarity between the connection and corresponding limbs of two key points to judge whether the key points belong to the same person. The human skeleton is a natural graph structure. Jin et al. (2020) and Wang J et al. (2020) used graph convolution to capture the relationship between key points, and then refined the output at the semantics level. Dang et al. (2022b) captured semantics-level guidance information, assisting in locating corresponding features in the next frame. Bertasius et al. (2019) and Liu et al. (2021) refined the current frame using the difference in heatmaps between the auxiliary frames and the current frame. However, the features of consecutive frames also affect output accuracy, and it may not be enough to refine only the heatmaps. Our method refines the result not only at the semantics level but also at the feature level.

3 Methodology

Given three consecutive frames I_{t-1}, I_t, and I_{t+1}, where I_t is the current frame and the others are auxiliary frames, the aim of the network is to estimate high-quality output pose P_t in I_t, approximating the ground-truth pose P_{g}. The pipeline of our proposed FaSRnet is illustrated in Fig. 2.

Since generally there are multiple persons in the image, the image needs to be preprocessed. We use a human detector to detect the persons in I_{t-1}, I_t, and I_{t+1}. In some human detection results, some body parts (such as palms and feet) far from the center of the human body cannot be completely included in the detection frame, resulting in incorrect detection. To avoid this, while remaining consistent with the previous approach, we do the following: we use the bounding box obtained from I_t to localize the same person in consecutive frames, and the bounding box is enlarged by 25%. Therefore, the human body in the image can be fully selected and multi-person pose estimation is transformed into single-person pose estimation. The input to the backbone network is the cropped video clips C^i_{t-1}, C^i_t, and C^i_{t+1} of human i.

The cropped video clips are fed into the backbone network to generate rough features F_{t-1}, F_t, and F_{t+1} and heatmaps H_{t-1}, H_t, and H_{t+1}. In the feature refinement stage, the generated features are fed into the feature alignment module to align the auxiliary frame features with the current frame feature. Then, the current feature and auxiliary features are fused in the attention fusion module to generate the refined current feature F^{c}_t. The heatmaps H^{c}_t refined at the feature level are generated by 1×1 convolution. During the semantics refinement phase, the module aggregates the difference between the refined heatmaps and the auxiliary heatmaps, and uses them as additional information to calibrate the heatmaps. Finally, the network outputs the final refined heatmaps. Next, we will introduce each module in turn.

3.1 Feature refinement module

3.1.1 Alignment with multiple receptive fields and deformable convolution

The motivations behind the design of our feature alignment module are as follows: the distributions of positions of valid features in adjacent frames differ due to human motion. As time goes by, the position of the human body in the image will change, and the poses of the persons will have a certain deformation. Generally, features with a large range of motion require a deeper network for extraction. However, feature fusion from multiple layers loses the details of features extracted by shallow layers. A shallower network

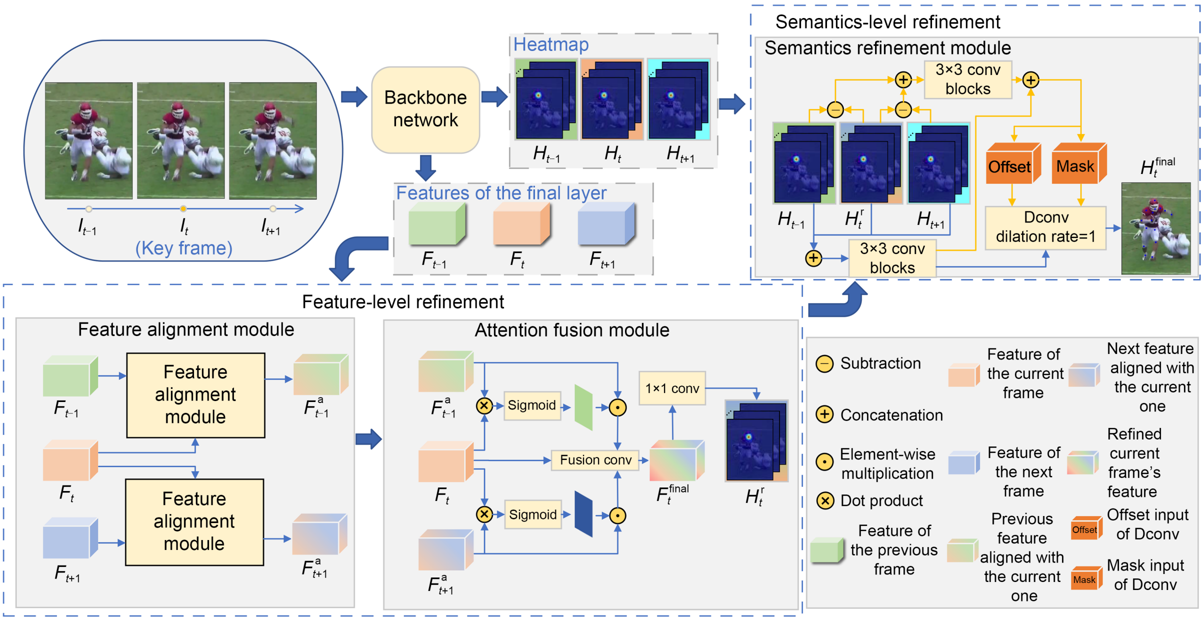

Fig. 2 The overall pipeline of our method. After the captured video sequence is inputted, the purpose is to identify the position of the human body in the current frame I_t. At the feature level, auxiliary features are aligned with current features and then fused with them through an attention mechanism. Finally, the heatmaps are generated by 1×1 convolution. At the semantics level, the current heatmaps are refined using the difference information between the heatmaps as auxiliary features

is used to align an object whose position changes in different frames to the same position, so that the object features can be in the same position.

Inspired by video super-resolution (Wang XT et al., 2019; Tian et al., 2020), we design a feature-level multi-receptive field feature alignment module, which uses a deformable convolution network (DCN) (Zhu et al., 2019) with different dilation rates to perform feature alignment at multiple scales. DCN has multiple inputs: input feature, offset feature, and mask feature.

