Pattern Recognition 103 (2020) 107258

Contents lists available at ScienceDirect

Pattern Recognition

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

Exploring temporal consistency for human pose estimation in videos

Yang Li[a,b], Kan Li[a,*], Xinxin Wang[a], Richard Yi Da Xu[b]

[a] School of Computer Science, Beijing Institute of Technology, 5 South Zhongguancun Street, Beijing, 100081, China
[b] Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Ultimo, NSW 2007, Australia

ARTICLE INFO

Article history:
Received 30 May 2019
Revised 28 January 2020
Accepted 1 February 2020
Available online 8 February 2020

Keywords:
Video-based pose estimation
Convolution neural network
Temporal information

ABSTRACT

In this paper, we introduce a method of exploring temporal information for estimating human poses in videos. The current state-of-the-art methods utilizing temporal information can be categorized into two major branches. The first category is a model-based method that captures the temporal information entirely by using a learnable function such as RNN or 3D convolution. However, these methods are limited in exploring temporal consistency, which is essential for estimating human joint positions in videos. The second category is the posterior enhancement method, where an independent post-processing step (e.g., using optical flow) is applied to enhance the prediction. However, operations such as optical flow estimation can be susceptible to the occlusion and motion blur problems, which will adversely affect the final performance. We propose a novel Temporal Consistency Exploration (TCE) module to address both shortcomings. Compared to previous approaches, the TCE module is more efficient as it captures the temporal consistency at the feature level without having to post-process and calculate extra optical flow. Further, to capture the rich spatial context in video data, we design a multi-scale TCE to explore the time consistency information at multi-scale spatial levels. Finally, a video-based pose estimation network is designed, which is based on the encoder-decoder architecture and extended with the powerful multi-scale TCE module. We comprehensively evaluate the proposed model on two video datasets, Sub-JHMDB and Penn, and our model achieves state-of-the-art performance on both datasets.

1. Introduction

Human pose estimation is a fundamental task in the computer vision community and has been broadly applied to many fields such as human activity recognition [1], sports analysis [2] and human-computer interaction [3]. The purpose of pose estimation is to locate the anatomical keypoints in human bodies. Previous methods have traditionally relied on hand-crafted features. They face challenges when handling unconstrained cases due to the highly articulated human body limbs, occlusion and change of viewpoint. Recently, as a result of the availability of large-scale human pose datasets [4,5] and the rapid development of Convolutional Neural Networks (CNNs) [6,7], plenty of deep learning based methods for pose estimation have been proposed and achieved significant progress. Traditionally, most of these methods predict human poses from single images. Although these methods can be directly applied to video data, they usually obtain suboptimal performances because the direct application of image-based methods cannot leverage the rich temporal information inherent in video data. In this paper, we focus on improving human pose estimation in videos by fully exploring the temporal information.

Some works already attempt to integrate temporal information into the deep models to estimate human poses in videos. These works can be generally classified into two categories: The **first category** focuses on model-based methods, which adopt 3D convolution [8] or RNN [9,10] techniques to learn spatio-temporal representations of video clips. These methods can model spatial and temporal information jointly in an end-to-end framework. However, 3D convolution and RNN have limited ability to explore the temporal consistency (e.g., geometric transformations of human body parts) between adjacent video frames. The **second category** concentrates on posterior enhancement methods [11–14] that adopt optical flow to warp predicted heatmaps of neighboring frames onto that of the target frame. Since optical flow defines the distribution of apparent velocities of the movement of brightness patterns in an image [15], these methods can explicitly exploit the temporal consistency. Despite their promising results, optical flow estimation is computationally intensive and susceptible to the occlusion and motion blur problems in unconstrained videos, which affects the performance of pose estimation to some extent.

To overcome the problems arising from these methods in the above two categories, we propose a video-based pose estimation

* Corresponding Author.
E-mail address: likan@bit.edu.cn (K. Li).

https://doi.org/10.1016/j.patcog.2020.107258
0031-3203/

model that effectively explores the temporal consistency of videos. The core of the model is the novel Temporal Consistency Exploration (TCE) module which has major advantages over the previous model-based and posterior enhancement methods. On the one hand, the TCE can explicitly explore the temporal consistency through a learnable module. On the other hand, it is more efficient as it does not need the post-processing and extra calculation of optical flow. The TCE module captures the temporal consistency at the feature level based on the fact that the spatial information of body joint locations is well preserved in feature maps [16]. In a nutshell, the TCE module follows a recurrent architecture and predicts the geometric transformations between neighboring feature maps through the learnable offset field. Then it deforms the neighboring feature maps, and the resultant deformed feature map is combined with the original map to produce enhanced feature maps through a temporal aggregation. Moreover, since the temporal information from both forward and backward directions are complementary for predicting human joint positions, the TCE module is designed to capture temporal consistency from both directions. In addition to temporal consistency, at the same time, recent researchers found that rich spatial context has proven to play an essential role in human pose estimation [17–19]. Therefore, in our work, we further design the multi-scale TCE which tightly integrates the spatial pyramid within the TCE module. The spatial pyramid increases the receptive field of the TCE module as well as facilitating the TCE module to explore the geometric transformations at multi-scale spatial levels. Using the powerful multi-scale TCE module, we extend the encoder-decoder network architecture for exploring temporal information and achieve significant improvements in video-based pose estimation.

We comprehensively evaluate the proposed model on the public challenging datasets: sub-JHMDB [20] and Penn [21]. The results demonstrated that our model outperforms recent methods and achieves state-of-the-art performances on the two video-based pose datasets. The contributions of our work are summarized as follows:

1. In this work, we propose a video-based pose estimation model that explicitly explores the temporal consistency in videos. To achieve that, we design a novel TCE module that captures geometric transformations between frames at the feature level using the learnable offset field.
2. We explore the multi-scale geometric transformations at the feature level by tightly integrating the spatial pyramid within the TCE module, which achieves further performance improvements.
3. Our model achieves 96.4% and 99.2% average accuracy on Sub-JHMDB and Penn datasets respectively using the PCK@0.2 metric, which outperforms all recent approaches.

2. Related work

2.1. Image-based pose estimation

Traditional methods for pose estimation in images mostly rely on hand-crafted features (e.g., SIFT, HOG) and seek powerful graph models, such as pictorial structure models [22], hierarchical models [23,24] and non-tree models [25–27], to represent the spatial correlations between human joints. However, these methods lack generalization ability in some cases where joints are either truncated or severely occluded.

With the availability of large-scale human pose datasets [4,5] and the rapid development of CNNs, deep learning based methods have proven to be more robust and effective for the task of human pose estimation. Mainstream works [18,19,28–30] commonly employed the multi-stage architecture to refine the output of each network stage iteratively. In particular, Wei et al. [28] proposed the Convolutional Pose Machine (CPM), which produces increasingly refined pose estimations by directly operating on belief maps from previous stages. Cao et al. [18] introduced the Part Affinity Fields (PAFs) to learn the association of body parts based on the CPM architecture, which significantly outperformed previous works. Newell et al. [19] introduced a “stacked hourglass” architecture that improves the performance by repeating bottom-up, top-down processing. This multi-stage architecture has achieved state-of-the-art results in many image-based benchmarks. Some other works [17,31,32] attempted to learn the feature pyramid in CNNs to capture the various spatial relationships across all scales. For example, He et al. [32] and Chen et al. [17] applied the feature pyramid structure for pose estimation by adopting the Feature Pyramid Network (FPN) [33]. Yang et al. [31] designed the Pyramid Residual Module (RPM) that learns feature pyramids using different subsampling ratios in a multi-branch network. Besides, there are also some methods [16,34–38] that combined CNNs with graphical models to learn both convolutional features and joint spatial constraints in an end-to-end network. For example, Tompson et al. [35] combined a CNN and a Markov Random Field (MRF) to exploit the spatial relationships between human joints in a unified model. Chu et al. [16] proposed a deep structured feature learning framework that models the correlations among the convolutional feature maps of body joints for accurate pose estimation. The success of all these methods demonstrates how a large spatial context is essential for CNN-based pose estimation methods.

2.2. Video-based pose estimation

Compared to pose estimation in images, estimating poses in videos is more challenging due to the complication in utilizing temporal and motion information. Early works [39–43] relying on hand-crafted features take into account temporal information through adding the temporal links between frames on the graph models. For example, Cherian et al. [44] cast the video-based pose estimation problem as an optimization problem defined on body parts with spatio-temporal links between frames.

Recent works attempt to integrate temporal cues in the advanced deep models to improve the performance of video-based pose estimation. Among them, the most common methods [11–14] investigate temporal context by using optical flow. As optical flow defines the distribution of apparent velocities of movement, it can help to capture the geometric transformations between frames to refine the predicted heatmaps. For example, Song et al. [13] used optical flow to exploit image evidence from adjacent frames. Pfister et al. [11] utilized optical flow to align output heatmaps from neighboring frames to improve the performance of video pose estimation. However, optical flow requires extra data pre-processing and cannot handle large appearance variations due to person occlusions or motion blur. Some other methods [9,10] capture the temporal dependency through LSTM, which has become a dominating tool for sequence tasks thanks to its power in long-range temporal representation. For example, Luo et al. [9] proposed a recurrent model with LSTM to consider the temporal information for pose estimation in videos. Gkioxari et al. [10] introduced a chained model using CNNs, where the pose prediction depends not only on the input but also on the output of the previous frame. There are also methods applying 3D convolution to learn representations of video clips. For example, Girdhar et al. [8] inflated the 2D convolution in the Mask R-CNN into 3D, which leverages temporal information over video clips to generate more robust pose predictions in videos. Although these methods can learn spatio-temporal representations in an end-to-end framework, they can not explicitly exploit the geometric transformation information between adjacent frames.

In our work, we propose a unified video-based pose estimation model, which explicitly explores multi-scale temporal consistency information at the feature level. It is worth mentioning that several works [45,46] designed for the task of video-based object segmentation are closely related to our work. They adopted Atrous Spatial Pyramid Pooling (ASPP) and ConvLSTM, and they focused on capturing the dynamic visual attention and achieved compelling performance for their objective of video-based objection segmentation. Although our method is also inspired by ASPP and ConvLSTM to some degree, it focuses on effectively and efficiently capturing temporal consistency in videos. More concretely, compared with the vanilla ConvLSTM, we equip it with the deform operation to capture geometric transformations between neighboring frames at the feature level. Besides, we apply the dilated spatial pyramid module following a reduce-split-merge principle to reduce the computational cost, and we integrate it with the proposed TCE module to explore temporal consistency at multi-scale spatial levels.

3. Method

In this section, we introduce the details of our proposed video-based pose estimation model.

3.1. Problem formulation

Given an input video with T frames as $\{\mathbf{I}_t \in \mathbb{R}^{W\times H\times 3}\}_{t=1}^{T}$ in which $W \times H$ is the spatial size of frames, the goal of our model is to generate the corresponding sequence of human joint heatmaps $\{\mathbf{M}_t \in \mathbb{R}^{w\times h\times K}\}_{t=1}^{T}$, where $w \times h$ is the spatial size of heatmaps, and K indicates the number of joints to be estimated. Each position in the kth channel of the heatmap corresponds to a score that indicates how much the position belongs to the kth joint. Usually, the heatmap resolution is smaller than the input's to reduce the number of model parameters. Thus, in order to obtain the final joint positions, we select the positions with the highest score of each channel and then re-scale them to the input size. Most recent works treat the video as a sequence of independent frames. They learn a CNN to project the input frame into a convolutional feature map $\mathbf{X}_t$, and then use the Fully Convolutional Network (FCN) to predict joint heatmaps:

$$\mathbf{X}_t = \mathrm{CNN}(\mathbf{I}_t),\ \mathbf{M}_t = \mathcal{F}_{\mathrm{FCN}}(\mathbf{X}_t). \tag{1}$$

These methods ignore the temporal information inherent in video data, in particular, temporal consistency between neighboring frames.

In this work, we focus on exploring temporal consistency to estimate human poses in videos. Specifically, we propose a Temporal Consistency Exploration (TCE) module that uses feature maps of the frame $\mathbf{I}_t$ and its N temporal neighboring frames as the input. By exploring the temporal consistency of adjacent frames, it associates the original feature map $\mathbf{X}_t$ with an enhanced feature map $\mathbf{H}_t$. Then, the enhanced feature map $\mathbf{H}_t$ is fed into the FCN to predict precise joint heatmaps.

$$\begin{aligned}\mathbf{H}_t &= \mathcal{F}_{\mathrm{TCE}}(\mathbf{X}_{t-N}, \ldots, \mathbf{X}_t, \ldots, \mathbf{X}_{t+N}),\\ \mathbf{M}_t &= \mathcal{F}_{\mathrm{FCN}}(\mathbf{H}_t).\end{aligned} \tag{2}$$

In the following sections, we first introduce a simple and effective base network for pose estimation in images. Then we introduce the details of the proposed TCE module and the multi-scale TCE module. Finally, we extend the base network with the proposed multi-scale TCE module to design a novel video-based pose estimation network.

3.2. Base pose estimation network

In order to build a solid foundation for the video-based pose estimation model, we first designed a base network for estimating human poses in images. As illustrated in Fig. 1, the network is designed based on the encoder-decoder architecture where the encoder network extracts high-level convolutional features, and the decoder network recovers the high-resolution spatial information for producing output heatmaps. For the encoder, we borrow the first four residual blocks from the Residual Network (ResNet) [6] that is a powerful CNN framework to extract high-level convolutional features. After that, the decoder adopts several deconvolutional layers to gradually enlarge the spatial dimension of the feature map. Finally, we apply a 1×1 convolutional layer to generate the output heatmap.

The architecture of the base pose estimation network is different from the Stacked Hourglass architecture [19] that uses the short-cut connections between the encoder and decoder. Our encoder-decoder architecture simply uses the de-convolutional layers to generate high-resolution feature maps without using short-cut connections. It is inspired by Xiao et al. [14] and proven to be simple yet surprisingly effective. Also, we introduce two techniques to improve the performance of the base network.

On the one hand, to capture rich spatial context in images, a dilated spatial pyramid module is built upon the encoder. Different from the Atrous Spatial Pyramid Pooling (ASPP) [45,47,48], our spatial pyramid module follows a reduce-split-merge principle for reducing computational cost. As shown in Fig. 2, it first applies the point-wise convolution to project the high-dimensional feature map to a low-dimensional space. Then, multiple dilated convolution kernels with increasing dilation rates are parallelly adopted onto the feature map. The dilated convolution can efficiently compute convolutional features at any receptive field size without loss of resolution. Finally, the multiple outputs are concatenated and further combined with the input feature map by residual summation to produce the multi-scale representation.

We also adopt a random erasing technique inspired by Zhong et al. [49] to improve the robustness of the occlusion problem, since samples in pose estimation datasets usually exhibit limited variance in occlusion. Specifically, in the training phase, an image within a mini-batch is randomly selected to 'erase' a rectangle region of arbitrary size, and assign the pixels within the region with the mean pixel value of the dataset as shown in Fig. 1. In this way, augmented images with various occlusion levels can be generated, and it is a simple yet effective technique for creating more robust models for the occlusion problem. Actually, other shapes or even occluding objects are also available for erasing, which has been discussed in [50]. In our work, for simplifying the experiment setting, we directly use the random rectangle boxes for erasing.