The model structure is shown in Fig. 3. The yellow arrows indicate the calculated offset and mask features, which are generated by concatenating the current feature and auxiliary features. The blue arrows indicate the features to be aligned, and the input is auxiliary features. The final output of the module is the auxiliary features aligned with the key features. During feature alignment, this module repeatedly aligns auxiliary features with key features in different receptive fields to reduce the difference caused by feature distribution (the blue path in Fig. 3). We connect auxiliary feature F_{t+i} and key feature F_t, and feed them into the convolutional layer to generate the base offset/mask feature $O_0^{\mathrm{f}}/M_0^{\mathrm{f}}$ with the same number of channels as the input features. Formally, the feature $O_0^{\mathrm{f}}/M_0^{\mathrm{f}}$ is computed as

$$F_t \oplus F_{t+i} \xrightarrow[\text{CNN}]{3\times 3} O_0^{\mathrm{f}}/M_0^{\mathrm{f}}. \tag{1}$$

In the first branch, F_{t+i}'s are connected with $O_0^{\mathrm{f}}/M_0^{\mathrm{f}}$ and sent to the convolutional layer to generate the offset/mask feature $O_1^{\mathrm{f}}/M_1^{\mathrm{f}}$ of this branch. Then, DCN with a dilation rate of 5 accepts features F_{t+i}, O_1^{f}, and M_1^{f} to generate feature F_1^{a} aligned under a large receptive field. The whole process can be expressed as

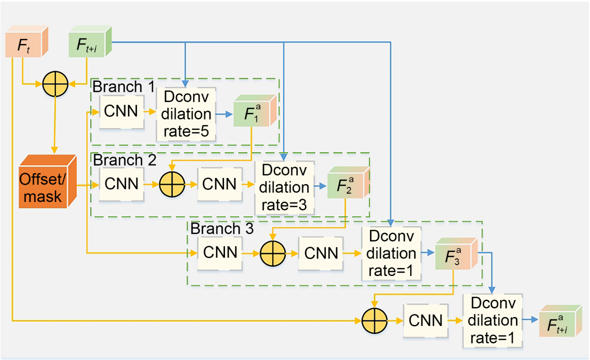

Fig. 3 The feature alignment module, taking the alignment of the previous feature and the current feature as an example (References to color refer to the online version of this figure)

$$F_{t+i} \oplus O_0^{\mathrm{f}}/M_0^{\mathrm{f}} \xrightarrow[\text{CNN}]{3\times 3} O_1^{\mathrm{f}}/M_1^{\mathrm{f}}, \tag{2}$$

$$(F_{t+i}, O_1^{\mathrm{f}}, M_1^{\mathrm{f}}) \xrightarrow[\text{DCN}]{\text{dilation rate} = 5} F_1^{\mathrm{a}}. \tag{3}$$

In branches 2 and 3, this module concatenates the generated alignment feature F_{n-1}^{a} with feature F_{t+i} and convolves to generate the aligned offset/mask feature $O_n^{\mathrm{f}}/M_n^{\mathrm{f}}$. Then features F_{t+i}, O_n^{f}, and M_n^{f} are fed into the DCN to generate aligned features. The whole process can be expressed as

$$F_{n-1}^{\mathrm{a}} \oplus O_0^{\mathrm{f}}/M_0^{\mathrm{f}} \xrightarrow[\text{CNN}]{3\times 3} O_n^{\mathrm{f}}/M_n^{\mathrm{f}}, \tag{4}$$

$$(F_{t+i}, O_n^{\mathrm{f}}, M_n^{\mathrm{f}}) \xrightarrow[\text{DCN}]{\text{dilation rate} = d} F_n^{\mathrm{a}}, \tag{5}$$

where n stands for the branch number and d stands for the dilation rate, $d \in \{3, 1\}$. Finally, the module connects F_3^{a} and feature F_t, and convolves them to obtain $O_4^{\mathrm{f}}/M_4^{\mathrm{f}}$. We feed F_t, O_4^{f}, and M_4^{f} to the DCN with a dilation rate of 1 and obtain the final aligned feature F_{t+1}^{a}. The whole process can be expressed as

$$F_t \oplus F_3^{\mathrm{a}} \xrightarrow[\text{CNN}]{3\times 3} O_4^{\mathrm{f}}/M_4^{\mathrm{f}}, \tag{6}$$

$$(F_{t+i}, O_4^{\mathrm{f}}, M_4^{\mathrm{f}}) \xrightarrow[\text{DCN}]{\text{dilation rate} = 1} F_{t+i}^{\mathrm{a}}. \tag{7}$$

The general deformable convolution has a dilation rate of 1. Although it has a learnable offset, its receptive field is not sufficient for feature fusion. The general method uses a pyramid structure (Wang XT et al., 2019), performs multiple downsampling and then upsampling, and fuses the current feature with the upsampled feature each time. The features extracted by the backbone network are generated after multiscale fusion. After feature downsampling, subsequent upsampling is generally done by bilinear sampling, resulting in a loss of the generated features compared to the original ones.

We use multiple deformable convolutions with different dilation rates in parallel to achieve feature alignment, avoiding feature loss caused by downsampling. The small dilation rate aggregates the local subtle features, the large dilation rate (Yu and Koltun, 2016) aggregates the overall feature information, and receptive fields of different sizes extract the multi-scale feature information. In the last branch, the module finally connects feature F_3^{a} with key feature F_t, which strengthens the connection between aligned features and key features and provides enough gradients for network training.

3.1.2 Fusion with spatial attention

For various reasons, such as motion blur, occlusion, and perspective transformation, there is no guarantee that auxiliary features can help improve the current features. If wrong auxiliary features are fused, the quality of the current features and output accuracy will be reduced. Therefore, it is necessary to judge whether auxiliary features can help improve the refinement. Specifically, the above problem is caused by the inconsistent distribution of effective information in consecutive features.

To reduce the impact of the above problems, we design an attention fusion module to fuse features of auxiliary frames. The structure of this module is shown in Fig. 2. Auxiliary features that are similar to current features should be given higher weights to increase their attention. Therefore, we first compute the similarity coefficients between the current feature and auxiliary features. Our attention calculation method is based on Wang XL et al. (2018)’s method. The similarity weight can be calculated as follows:

$$(f_{\theta 1}(F_{t+i}))^{\mathrm{T}}\cdot f_{\theta 2}(F_t)\xrightarrow{\text{Sigmoid}}S(F_{t+i},F_t),\tag{8}$$

where F_{t+i} and F_t represent the auxiliary feature and key feature, respectively. $f_{\theta 1}(F_{t+i})$ and $f_{\theta 2}(F_t)$ are two embedded features, which can be obtained by conventional convolution. The sigmoid function restricts the similarity coefficients to [0, 1]. Specifically, $S(F_{t+i}, F_t)$ has the same size as F_{t+i}.