3.3. Temporal consistency exploration

In this section, we introduce the proposed TCE module in detail. The TCE module is designed to capture temporal consistency from both temporal directions, and it processes the preceding and subsequent adjacent frames in the same way. Here, we first only consider the preceding adjacent frames to make the technique presentation clearly and briefly. Specifically, given an input frame $\mathbf{I}_t$ and its preceding temporal neighborhoods of N frames $\{\mathbf{I}_{t-N}, \ldots, \mathbf{I}_{t-1}\}$, we first produce their corresponding features maps $\{\mathbf{X}_{t-N}, \ldots, \mathbf{X}_t\}$ using the encoder described above. With the feature maps of neighboring frames, the TCE module produce an enhanced feature map for the target frame $\mathbf{I}_t$. The proposed TCE module follows a recurrent architecture and is formulated as:

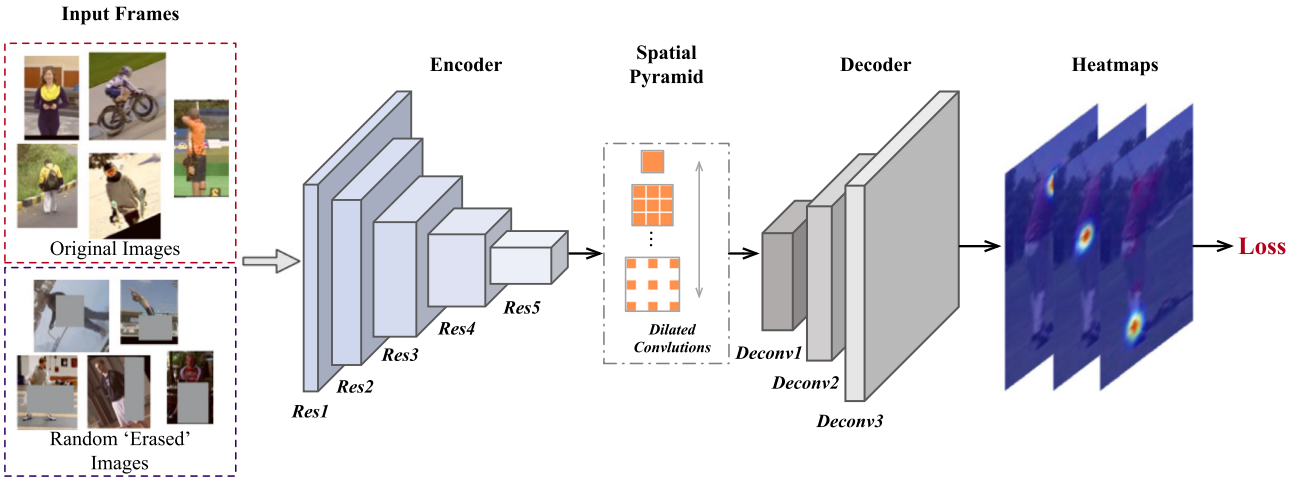

Fig. 1. The base pose estimation network is based on the encoder-decoder architecture, where the encoder network extracts high-level convolutional features and the decoder network recovers high-resolution heatmaps. In addition, we apply the spatial pyramid module and the random erasing technique to improve the robustness of the network.

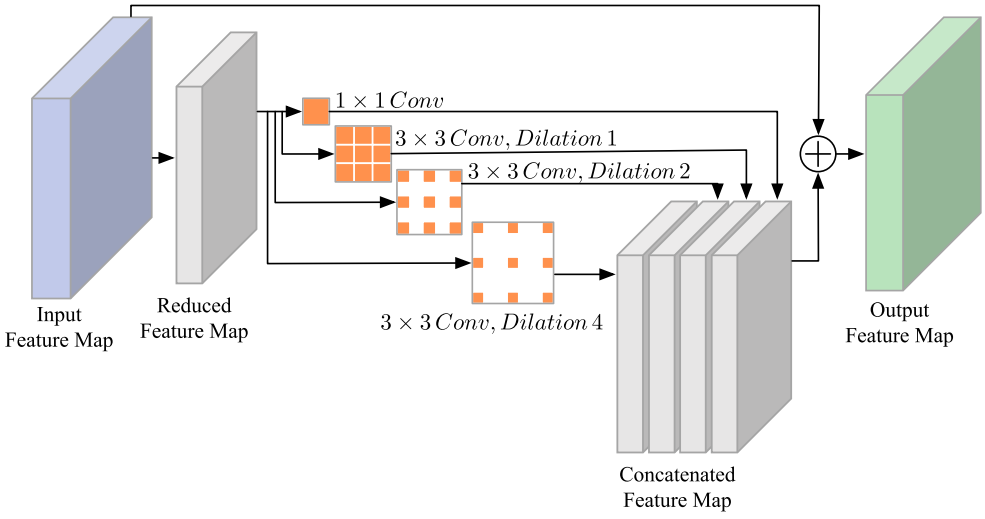

Fig. 2. Architecture of the dilated spatial pyramid module.

$$\begin{aligned}\mathbf{H}_t^p &= \mathcal{T}(\mathbf{X}_t, \mathcal{A}(\mathbf{H}_{t-1})),\\ \mathbf{H}_{t-1} &= \mathcal{T}(\mathbf{X}_{t-1}, \mathcal{A}(\mathbf{H}_{t-2})),\\ &\dots,\\ \mathbf{H}_{t-N} &= \mathbf{X}_{t-N},\end{aligned} \tag{3}$$

where $\mathcal{T}$ refers to the aggregation operation, and $\mathcal{A}$ indicates the deform operation. $\mathbf{H}_{t-1}, \dots, \mathbf{H}_{t-N}$ represent the hidden states, and we initialize the $\mathbf{H}_{t-N}$ using the original feature map $\mathbf{X}_{t-N}$. After a series of deform and aggregation operations, as a result, we can obtain the enhanced feature map $\mathbf{H}_t^p$. The details of the TCE module are illustrated in Fig. 3. In the following, we will introduce the deform operation and aggregation operation separately.

Deform Operation: In order to reinforce the feature map of the target frame, it is important to capture the geometric transformation between neighboring feature maps. To this end, we introduce the learnable offset field that is predicted based on the hidden state. And then, the hidden state is deformed according to the offset field for aligning it to the next feature map. In detail, we define the deform operation $\mathcal{A}$ as follows:

$$\begin{aligned}\Delta\mathbf{P} &= \mathbf{W}_{of} * \mathbf{H}_{t-1},\\ \mathbf{H}_{t-1}^{de} &= \text{Deform}(\mathbf{H}_{t-1}, \Delta\mathbf{P}).\end{aligned} \tag{4}$$

The offset field $\Delta\mathbf{P}$ is composed of the offsets of each spatial position. It is obtained by applying the convolution operation on $\mathbf{H}_{t-1}$, and $\mathbf{W}_{of}$ refers to the filter weights of the 2D convolution kernel. The offset field has the same spatial resolution with $\mathbf{H}_{t-1}$, and the channel dimension is 2 corresponding to 2-dim offset of each spatial position. With the offset field, we can obtain the deformed hidden state $\mathbf{H}_{t-1}^{de}$. For each spatial position $\mathbf{p}$ in $\mathbf{H}_{t-1}^{de}$, the value can be obtained via bilinear interpolation:

$$\begin{aligned}\mathbf{H}_{t-1}^{de}(\mathbf{p}) &= \mathbf{H}_{t-1}(\mathbf{p} + \Delta\mathbf{p})\\ &= \sum_{\mathbf{q}} G(\mathbf{q}, \mathbf{p} + \Delta\mathbf{p}) \cdot \mathbf{H}_{t-1}(\mathbf{q}),\end{aligned} \tag{5}$$

where $\mathbf{q}$ enumerates all integral spatial positions on $\mathbf{H}_{t-1}$, and $G(\cdot, \cdot)$ is the bilinear interpolation operation, which can be separated into two dimensional operation as:

$$G(\mathbf{q}, \mathbf{p}) = g(q_x, p_x) \cdot g(q_y, p_y), \tag{6}$$

where $g(a, b) = \max(0, 1 - |a - b|)$.

In comparisons, the proposed deform operation is different from the deformable convolution [51] that is generally used in object detection. The deformable convolution attempts to learn deformation of spatial configuration within a single image. Our ap-

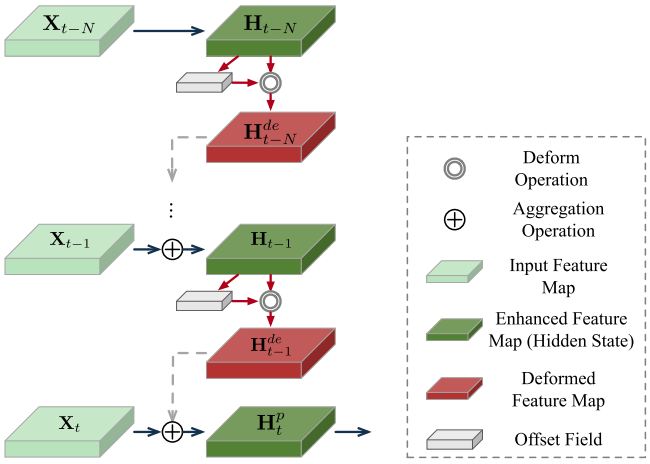

Fig. 3. Illustration of the TCE module. The TCE module follows a recurrent structure. It predicts the offset field to capture the geometric transformations between adjacent frames and produces the enhanced feature map through the deform and aggregation operations.

proach predicts the offset field in the temporal dimension aiming to model geometric transformations between neighboring frames.

Besides, compared with optical flow that has been widely used as a generic representation of motion, our method also have several differences in the following aspects: Firstly, the TCE module predicts the offset field upon the convolutional feature maps extracted from the encoder instead of raw video frames. The feature map encodes high-level semantic information, and each position on the feature map can be seen as a response of a large receptive field in the image. Thus, the predicted offset map cannot be simply viewed as the low-resolution optical flow. Secondly, the TCE module is trained without the need of alignment between video frames. Recently, many works proposed to use CNN to estimate optical flow, such as FlowNet [54] family. They are learned in a supervised manner on large-scale simulated flow datasets. Differently, our proposed TCE module is embedded into an encoder-decoder architecture, and the whole network is trained only using ground-truth human poses.

Aggregation Operation: The aggregation operation can be implemented in a variety of ways. The most simple and intuitive way is through the summation operation:

$$\mathbf{H}^p_t = \mathbf{X}_t + \mathbf{H}^{de}_{t-1}. \tag{7}$$

To further improve the capability of the TCE module, we follow the architecture and gating mechanism of ConvLSTM, which can preserve the spatial details as well as model long-term temporal aggregation. Thus, $\mathcal{T}$ can be formulated as :

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}^X_i * \mathbf{X}_t + \mathbf{W}^H_i * \mathcal{A}(\mathbf{H}_{t-1})\right)\\
\mathbf{f}_t &= \sigma\left(\mathbf{W}^X_f * \mathbf{X}_t + \mathbf{W}^H_f * \mathcal{A}(\mathbf{H}_{t-1})\right)\\
\mathbf{o}_t &= \sigma\left(\mathbf{W}^X_o * \mathbf{X}_t + \mathbf{W}^H_o * \mathcal{A}(\mathbf{H}_{t-1})\right)\\
\mathbf{c}_t &= \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tanh\left(\mathbf{W}^X_c * \mathbf{X}_t + \mathbf{W}^H_c * \mathcal{A}(\mathbf{H}_{t-1})\right)\\
\mathbf{H}^p_t &= \mathbf{o}_t \circ \tanh(\mathbf{c}_t),
\end{aligned} \tag{8}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$ are the gates, σ and tanh are the activation function of sigmoid and hyperbolic tangent respectively. For simplicity, bias terms are omitted. '$*$' denotes the convolution operation and '$\circ$' represents Hadamard product. It is worth mentioning that the convolutional kernels for generating offset fields and output features are learned simultaneously in an end-to-end manner. This guarantees the efficiency of the proposed method.

Bidirectional Temporal Consistency Exploration: In the above, we only exploit the temporal consistency information from the preceding frame sequence. However, information from both the preceding and subsequent frames are important and complementary for predicting human joint positions. As for the original feature maps $\{\mathbf{X}_{t+1}, \ldots, \mathbf{X}_{t+N}\}$ of the N subsequent frames, the TCE module processes them in the same way:

$$\begin{aligned}
\mathbf{H}^s_t &= \mathcal{T}(\mathbf{X}_t, \mathcal{A}(\mathbf{H}_{t+1})),\\
\mathbf{H}_{t+1} &= \mathcal{T}(\mathbf{X}_{t+1}, \mathcal{A}(\mathbf{H}_{t+2})),\\
&\ldots,\\
\mathbf{H}_{t+N} &= \mathbf{X}_{t+N},
\end{aligned} \tag{9}$$

where $\mathbf{H}_{t+1}, \ldots, \mathbf{H}_{t+N}$ are the hidden states, and $\mathbf{H}^s_t$ is the enhanced feature map of the subsequent frame sequence. At last, $\mathbf{H}^p_t$ and $\mathbf{H}^s_t$ are summed up to formulate the final enhanced feature map $\mathbf{H}_t$ of the frame $\mathbf{I}_t$:

$$\mathbf{H}_t = \mathbf{H}^p_t + \mathbf{H}^s_t. \tag{10}$$

3.4. Multi-Scale temporal consistency exploration

To capture rich spatial context in video data, we extend the spatial pyramid module described in Section 3.2 and designed a multi-scale TCE module that explores the geometric transformation at multi-scale spatial levels. We first apply a point-wise convolution to project the high-dimensional feature map to a low-dimensional space. Then, we simultaneously apply M dilated convolution kernels with increasing dilation rates over the feature map $\mathbf{X}_t$. After that, multi-scale feature maps are generated and fed into their respective TCE modules. Finally, $\mathbf{H}^\star_t$ that captures multi-scale spatio-temporal information is generated through concatenating the outputs of multiple TCE modules:

$$\mathbf{H}^\star_t = \left[\mathbf{H}^1_t, \ldots, \mathbf{H}^M_t\right], \tag{11}$$

where [. , .] represents the concatenation operation, and $\{\mathbf{H}^n_t\}^M_{n=1}$ indicate the outputs of M TCE modules.

3.5. Video-based pose estimation network

In this section, we introduce the video-based pose estimation network. As shown in Fig. 4, we extend the base network with the multi-scale TCE module. The overall network architecture is similar with Peng et al. [52], which is an encoder-decoder network together with RNN-based feature refinement for face alignment. We present the details about the network architecture below.

At the bottom of the model, we use the ResNet-50 and reserve the first four residual blocks as the encoder. Given an input frame $\mathbf{I}_t \in \mathbb{R}^{256\times256\times3}$ and its N temporal neighborhoods of both directions, $\{\mathbf{I}_{t-N}, \ldots, \mathbf{I}_{t-1}\}$ and $\{\mathbf{I}_{t+1}, \ldots, \mathbf{I}_{t+N}\}$, the convolutional feature maps $\{\mathbf{X}_i \in \mathbb{R}^{8\times8\times2048}\}^{t+N}_{i=t-N}$ are first extracted through the encoder.

The multi-scale TCE module consists of $M = 4$ parallel TCE modules, which take four different scale feature maps as input. To achieve that, a point convolution is first adopted to project the feature map $\{\mathbf{X}_i\}^{t+N}_{i=t-N}$ into a low-dimensional space $\mathbb{R}^{8\times8\times512}$. Then, four convolutional kernels, including one 1×1 convolutional kernel and three 3×3 convolutional kernels with increasing dilation rates as {1, 2, 4}, are parallelly adopted to generate four different scale feature maps. For each TCE module, we equip the deform operation with 3×3 convolution kernel and produce $8 \times 8 \times 2$ offset field. The aggregation operation uses the ConvLSTM equipped with 3×3 convolutional kernels, and the hidden state of ConvLSTM cell is with the size of $8 \times 8 \times 512$. Next, the output features from four TCE modules are further concatenated to generate the enhanced feature maps $\mathbf{H}^\star_t \in \mathbb{R}^{8\times8\times2048}$.

The multi-scale enhanced feature map $\mathbf{H}^\star_t$ are then fed into the decoder. The decoder consists of three deconvolutional layers with batch normalization and ReLU activation. Each deconvolutional layer has 256 filters with 4×4 kernel (stride 2, padding 1)