Then, the calculated similarity coefficients $S(F_{t+i}, F_t)$ are used to weight the auxiliary features F_{t+i}. The specific operation is shown in the following formula:

$$F_{t+i}\odot S(F_{t+i},F_t)\xrightarrow[\text{CNN}]{3\times 3}F'_{t+i},\tag{9}$$

where $\odot$ is the element-wise multiplication. Finally, the aligned auxiliary features F'_{t-i}, F'_{t+i} and current feature F_t are concatenated and inputted into the convolutional layer for fusion to generate the fused current feature F_t^{final}. F_t^{final} is inputted to the 1×1 convolutional layer to generate the refined heatmaps H_t^{r}. The operation is shown in the following formula:

$$F'_{t+1}\oplus F_t\oplus F'_{t-1}\xrightarrow[\text{CNN}]{3\times 3}F_t^{\mathrm{final}}\xrightarrow[\text{CNN}]{1\times 1}H_t^{\mathrm{r}}.\tag{10}$$

This module computes the similarity coefficients between the current feature and aligned auxiliary features by spatial attention. The ability to perceive the effective information of the auxiliary features after alignment is enhanced, and the quality of the current feature after fusion is improved.

3.2 Semantics refinement module

The motivation for designing the semantics refinement module is as follows: the adjacent rough heatmaps contain obvious human body key point position information, the difference between adjacent heatmaps reflects the amount of human motion within two frames, and the difference information can effectively supplement the key point positioning error in the current frame due to motion blur and other reasons. At the same time, refining the prediction results at the feature level alone has limited effect.

The current method uses mostly parallel multi-expansion rate deformable convolution at the semantics level, which requires a large memory space during training. It merges multi-branch features using average weighted summation, which does not make good use of the revised heatmaps at multiple expansion rates. Our experiments showed that multi-branch features lead to limited improvement in network refinement performance (see Section 4.4 for details). We design the following semantics refinement module, which reduces the resources required for training while preserving performance. Ablation experiments demonstrated the effectiveness of our module design. The structure of this module is as shown in Fig. 2.

Specifically, we use the refined heatmaps H_t^{r} to subtract the auxiliary heatmaps $[H_{t-1}, H_{t+1}]$ and input the results into a 3×3 convolutional block (Cai et al., 2020) to generate difference information $D_{t,t-1}$ and $D_{t,t+1}$. This module connects features H_t^{r}, $D_{t,t-1}$, and $D_{t,t+1}$, and then inputs the 3×3 convolutional block to generate the aggregated differential information D_{a}:

$$H_t^{\mathrm{r}} - H_{t-1} \xrightarrow[\text{CNN}]{3\times3} D_{t,t-1}, \tag{11}$$

$$H_t^{\mathrm{r}} - H_{t+1} \xrightarrow[\text{CNN}]{3\times3} D_{t,t+1}, \tag{12}$$

$$D_{t,t-1} \oplus H_t^{\mathrm{r}} \oplus D_{t,t+1} \xrightarrow[\text{CNN}]{3\times3} D_{\mathrm{a}}. \tag{13}$$

At the same time, H_{t-1}, H_{t+1}, and H_t^{r} are connected, followed by a convolutional layer that outputs aggregated heatmap H_{a}:

$$H_{t-1} \oplus H_t^{\mathrm{r}} \oplus H_{t+1} \xrightarrow[\text{CNN}]{3\times3} H_{\mathrm{a}}. \tag{14}$$

After that, we connect H_{a} with D_{a}, and feed them into the convolutional layer to generate an intermediate feature mask/offset $O^{\mathrm{s}}/M^{\mathrm{s}}$:

$$D_{\mathrm{a}} \oplus H_{\mathrm{a}} \xrightarrow[\text{CNN}]{3\times3} O^{\mathrm{s}}/M^{\mathrm{s}}. \tag{15}$$

Finally, the aggregated heatmap H_{a}, offset O^{s}, and mask M^{s} are fed into deformable convolutions to generate semantics-level refined heatmap H_t^{final}:

$$(H_{\mathrm{a}}, O^{\mathrm{s}}, M^{\mathrm{s}}) \xrightarrow[\text{CNN}]{3\times3} \xrightarrow[\text{DCN}]{\text{dilation rate}=1} H_t^{\mathrm{final}}. \tag{16}$$

Since the size of the valid information in each generated heatmap is about 10–15 pixels, only a small receptive field is required to implement semantics-level refinement. In this module, a deformable convolution with a kernel size of 3 and a dilation rate of 1 is used. The effectiveness of this module is examined in the experiments in Section 4.

3.3 Loss function

We compute the L2 distance between the predicted heatmaps and the ground-truth heatmaps as the loss function. The loss is applied to predict both levels using the same ground truth.

During the experiments, we found that the performance improvement of the network is limited only if the final outputs are used to calculate the loss. This is because the network refines the results at both feature and semantics levels. Computing the loss using only the output of the final layer means that only semantics-level refinement is supervised. There is no effective optimization object for feature correction, resulting in poor refinement results. The final loss is the sum of the two losses. The loss function is defined as

$$L_1 = \frac{1}{J}\sum_{j=1}^{J} V_j l_2(H_j^{\mathrm{F}}, \hat{H}_j), \tag{17}$$

$$L_2 = \frac{1}{J}\sum_{j=1}^{J} V_j l_2(H_j^{\mathrm{S}}, \hat{H}_j), \tag{18}$$

$$\text{Loss} = \alpha L_1 + (1-\alpha) L_2, \tag{19}$$

where H_j^{F} represents the heatmaps generated by 1×1 convolution after feature-level refinement, H_j^{S} represents the heatmaps generated after semantics-level refinement, $\hat{H}_j$ represents the ground-truth heatmaps, j is the key point number, V_j visualizes the key points in the label, and α is the weight coefficient of L_1, set to 0.4 in this method.