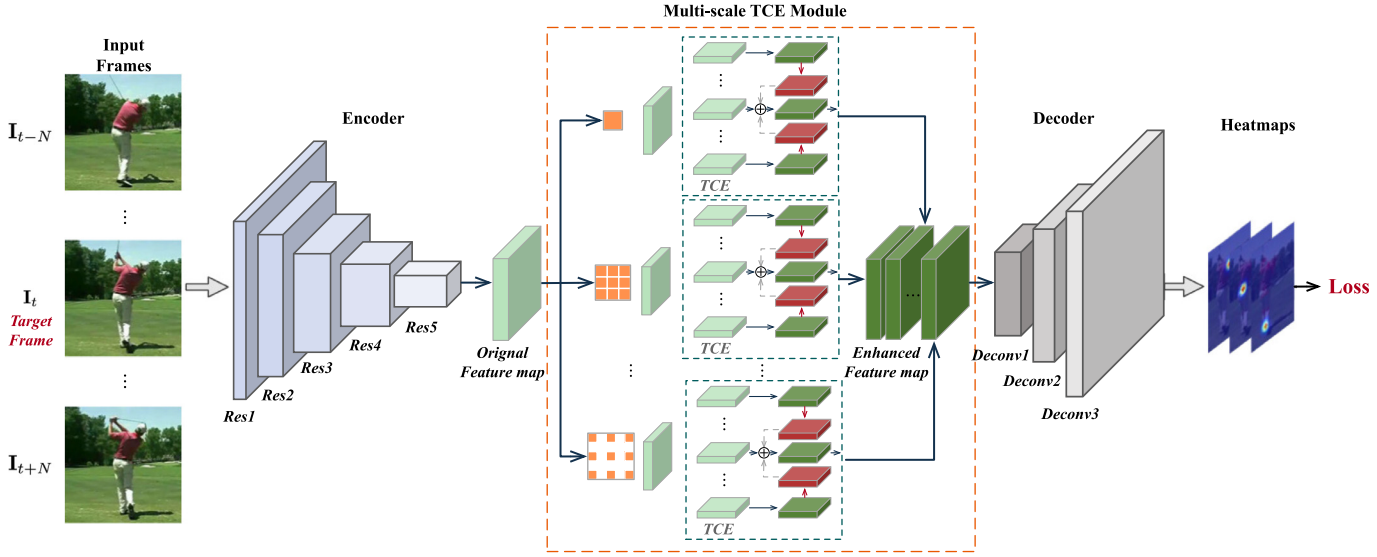

Fig. 4. Overall architecture of the proposed video-based pose estimation network. The proposed network is based on the encoder-decoder network architecture and extended with the multi-scale TCE module. The multi-scale TCE model fully explores the bidirectional temporal consistency information at multi-scale spatial levels. In this way, our model can generate temporally enhanced feature maps and obtain more precise human pose results.

resulting in 2×2 up-sampling scale. Finally, a 1×1 convolutional layer is adopted to generate the output heatmaps $\mathbf{M}_t$ with spatial resolution 64×64.

Loss Function: The ground truth heatmap of the joint k of frame t, which is written as $\mathbf{M}_t^{*k}$, is created by placing a Gaussian peak at the center location of the joint. In our work, we minimize the l_2 distance between the predicted and ground truth heatmap for each joint, and the loss function is formulated as:

$$\mathcal{L} = \sum_{k=1}^{K} \sum_{\mathbf{p}} \|\mathbf{M}_t^k(\mathbf{p}) - \mathbf{M}_t^{*k}(\mathbf{p})\|^2, \tag{12}$$

where $\mathbf{p}$ enumerates all integral spatial positions on the heatmap.

4. Experiments

4.1. Datasets

We report our performance on two public video benchmark datasets: Sub-JHMDB [20] and Penn [21]. The Sub-JHMDB dataset has 316 video clips with all 11,200 frames in the same size. It contains complete bodies with 15 joints annotated, and no invisible joint is annotated. Sub-JHMDB has three different splits of training and testing. The three splits separately have 227, 236 and 224 video clips for training and 89, 80 and 92 video clips for testing. We train our model separately and report the average result over the three splits for fair comparisons with recent methods. Besides, we also report performance on the Penn Action Dataset, which is another large video-based dataset for pose estimation. It contains in total 2326 video clips in total, with 1258 clips for training and 1068 clips for testing. 13 joints including head, shoulders, elbows, wrists, hips, knees and ankles are annotated in all frames. An additional label indicates whether a joint is visible or not in a single image.

Even though Sub-JHMDB and Penn are large-scale video datasets, the amount of training data is still insufficient considering the high correlation among frames within the same video. In order to improve the generalization ability of the model, we pre-train the base network on the MPII dataset [4], which is a large image-based pose dataset. The MPII dataset consists of images taken from a wide range of human activities with a challenging array of widely articulated full-body poses, and it has around 25k images with annotations for multiple people providing 40k annotated samples (28k training, 11k testing).

4.2. Implementation details

4.2.1. Data augmentation

Data augmentation can increase the variation of the inputs and is critical for learning robust pose estimation model. During training, we crop the frames with the target human boxes centered at images. Here, we use the person ground-truth locations provided by the datasets. Penn already annotates the bounding box within each image; the bounding boxes for sub-JHMDB are deduced from the puppet masks used for segmentation. Then, we extend either the height or the width of the human boxes to make all boxes have the same aspect ratio (1:1). Next, we further enlarge the boxes to include additional image context by rescaling the boxes with a fixed factor 1.25. After that, boxes are randomly rotated with degree $[-40^\circ, 40^\circ]$, scaled with degree $[-25\%, 25\%]$ and flipped for data augmentation. Finally, all boxes are resized to a fixed resolution (256×256). Note that the transformations will be consistent for the frames within a video.

In addition to these regular data augmentation operations, random erasing is applied for improving the robustness of the model to the occlusion problem. Specifically, we set the probability of an image undergoing random erasing is 0.5. The ratio of the area of the erased rectangle region to the original image is randomly specified between [0.02, 0.4], and the aspect ratio is randomly initialized between $[0.3, \frac{1}{0.3}]$.

4.2.2. Training details

The training procedure of our model has two steps. In the first step, we pre-train the base network on MPII dataset. We set the batch size as 32 images, and optimize the parameters using Adam [53] algorithm. The learning rate is initialized as $1e-3$ and dropped to $1e-4$ at 90 epochs and $1e-5$ at 120 epochs. We train the image-based network for 140 epochs in total.

In the second step, we fine-tune the video-based network on the Sub-JHMDB and Penn datasets respectively. We initialize the encoder using the parameters of the pre-trained base network. And then, we fix the parameters of the encoder and train the multi-scale TCE module and the decoder. The batch size is set to be 24

Table 1
Comparisons with the state-of-the-art methods on Sub-JHMDB dataset using PCK@0.2.

Method	Pre-train	Optical Flow	Head	Sho	Elb	Wri	Hip	Knee	Ank	Mean
N-best [39]	–	–	79.0	60.3	28.7	16.0	74.8	59.2	49.3	52.5
ST-Part [40]	–	–	80.3	63.5	32.5	21.6	76.3	62.7	53.1	55.7
ACPS [41]	–	–	90.3	76.9	59.3	55.0	85.9	76.4	73.0	73.8
Thin-Slicing [13]	–	✓	97.1	95.7	87.5	81.6	98.0	92.7	89.8	92.1
LSTM PM [9]	MPII&LSP	–	98.2	96.5	89.6	86.0	98.7	95.6	90.9	93.6
Ours	–	–	97.5	97.8	88.9	85.7	98.9	94.5	90.1	93.3
Ours	MPII	–	**99.3**	**98.9**	**96.5**	**92.5**	**98.9**	**97.0**	**93.7**	**96.5**

Table 2
Comparisons with the state-of-the-art methods on Penn dataset using PCK@0.2.

Method	Pre-train	Optical Flow	Head	Sho	Elb	Wri	Hip	Knee	Ank	Mean
ST-Part [40]	–	–	64.2	55.4	33.8	24.4	56.4	54.1	48.0	48.0
ACPS [41]	–	–	89.1	86.4	73.9	73.0	85.3	79.9	80.3	81.1
Chain [10]	–	–	95.6	93.8	90.4	90.7	91.8	90.8	91.5	91.8
Thin-Slicing [13]	–	✓	98.0	97.3	95.1	94.7	97.1	97.1	96.9	96.5
LSTM PM [9]	MPII&LSP	–	98.9	98.6	96.6	96.6	98.2	98.2	97.5	97.7
Ours	–	–	99.3	98.5	97.6	97.2	98.6	98.1	97.4	98.0
Ours	MPII	–	**99.8**	**99.7**	**99.2**	**98.6**	**99.2**	**99.2**	**98.7**	**99.2**

videos. We set $N = 6$ to compare with recent methods, and we set N with different values to analyze the influence of the number of neighboring frames in Section 4.4.4. Adam algorithm is used to optimize the network parameters. The learning rate is initialized as $1e-3$ and dropped by 10 times every 20 epochs, and there are 50 epochs in total. The method is implemented using the PyTorch framework and trained with Intel Xeon E5-2698 2.2GHz and one NVIDIA Tesla V100 GPU.

4.2.3. Evaluation metric

For quantitative evaluation, we adopt the PCK metric [54] to evaluate the results. An estimation is considered correct if it lies within $\alpha \cdot \max(h, w)$ from the ground truth position, where h and w refer to the height and width of the person bounding box. In our work, α is set to be 0.2 to compare with other methods consistently.

4.3. Performance on video-based pose estimation

In this section, we compare our model with recent video-based pose estimation approaches on the Sub-JHMDB and Penn datasets. Among them, N-best [39], ST-Part [40] and ACPS [41] are conventional methods that rely on hand-crafted features. They model video temporal information through the graphical model, such as spatial-temporal And-Or graph model [40]. Thin-Slicing [13] and LSTM PM [9] are recent deep learning based models. They use the advanced CNN architecture to extract deep features of frames and adopt optical flow or LSTM to capture the temporal information in videos. Note that LSTM PM [9] pre-trained the model using the combination of two image-based datasets, MPII [4] and LSP [55].

Tables 1 and 2 show the result comparisons. To fairly compare, we predict all joint positions, but only the visible ones are participating in the evaluation. Also, we report the results with and without pre-training on the MPII dataset respectively. Our model obtains the average accuracy of 96.4% and 99.2% on the two datasets. This is an improvement over the current supposed best performing method, LSTM PM [9], by 2.8% and 1.5% respectively. Compared with the optical flow based method [13], our model without pre-training also obtains largely accurate results. This demonstrates that our method can effectively exploit the temporal information even if optical flow is not used. When we pre-trained the model on the MPII dataset, the performance improvement on the Sub-JHMDB dataset was greater than on the Penn dataset since Sub-JHMDB is a relatively small-scale dataset. This demonstrates how pre-training the model on the image-based dataset with high diversity can avoid the risk of over-fitting on relatively small-scale video datasets and improve the generalization ability of the model.

In Fig. 5, we present the precision-recall curves of our method (with pre-training on the MPII dataset) on both datasets. We plot the precision-recall curves using different PCK thresholds α to show the effect of the threshold on the final accuracy. Fig. 6 shows some examples of visual results in challenging settings. It shows that our model can produce accurate human poses, which demonstrates our method is robust to the problems, such as motion blur, occlusion background and scale variations.

4.4. Ablation study

4.4.1. Analysis of the TCE module

In order to evaluate the effect of the proposed TCE module, we design a baseline and several variants to compare their performance on the split 1 of the Sub-JHMDB dataset. The baseline and variants are listed as follows:

- **Res50:** This is the baseline network that considers the video as independent frames. It is based on the encoder-decoder network described in Section 3.2 and uses the ResNet-50 as the encoder.
- **Res50-OF:** This variant is designed for comparing with the method using optical flow post-processing. Here, we use Flownet v2.0 [56] to extract the backward and forward flow of input videos. Then, we use the technique proposed by Pfister et al. [11] to warp the heatmaps of the neighboring frames, and average them with the heatmap of the target frame to get the final heatmap.
- **Res50-TCE-S:** In this variant, we adopt the proposed TCE module and use the basic summation operation as the aggregation operation.
- **Res50-TCE-C:** This variant adopts ConvLSTM for temporal aggregation.
- **Res50-TCE-BC:** This variant applies bidirectional ConvLSTM to consider temporal consistency information from both directions.

We initialize them using the parameters of the pre-trained base network and separately train them on the split 1 of Sub-JHMD dataset. Table 3 illustrates the results. We can observe that even the baseline network can achieve state-of-the-art results thanks to the solid foundation of the base network. As a post-processing

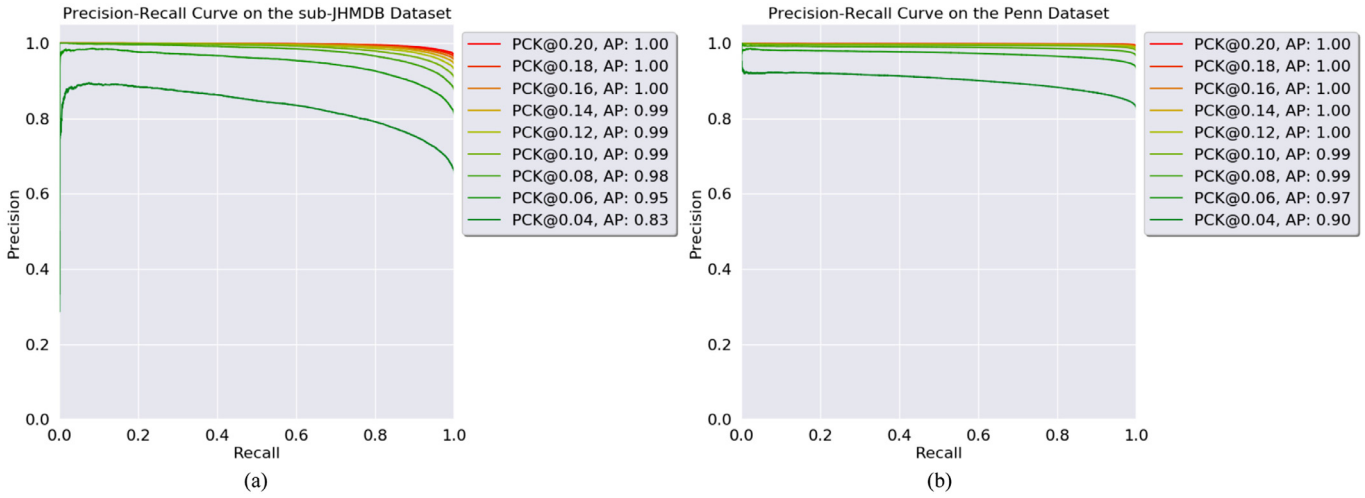

Fig. 5. Precision-recall curves of our method on the Sub-JHMDB and Penn datasets under different PCK thresholds.

Table 3
PCK@0.2 of different variants on the split 1 of Sub-JHMDB dataset.

Model	Head	Sho	Elb	Wri	Hip	Knee	Ank	Mean	Speed(ms)
Res50	98.2	96.0	91.2	88.0	98.5	95.0	93.2	94.0	10.1
Res50-OF	99.1	97.6	92.8	88.7	98.5	95.5	93.3	94.8	101.2 †
Res50-TCE-S	98.8	97.2	92.5	89.9	98.5	96.4	94.4	95.1	11.7
Res50-TCE-C	99.4	97.9	94.1	91.0	98.7	96.8	94.0	95.7	14.0
Res50-TCE-BC	99.3	97.9	94.8	91.7	98.8	97.0	94.2	96.0	23.5

† indicates that Res50-OF requires extra computation of calculating optical flow.

method, Res50-OF achieves 0.8% improvements compared with the baseline. Overall, the variants equipped with the TCE module achieve more significant improvements and outperform the Res50-OF. Res50-TCE-S outperforms the baseline by 1.1%, and Res50-TCE-C equipped with ConvLSTM aggregation function achieves better performance by 1.7%. The performance of Res50-TCE-BC is further improved and outperforms the baseline by 2.0% due to the exploration of bidirectional temporal consistency information. To validate that our method can obtain smooth results, we present a quantitative analysis in Fig. 7. We present the mean error (distance from ground-truth in pixels) curves over time of two action categories (pull up and shoot ball). It shows that Res50-TCE-BC significantly reduces the joint position errors compared with the frame-based method and optical flow-based method, which improves the prediction stability over frames. Also, we visualize two examples of results obtained by the three kinds of methods. We can observe that our method obtains apparent improvements especially for the joints with severe occlusion (i.e., elbow and hand).

In Fig. 8, we visualize two examples of the predicted offset field ΔP using the technique [57]. We found that the offset field ΔP is not directly correlated with the optical flow, and it cannot explicitly indicate the pixel-level motion information as the optical flow does. This is because the predicted offset field is based on high-level convolutional feature maps, where each position on the feature map represents a response value of a large receptive field of the image. Moreover, the TCE module is trained without explicit alignment between video frames. Thus, the TCE module can not guarantee that the predicted offset filed has the same semantics as the image-level optical flow.

In order to analyze the efficiency of the proposed TCE module, we report the runtime of the baseline and variants in Table 3, where all methods are evaluated using the same experimental platform. The baseline, which considers a video stream as a set of independent frames, achieves a processing speed of about 10.1 ms per-frame. The Res50-OF consumes 17.2 ms per-frame excluding the computation time for optical flow. Here, we use Flownet2 [56] to extract both backward and forward flows, which takes around 84ms per-frame. Thus, Res50-OF consumes 101.2 ms per-frame in total. As for our methods that solely rely on the proposed TCE module, the results show significant performance improvements at the expense of a little speed overhead relative to the baseline (range between 0.1–1.3 times). This overhead is far less than using Optical Flow, which is typically 9 times more than the speed of the baseline.