4 Experiments

4.1 Experimental settings

PoseTrack contains two large-scale public datasets for human pose estimation and joint tracking in unconstrained videos. The PoseTrack2017 dataset (Iqbal et al., 2017) contains 514 video clips and 16 219 pose annotations; we used 250 clips for training, 50 clips for validation, and 214 clips for testing. The PoseTrack2018 dataset (Andriluka et al., 2018) has 1138 video clips; we used 593 for training, 170 for validation, and 375 for testing. Training videos were densely annotated within the central 30 frames of each video clip. For the validation video, annotations were provided every four frames for the entire video segment, in addition to the dense annotation of the central 30 frames. Both PoseTrack2017 and PoseTrack2018 annotate 15 joints, two-dimensional (2D) coordinates of human joint points in the image, and an additional label annotating joint visibility. We trained and evaluated only visible joints.

We trained the network independently on PoseTrack2017 and PoseTrack2018 using the same configuration. During training, we incorporated data augmentation, including random rotation [−45°, 45°], random scaling [0.65, 1.35], truncation, and a horizontal flip probability of 0.5. The input image size was fixed to 384×288. The default radius D for generating Gaussian heatmaps was set to 2. Hyperparameter α was set to 0.4. We used HRNet-W48 pretrained on the COCO dataset (Lin et al., 2014) as the backbone network and fixed its weights in subsequent training. All subsequent weight parameters were randomly initialized from a Gaussian distribution with μ=0 and σ=0.001, while the bias was always initialized to 0. The Adam optimizer was used during training. Its basic learning rate was le-4, which decayed to 1/10 at the 8th, 16th, 20th, and 25th generations. We used an Nvidia GeForce 3090 GPU to train our model for 30 epochs with a training batch size of 64 each.

Following the evaluation methods of Andriluka et al. (2018), we evaluated the performance of the human pose estimation network by computing the average precision (AP). This metric was calculated independently for each joint and finally divided by the number of joints to obtain the final mean AP (mAP).

4.2 Comparison with existing methods

We first show the evaluation results for the validation set and test set of the PoseTrack2017 dataset. The evaluation metric was AP. Table 1 shows the results of a quantitative comparison of our method with state-of-the-art methods (Doering et al., 2018; Girdhar et al., 2018; Xiao et al., 2018; Xiu et al., 2018; Bertasius et al., 2019; Guo et al., 2019; Hwang et al., 2019; Jin et al., 2019; Sun et al., 2019; Zhang et al., 2019; Liu et al., 2021; Yang YD et al., 2021; Fang HS et al., 2023) on the PoseTrack2017 dataset. The comparison includes the AP of each keypoint and the mAP of all keypoints. Our method achieved an mAP of 83.0 on the validation set, outperforming state-of-the-art methods. In addition, Table 2 shows the comparison results on the test set (Doering et al., 2018; Girdhar et al., 2018; Xiao et al., 2018; Xiu et al., 2018; Bertasius et al., 2019; Hwang et al., 2019; Sun et al., 2019; Snower et al., 2020; Liu et al., 2021). Fig. 4 shows the visualization results of our method on some challenging scenes in the PoseTrack2017 dataset.

Our model was also evaluated on the validation and test sets in the PoseTrack2018 dataset. Table 3 shows the results from a comparison of our method with state-of-the-art methods on the validation set, and Table 4 shows the results on the test set (Bertasius et al., 2019; Guo et al., 2019; Hwang et al., 2019; Sun et al., 2019; Wang MC et al., 2020; Fang HS et al., 2023). Fig. 5 shows the visualization results of our method on some challenging scenes in the PoseTrack2018 dataset.

Due to the algorithm platform and incomplete open source, it is difficult to directly compare the time cost of each model. We compared the number of model parameters and computation complexity to estimate the time cost. Table 5 presents the parameter number and computation complexity of our approach and those of the representative top competitors, such as HRNet (Sun et al., 2019), PoseWarper (Bertasius et al., 2019), and DCpose (Liu et al., 2021).

Table 1 Quantitative results of our method and state-of-the-art methods on the PoseTrack2017 validation set

Method	Year	AP							mAP
		Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	
PoseTracker	2018	67.5	70.2	62.0	51.7	60.7	58.7	49.8	60.6
PoseFlow	2018	66.7	73.3	68.3	61.1	67.5	67.0	61.3	66.5
JointFlow	2018	–	–	–	–	–	–	–	69.3
SimpleBaseline	2018	81.7	83.4	80.0	72.4	75.3	74.8	67.1	76.7
TML++	2019	–	–	–	–	–	–	–	71.5
FastPose	2019	80.0	80.3	69.5	59.1	71.4	67.5	59.4	70.3
STEmbedding	2019	83.8	81.6	77.1	70.0	77.4	74.5	70.8	77.0
HRNet	2019	82.1	83.6	80.4	73.3	75.5	75.3	68.5	77.3
MDPN	2019	85.2	88.5	83.9	77.5	79.0	77.0	71.4	80.7
PoseWarper	2019	81.4	88.3	83.9	78.0	82.4	80.5	73.6	81.2
Dynamic-GNN	2021	**88.4**	88.4	82.0	74.5	79.1	78.3	73.1	81.1
DCpose	2021	88.0	88.7	84.1	**78.4**	83.0	**81.4**	**74.2**	82.8
AlphaPose	2023	–	–	–	–	–	–	–	76.9
Ours	2022	88.1	**88.8**	**84.2**	**78.4**	**83.1**	**81.4**	**74.2**	**83.0**

The bold font denotes the best result

Table 2 Quantitative results of our method and state-of-the-art methods on the PoseTrack2017 test set

Method	Year	AP							mAP
		Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	
PoseTracker	2018	–	–	–	51.5	–	–	50.2	59.6
PoseFlow	2018	64.9	67.5	65.0	59.0	62.5	62.8	57.9	63.0
JointFlow	2018	–	–	–	53.1	–	–	50.4	63.4
SimpleBaseline	2018	80.1	80.2	76.9	71.5	72.5	72.4	65.7	74.6
TML++	2019	–	–	–	60.9	–	–	–	67.8
HRNet	2019	80.1	80.2	76.9	72.0	73.4	72.5	67.0	74.9
PoseWarper	2019	79.5	84.3	80.1	75.8	77.6	76.8	70.8	77.9
KeyTrack	2020	–	–	–	71.9	–	–	65.0	74.0
DCpose	2021	84.3	**84.9**	80.5	76.1	77.9	77.1	**71.2**	79.2
Ours	2022	**84.6**	84.8	**80.6**	**76.2**	**78.0**	**77.2**	71.0	**79.3**

The bold font denotes the best result

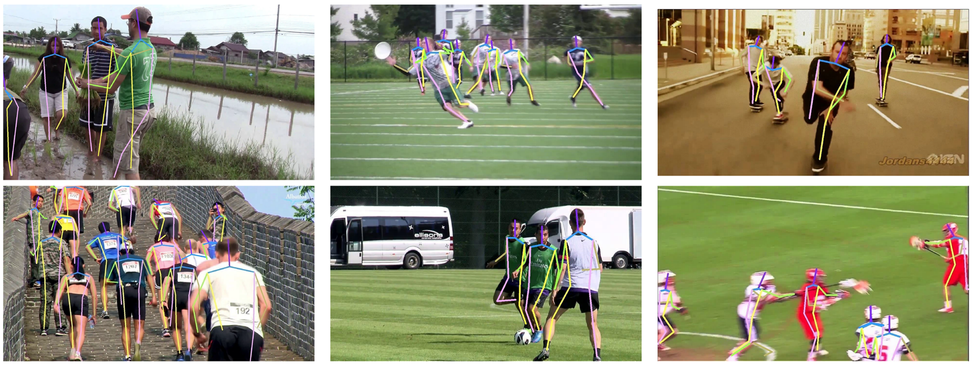

Fig. 4 Visualization results of some challenging scenarios in the PoseTrack2017 dataset. Scenes include motion blur, occlusion, and multiple persons

Table 3 Quantitative results of our method and state-of-the-art methods on the PoseTrack2018 validation set

Method	Year	AP							mAP
		Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	
TML++	2019	–	–	–	–	–	–	–	74.6
MDPN	2019	75.4	81.2	79.0	74.1	72.4	73.0	69.9	75.0
HRNet	2019	80.1	80.2	76.9	72.0	73.4	72.5	67.0	74.9
PoseWarper	2019	79.5	84.3	80.1	75.8	77.6	76.8	70.8	77.9
AlphaPose	2023	–	–	–	–	–	–	–	74.7
Ours	2022	**83.8**	**96.9**	**82.7**	**77.6**	**80.3**	**79.5**	**74.0**	**80.9**

The bold font denotes the best result

Table 4 Quantitative results of our method and state-of-the-art methods on the PoseTrack2018 test set

Method	Year	AP							mAP
		Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	
AlphaPose++	2019	–	–	–	66.2	–	–	65.0	67.6
TML++	2019	–	–	–	–	–	–	–	67.8
MDPN	2019	–	–	–	74.5	–	–	69.0	76.4
PoseWarper	2019	78.9	**84.4**	**80.9**	**76.8**	75.6	**77.5**	71.8	78.0
DetTrack	2020	–	–	–	69.8	–	–	67.1	73.5
Ours	2022	**83.3**	84.2	80.8	**76.8**	**75.7**	77.4	**72.0**	**78.9**

The bold font denotes the best result

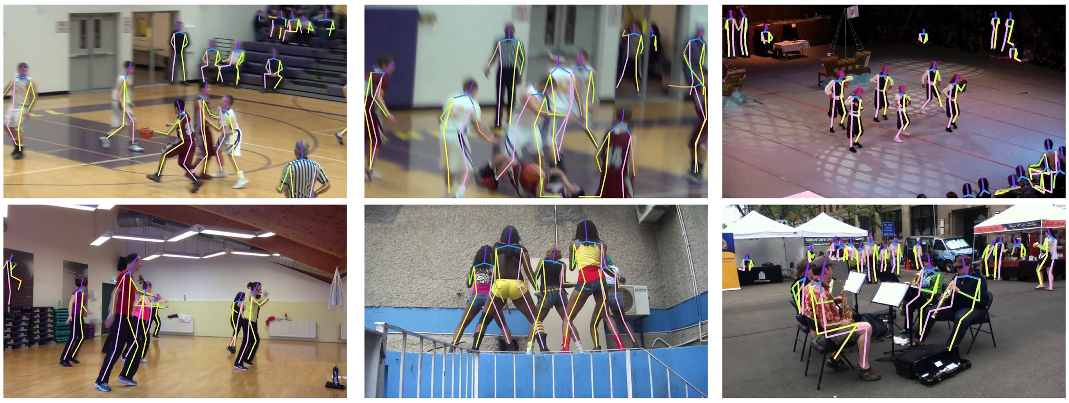

Fig. 5 Visualization results of some challenging scenarios in the PoseTrack2018 dataset. Scenes include motion blur, occlusion, and multiple persons

Table 5 Comparison of GFLOPs and the number of parameters among state-of-the-art methods and our method

Method	Input size	Parameter number	GFLOPs
HRNet	3×384×288	6.36×10^7	35.54
PoseWarper	15×384×288	7.114×10^7	438.58
DCpose	9×384×288	6.519×10^7	117.61
Ours	9×384×288	6.523×10^7	117.66

HRNet serves as the backbone network for DCpose, PoseWarper, and our method. Table 5 demonstrates that our method is similar to DCpose but better than PoseWarper in terms of the parameter number and giga floating-point operations per second (GFLOPs).

4.3 Visual comparison with existing methods

To visually demonstrate the performance of our method, the visualization results of our method were compared to those of existing methods under challenging scenarios in PoseTrack2017 and PoseTrack2018 (Fig. 6). The scenes were defocus, occlusions, nearby persons, and rapid motion. The results of our method were better than those of PoseWarper (Bertasius et al., 2019) and HRNet (Sun et al., 2019). HRNet is based on a single-frame image, and the performance is degraded when the image quality is poor. PoseWarper improves keyframe results only at the semantics level, without considering feature-level refinements. Our method uses adjacent frame features and heatmaps as auxiliary information, and refines the keyframe results at the feature and semantics levels simultaneously, resulting in more accurate outputs.

4.4 Ablation study

To reduce the training time, we conducted ablation experiments on the smaller PoseTrack2017 dataset to verify the effectiveness of each module in our method. The modules for ablation include the feature alignment module, attention fusion module, and semantics refinement module. Our experimental results showed that the designed modules are effective in improving output accuracy.

Feature-level refinements: In this setting, we explored the impact of feature alignment on network performance. Three settings were adopted in the experiment: removing the feature alignment module, feature alignment with a pyramid structure, and feature alignment with multiple dilation rates.