Table 4
PCK@0.2 of variants with different loss functions on the split1 of the Sub-JHMDB dataset.

Model	Regression Loss	Integral Loss	Heatmap Loss
Res50	92.0	92.8	94.0
Res50-TCE-BC	93.8	95.4	96.0

4.4.2. Analysis of the loss functions

Table 4 presents a comparison of different loss functions. It compares the performance of the variants Res50 and Res50-TCE-BC using three different kinds of loss functions: heatmap loss, regression loss, and integral loss. Here, heatmap loss is what we are using in our model. As for the regression loss, we replace the decoder with a fully connected layer to predict joint coordinates and calculate the $L1$ distances between predicted joints and ground-truth ones. Integral loss, which is proposed by Sun et al. [58], adopts the soft-argmax upon predicted heatmaps to convert them into joint coordinates in a differentiable way, and then calculates the joint location loss as supervisions. As shown in Table 4, the heatmap loss is overall best-performing, and Res50-TCE-BC achieves significant improvement in every loss function compared with Res50. This shows that the TCE module works for different prediction strategies and loss functions of 2D pose estimation.

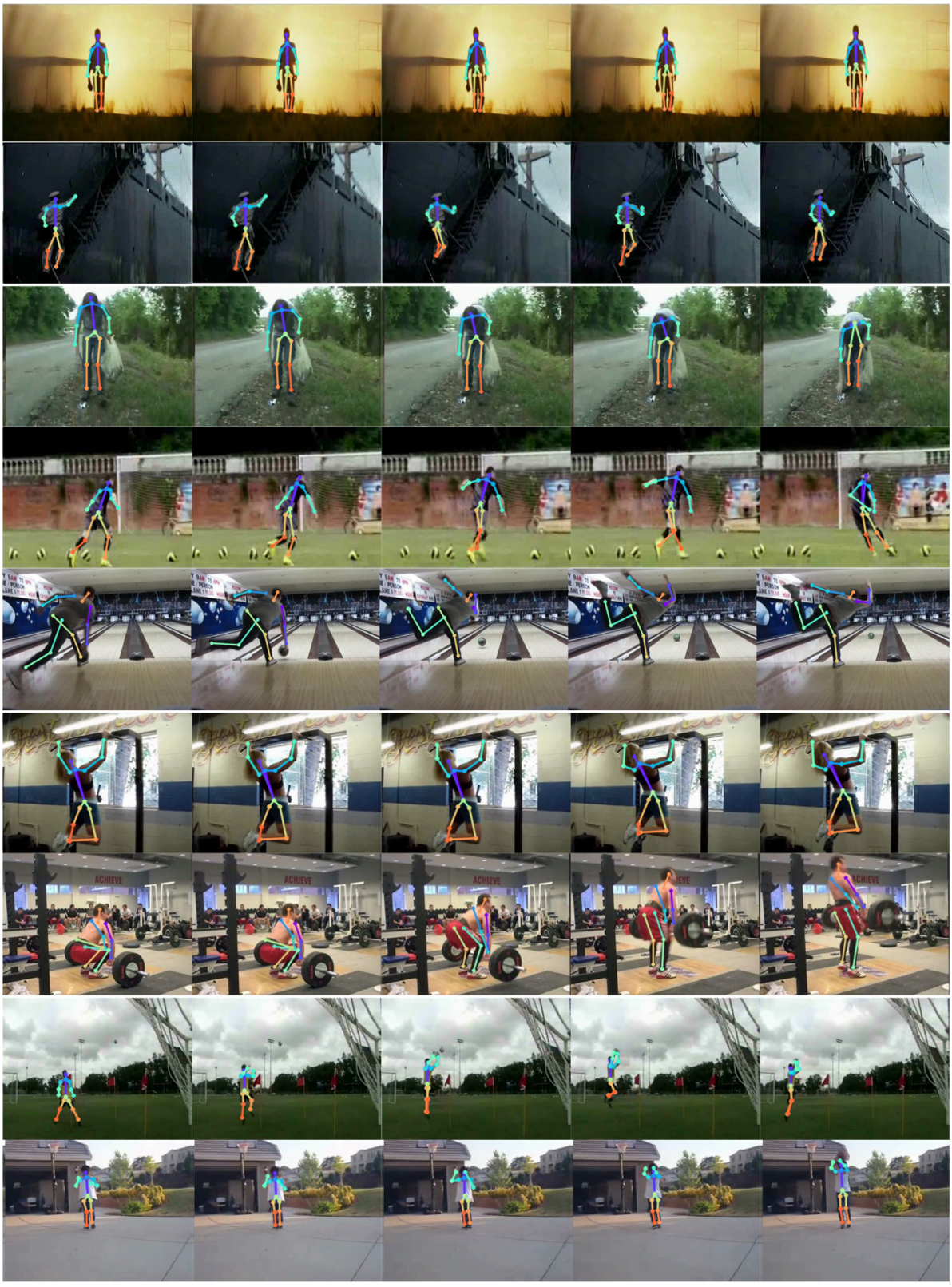

Fig. 6. Examples of pose estimation results on the Sub-JHMDB and Penn datasets. (row 1,2,3,4,5) Results of challenging samples (i.e., occlusion background and motion blur); (row 6,7,8,9) Results of persons with significant scale variations. Zoom-in for details.

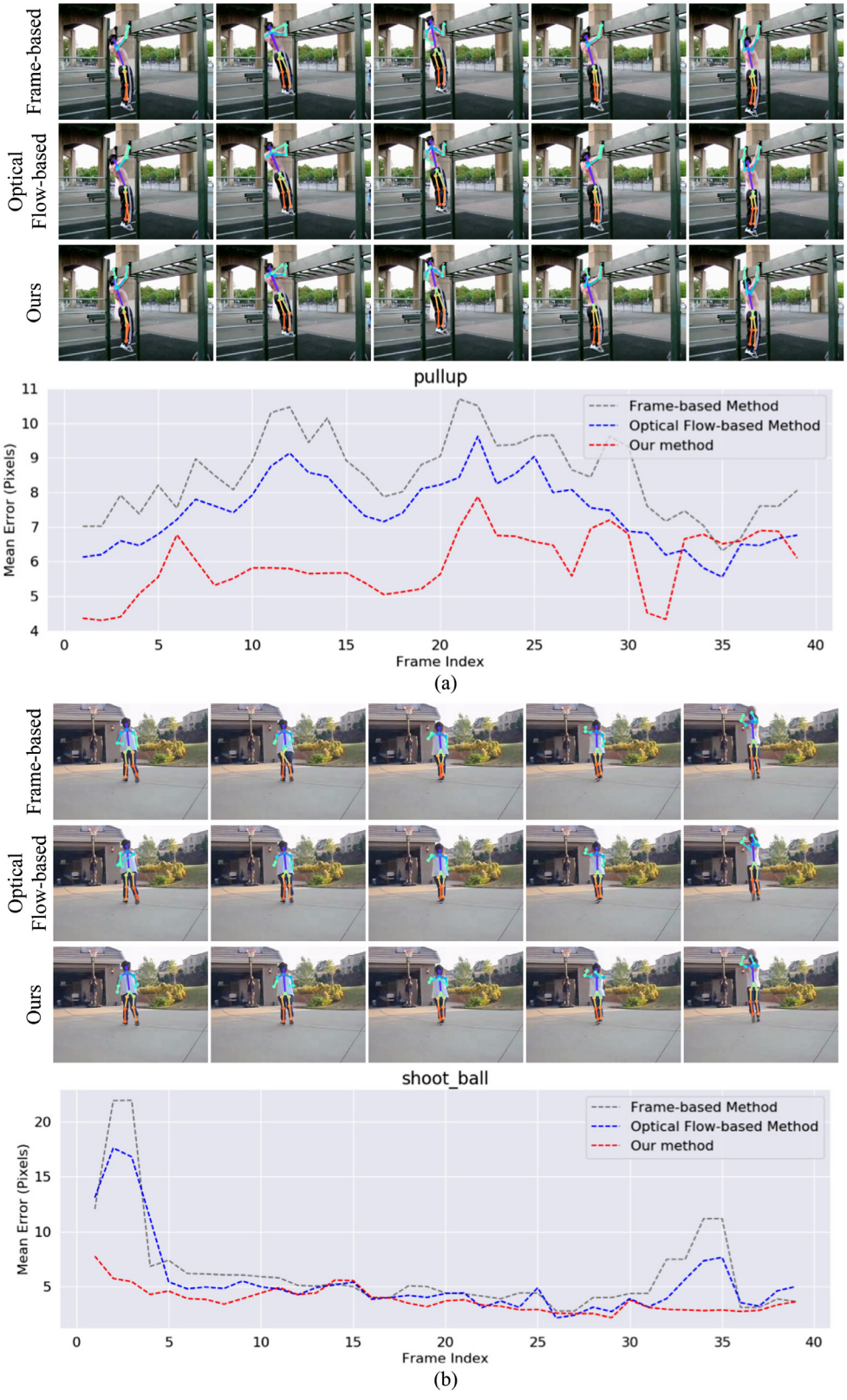

Fig. 7. Quantitative analysis of the smoothness of our results.