Table 6 shows the following results: (1) When adding our feature-level refinement, mAP increased from 80.0330 to 82.4236. (2) The mAP was 82.2564 when with a pyramid structure. The results proved our idea that aligned auxiliary features can effectively improve the current features. The downsampling and upsampling structures in the pyramid structure will lose part of the feature information, resulting in inaccurate alignment and decreasing mAP. The parallel multi-dilation rate structure increased the receptive field of feature alignment, enabling the alignment module to adapt to the size of different features, thereby improving the efficiency of the alignment module.

Attention fusion module: In this ablation setting, we explored the effect of attention mechanism on feature fusion. We removed the attention mechanism in the fusion module and used only convolution for feature

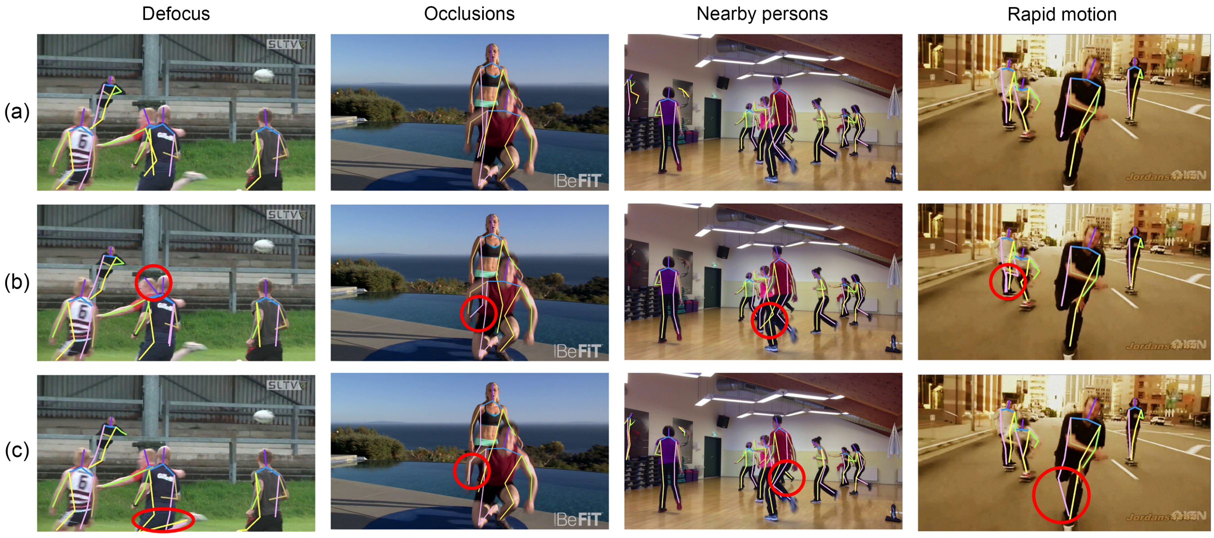

Fig. 6 Visual results of the pose predictions of our method (a), PoseWarper (b), and HRNet (c) in challenging situations from the PoseTrack2017 and PoseTrack2018 datasets. Scenes include defocus, occlusions, nearby persons, and rapid motion. Inaccurate results are marked with red circles. References to color refer to the online version of this figure

Table 6 Ablation study at the feature level on the PoseTrack2017 validation set

Method	AP							mAP
	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	
FaSRnet	**88.1060**	**88.7576**	**84.2307**	**78.3686**	**83.0849**	**81.4284**	**74.2305**	**82.9680**
FaSRnet (Baseline)	87.3318	88.1351	83.3196	77.1669	82.1129	80.2089	73.3062	80.0330
FaSRnet (PCDA)	87.4504	88.3379	83.4284	77.3575	82.6070	80.5165	73.4997	82.2564
FaSRnet (FDA)	87.5150	88.4070	83.6207	77.6307	82.8760	80.5899	73.7806	82.4236
FaSRnet w/o Att	87.4220	88.2743	83.4160	77.2839	82.4650	80.1781	73.2446	82.1327

PCDA: feature alignment module with a pyramid structure; FDA: our feature alignment module; Att: our attention fusion module. The bold font denotes the best result

fusion. The resulting mAP decreased from 82.4236 to 82.1327, which indicates that the auxiliary features after feature alignment have a negative impact on the refinement. It is necessary to introduce an attention mechanism to reduce this negative effect.

Semantics-level refinement: In this ablation setting, we explored mainly the effects of semantics-level refinement and multiple receptive fields (Bertasius et al., 2019; Liu et al., 2021) on the output. When the network added semantics-level refinement based on feature-level modules, its mAP increased from 82.4236 to 82.9680. This result showed that refinements at the semantics level can improve network performance. Since Liu et al. (2021) used a deformable convolution module with parallel multi-dilation rates for refinement, the effect of different dilation rates on network performance was also explored in ablation.

Table 7 shows that the mAP was 82.9440 with multi-dilation rates, and 82.9680 with a single dilation rate. Multi-dilation rates had little effect on semantics-level refinement. The subtle effects of multi-dilation rates on performance indicated that the effective information in the heatmaps was concentrated in local areas. When generating label heatmaps, their effective area was limited to about a dozen of pixels. Therefore, adding parallel multi-dilation rate branches has little effect on network performance. However, parallel multi-dilation rate convolutions require a lot of memory in a graphics processing unit (GPU) during training, so in our network, deformable convolutions with a dilation rate of 1 were used for refinement.

Auxiliary frame effects: We also explored the impact of using a single auxiliary frame on the network. When a certain auxiliary frame was removed,

Table 7 Ablation study at the semantics level on the PoseTrack2017 validation set

Method	AP							mAP
	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	
FaSRnet	**88.1060**	88.7576	**84.2307**	**78.3686**	83.0849	81.4284	**74.2305**	**82.9680**
FaSRnet w/o FR	87.6666	88.5231	84.0318	78.2326	82.9212	81.1459	74.0071	82.7149
FaSRnet (SMD)	88.0695	**88.8296**	84.1985	78.2468	**83.1996**	**81.4349**	74.0665	82.9440

FR: feature refinement module; SMD: using semantics with multiple dilation rates, with the dilation rate *d* being 3, 6, 9, 12, or 15. The bold font denotes the best result

Table 8 Ablation study of the auxiliary frame on the PoseTrack2017 validation set

Method	AP							mAP
	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	
FaSRnet	**88.1060**	**88.7576**	**84.2307**	**78.3686**	**83.0849**	**81.4284**	**74.2305**	**82.9680**
FaSRnet w/o N	87.8460	88.5093	83.7763	77.8341	82.8088	80.6752	73.6541	82.5369
FaSRnet w/o P	87.9876	88.4478	83.7902	77.9312	82.5358	80.8370	73.5315	82.5407

N and P stand for the next frame and previous frame, respectively. The bold font denotes the best result

the network performance dropped significantly (from 82.9680 to 82.5407 and 82.5369) (Table 8). This was expected as time information plays an important role in the refinement, and each auxiliary frame can provide useful information for the refinement of the current frames.

Loss hyperparameter α: We explored the effect of hyperparameter α in the loss function. The value of α ranged from 0 to 0.9 with a step of 0.1. The network performed the best when the value was 0.4 (Table 9). The results indicated that feature-level correction results are worth using. The final output combines the two levels of refinement results and needs to be given larger weights, which is in line with our expectations.

Table 9 Ablation study of the loss function hyperparameter on the PoseTrack2017 validation set

α	Best mean AP	α	Best mean AP
0	82.9188	0.5	82.9645
0.1	82.9507	0.6	82.9601
0.2	82.9445	0.7	82.9619
0.3	82.9669	0.8	82.9518
0.4	**82.9997**	0.9	82.9332

The bold font denotes the best result

Training stability: We verified the stability of the training of our method. We trained and tested 10 times on PoseTrack2017. The results indicated that our method has good training stability (Table 10).

Table 10 Training stability of our method

Parameter	Value
Number of training times	10
Maximum mAP	82.9997
Average mAP	82.9759
Variance	0.000 254 571

5 Conclusions

We propose a video-based human pose estimation model. The method refines the current frame at feature and semantics levels. A multi-receptive field feature refinement module is designed to refine the predicted pose. Our semantics correction module uses the difference information between heatmaps to further refine the predicted pose. Our method has been validated on large-scale benchmark datasets PoseTrack2017 and PoseTrack2018, outperforming most existing methods. The image-based pose estimation method is used as a backbone to estimate multi-person poses in video, making it intuitive and easy to understand. However, our semantics correction module uses the heatmaps generated by the backbone network directly as auxiliary features. Subsequent work could explore the use of refined heatmaps to further improve the performance.

Contributors

Yuanhong ZHONG designed the research. Qianfeng XU and Daidi ZHONG processed the data. Yuanhong ZHONG, Qianfeng XU, and Daidi ZHONG drafted the paper. Xun YANG and Shanshan WANG helped organize the paper. All the authors revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The code is available at https://github.com/Elvis-Aron/FaSRnet. The other data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Andriluka M, Pishchulin L, Gehler P, et al., 2014. 2D human pose estimation: new benchmark and state of the art analysis. IEEE Conf on Computer Vision and Pattern Recognition, p.3686-3693. https://doi.org/10.1109/CVPR.2014.471

Andriluka M, Iqbal U, Insafutdinov E, et al., 2018. PoseTrack: a benchmark for human pose estimation and tracking. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.5167-5176. https://doi.org/10.1109/CVPR.2018.00542

Bertasius G, Feichtenhofer C, Tran D, et al., 2019. Learning temporal pose estimation from sparsely-labeled videos. Proc 33rd Int Conf on Neural Information Processing Systems, p.3027-3038.

Cai YH, Wang ZC, Luo ZX, et al., 2020. Learning delicate local representations for multi-person pose estimation. 16th European Conf on Computer Vision, p.455-472. https://doi.org/10.1007/978-3-030-58580-8_27

Cao Z, Hidalgo G, Simon T, et al., 2021. OpenPose: realtime multi-person 2D pose estimation using part affinity fields. *IEEE Trans Patt Anal Mach Intell*, 43(1):172-186. https://doi.org/10.1109/TPAMI.2019.2929257

Chu X, Yang W, Ouyang WL, et al., 2017. Multi-context attention for human pose estimation. IEEE Conf on Computer Vision and Pattern Recognition, p.5669-5678. https://doi.org/10.1109/CVPR.2017.601

Dang YH, Yin JQ, Zhang SJ, et al., 2022a. Learning human kinematics by modeling temporal correlations between joints for video-based human pose estimation. https://doi.org/10.48550/arXiv.2207.10971

Dang YH, Yin JQ, Zhang SJ, 2022b. Relation-based associative joint location for human pose estimation in videos. *IEEE Trans Image Process*, 31:3973-3986. https://doi.org/10.1109/TIP.2022.3177959

Doering A, Iqbal U, Gall J, 2018. Joint flow: temporal flow fields for multi person tracking. https://doi.org/10.48550/arXiv.1805.04596

Fang HS, Xie SQ, Tai YW, et al., 2017. RMPE: regional multi-person pose estimation. IEEE Int Conf on Computer Vision, p.2353-2362. https://doi.org/10.1109/ICCV.2017.256

Fang HS, Li JF, Tang HY, et al., 2023. AlphaPose: whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Trans Patt Anal Mach Intell*, 45(6):7157-7173. https://doi.org/10.1109/TPAMI.2022.3222784

Fang ZJ, López AM, 2020. Intention recognition of pedestrians and cyclists by 2D pose estimation. *IEEE Trans Intell Transp Syst*, 21(11):4773-4783. https://doi.org/10.1109/TITS.2019.2946642

Girdhar R, Gkioxari G, Torresani L, et al., 2018. Detect-and-track: efficient pose estimation in videos. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.350-359. https://doi.org/10.1109/CVPR.2018.00044

Guo HK, Tang T, Luo GZ, et al., 2019. Multi-domain pose network for multi-person pose estimation and tracking. European Conf on Computer Vision, p.209-216. https://doi.org/10.1007/978-3-030-11012-3_17

Hwang J, Lee J, Park S, et al., 2019. Pose estimator and tracker using temporal flow maps for limbs. Int Joint Conf on Neural Networks, p.1-8. https://doi.org/10.1109/IJCNN.2019.8851734

Insafutdinov E, Andriluka M, Pishchulin L, et al., 2017. ArtTrack: articulated multi-person tracking in the wild. Conf on Computer Vision and Pattern Recognition, p.1293-1301. https://doi.org/10.1109/CVPR.2017.142

Iqbal U, Milan A, Gall J, 2017. PoseTrack: joint multi-person pose estimation and tracking. IEEE Conf on Computer Vision and Pattern Recognition, p.4654-4663. https://doi.org/10.1109/CVPR.2017.495

Jin S, Liu WT, Ouyang WL, et al., 2019. Multi-person articulated tracking with spatial and temporal embeddings. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.5657-5666. https://doi.org/10.1109/CVPR.2019.00581

Jin S, Liu WT, Xie EZ, et al., 2020. Differentiable hierarchical graph grouping for multi-person pose estimation. 16th European Conf on Computer Vision, p.718-734. https://doi.org/10.1007/978-3-030-58571-6_42

Li DW, Chen XT, Zhang Z, et al., 2018. Pose guided deep model for pedestrian attribute recognition in surveillance scenarios. IEEE Int Conf on Multimedia and Expo, p.1-6. https://doi.org/10.1109/ICME.2018.8486604

Lin TY, Maire M, Belongie S, et al., 2014. Microsoft COCO: common objects in context. 13th European Conf on Computer Vision, p.740-755. https://doi.org/10.1007/978-3-319-10602-1_48

Liu ZG, Wu S, Jin SY, et al., 2019. Towards natural and accurate future motion prediction of humans and animals. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9996-10004. https://doi.org/10.1109/CVPR.2019.01024

Liu ZG, Chen HM, Feng RY, et al., 2021. Deep dual consecutive network for human pose estimation. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.525-534. https://doi.org/10.1109/CVPR46437.2021.00059

Liu ZG, Feng RY, Chen HM, et al., 2022. Temporal feature alignment and mutual information maximization for video-based human pose estimation. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.10996-11006. https://doi.org/10.1109/CVPR52688.2022.01073

Luo Y, Ren J, Wang ZX, et al., 2018. LSTM pose machines. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.5207-5215. https://doi.org/10.1109/CVPR.2018.00546

Martinez J, Hossain R, Romero J, et al., 2017. A simple yet effective baseline for 3D human pose estimation. IEEE Int Conf on Computer Vision, p.2659-2668. https://doi.org/10.1109/ICCV.2017.288

Pfister T, Charles J, Zisserman A, 2015. Flowing ConvNets for human pose estimation in videos. IEEE Int Conf on Computer Vision, p.1913-1921. https://doi.org/10.1109/ICCV.2015.222

Sapp B, Taskar B, 2013. MODEC: multimodal decomposable models for human pose estimation. IEEE Conf on Computer Vision and Pattern Recognition, p.3674-3681. https://doi.org/10.1109/CVPR.2013.471

Shao ZP, Zhou W, Wang WZ, et al., 2023. A temporal densely connected recurrent network for event-based human pose estimation. https://doi.org/10.48550/arXiv.2209.07034

Snower M, Kadav A, Lai F, et al., 2020. 15 keypoints is all you need. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.6737-6747. https://doi.org/10.1109/CVPR42600.2020.00677

Song J, Wang LM, van Gool L, et al., 2017. Thin-slicing network: a deep structured model for pose estimation in videos. IEEE Conf on Computer Vision and Pattern Recognition, p.5563-5572. https://doi.org/10.1109/CVPR.2017.590

Sun K, Xiao B, Liu D, et al., 2019. Deep high-resolution representation learning for human pose estimation. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.5686-5696. https://doi.org/10.1109/CVPR.2019.00584

Tian YP, Zhang YL, Fu Y, et al., 2020. TDAN: temporally-deformable alignment network for video super-resolution. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.3357-3366. https://doi.org/10.1109/CVPR42600.2020.00342

Wang J, Long X, Gao Y, et al., 2020. Graph-PCNN: two stage human pose estimation with graph pose refinement. 16th European Conf on Computer Vision, p.492-508. https://doi.org/10.1007/978-3-030-58621-8_29

Wang M, Hong RC, Yuan XT, et al., 2012. Movie2Comics: towards a lively video content presentation. *IEEE Trans Multim*, 14(3):858-870. https://doi.org/10.1109/TMM.2012.2187181

Wang MC, Tighe J, Modolo D, 2020. Combining detection and tracking for human pose estimation in videos. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.11085-11093. https://doi.org/10.1109/CVPR42600.2020.01110

Wang XL, Girshick R, Gupta A, et al., 2018. Non-local neural networks. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.7794-7803. https://doi.org/10.1109/CVPR.2018.00813

Wang XT, Chan KCK, Yu K, et al., 2019. EDVR: video restoration with enhanced deformable convolutional networks. IEEE/CVF Conf on Computer Vision and Pattern Recognition Workshops, p.1954-1963. https://doi.org/10.1109/CVPRW.2019.00247

Weinzaepfel P, Revaud J, Harchaoui Z, et al., 2013. DeepFlow: large displacement optical flow with deep matching. IEEE Int Conf on Computer Vision, p.1385-1392. https://doi.org/10.1109/ICCV.2013.175

Xiao B, Wu HP, Wei YC, 2018. Simple baselines for human pose estimation and tracking. 15th European Conf on Computer Vision, p.472-487. https://doi.org/10.1007/978-3-030-01231-1_29

Xiu YL, Li JF, Wang HY, et al., 2018. Pose flow: efficient online pose tracking. https://doi.org/10.48550/arXiv.1802.00977

Yang X, Wang M, Hong RC, et al., 2017. Enhancing person re-identification in a self-trained subspace. *ACM Trans Multim Comput Commun Appl*, 13(3):27. https://doi.org/10.1145/3089249

Yang X, Wang M, Tao DC, 2018. Person re-identification with metric learning using privileged information. *IEEE Trans Image Process*, 27(2):791-805. https://doi.org/10.1109/TIP.2017.2765836

Yang YD, Ren Z, Li HX, et al., 2021. Learning dynamics via graph neural networks for human pose estimation and tracking. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.8070-8080. https://doi.org/10.1109/CVPR46437.2021.00798

Yu F, Koltun V, 2016. Multi-scale context aggregation by dilated convolutions. https://doi.org/10.48550/arXiv.1511.07122

Zhang JB, Zhu Z, Zou W, et al., 2019. FastPose: towards real-time pose estimation and tracking via scale-normalized multi-task networks. https://doi.org/10.48550/arXiv.1908.05593

Zheng W, Li L, Zhang ZX, et al., 2019. Relational network for skeleton-based action recognition. IEEE Int Conf on Multimedia and Expo, p.826-831. https://doi.org/10.1109/ICME.2019.00147

Zhu XZ, Hu H, Lin S, et al., 2019. Deformable ConvNets V2: more deformable, better results. IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9300-9308. https://doi.org/10.1109/CVPR.2019.00953