4.4.3. Analysis of the spatial pyramid and random erasing techniques

Here, we evaluate the effectiveness of the techniques, spatial pyramid and random erasing, used in our work. First, we design a variant named Res50-MS-TCE-BC that is equipped with the multi-scale TCE, and compare the performance between Res50-MS-TCE-BC and Res50-TCE-BC on the split 1 of the Sub-JHMDB dataset. Compared with the TCE module, the multi-scale TCE module increases the number of channels of the feature map fed into the decoder. To validate the performance improvement is indeed caused by the spatial pyramid, we design another variant named Res50-

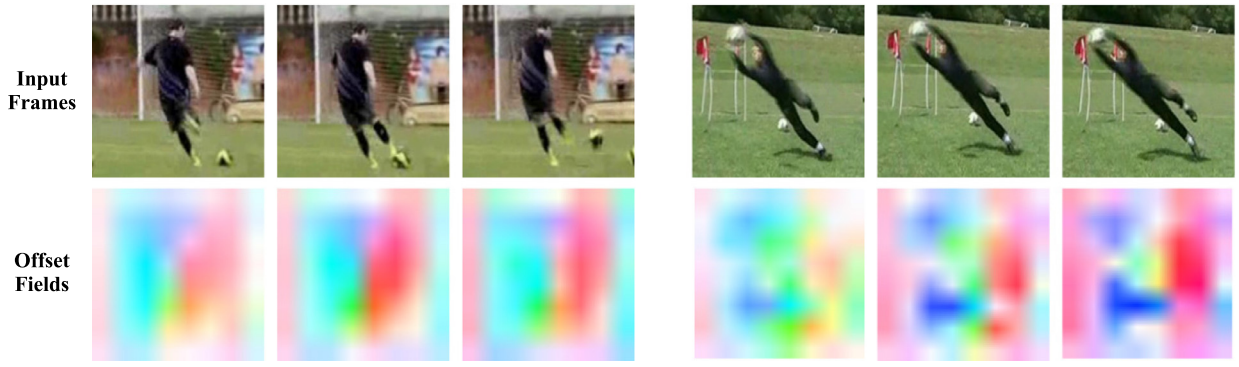

Fig. 8. Visualization of the predicted offset fields.

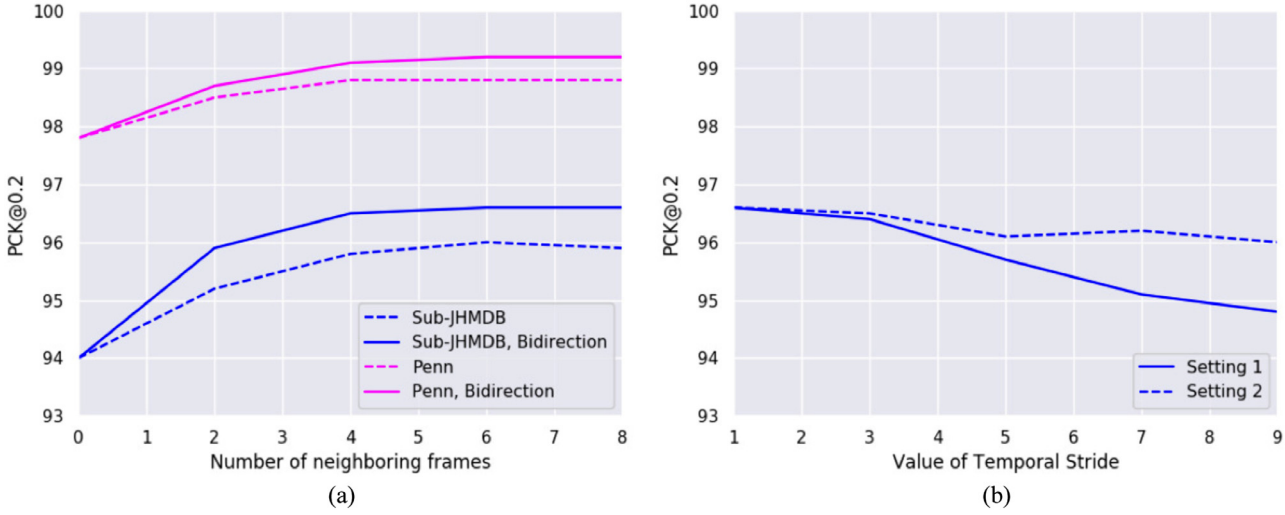

Fig. 9. Plots of the results with different numbers of neighboring frames and temporal stride values.

Table 5
Analysis of the random erasing and spatial pyramid on the split1 of the Sub-JHMDB dataset using PCK@0.2.

Model	Random Erasing	Head	Sho	Elb	Wri	Hip	Knee	Ank	Mean
Res50-TCE-BC	–	99.2	97.7	93.6	90.1	98.5	96.3	93.7	95.3
	✓	99.3	97.9	94.8	91.7	98.8	97.0	94.2	96.0
Res50-SS-TCE-BC	–	98.9	98.2	93.4	89.5	99.1	96.8	93.9	95.4
	✓	99.4	98.5	95.6	91.4	98.8	97.5	93.9	96.2
Res50-MS-TCE-BC	–	99.1	97.7	94.2	90.9	97.9	97.0	93.7	95.6
	✓	99.3	98.9	96.5	92.5	98.9	97.0	93.7	96.5

SS-TCE-BC. Different from the Res50-MS-TCE-BC, this variant applies the same convolutional kernel (3×3 convolutional kernel with dilation rate 1) for the four parallel TCE modules. Furthermore, we apply different data augmentation strategies (using random erasing or not) to train all variants for analyzing the random erasing technique. As the results shown in Table 5, Res50-MS-TCE-BC achieves consistently better performance than Res50-TCE-BC and Res50-SS-TCE-BC. This proves that the fusion of spatial pyramid and the TCE module can further boost the performance of pose estimation in videos. Besides, the models that are trained equipped with random erasing can obtain higher accuracy, especially for joints that are easily obscured like Elbow and Wrist. This illustrates that random erasing technique can effectively improve the robustness of the model for the occlusion problem.

4.4.4. Analysis of the number of neighboring frames and temporal stride

In this section, we first explore the effect of the number of neighboring frames by training the model with different N, i.e., $N = 0, 2, 4, 6, 8$. We train the model under two strategies: one only considers the preceding frames, the other considers both preceding and subsequent frames. The experiment results on the split 1 of Sub-JHMDB dataset and Penn dataset are shown in Fig. 9(a). It is obvious that bidirectional temporal consistency modeling helps to achieve better performance. When $N = 0$, the performance drops a lot since no temporal information is considered. When N increases to around 4, the performance improves, and the rate of rising decreases. At last, the accuracy remains stable when N increases to 6. This illustrates that more neighboring frames can provide more temporal information and improve performance. Meanwhile, it shows that the frames far from the target frame have relatively small effects on the results.

To analyze the influence of using different temporal strides, we consider bi-directional neighboring frames with $N = 6$ and set different temporal stride values TS, i.e., $TS = 1, 3, 5, 7, 9$. Here, we run experiments on the split 1 of the Sub-JHMDB dataset and consider two different settings. The first is training the network with $TS = 1$ and testing it using different temporal strides. The second is

using the same temporal stride during training and testing. The results of the two settings are shown in Fig. 9(b). We can observe that the performance drops as the temporal stride value becomes larger in the first setting. Thus the temporal stride during training and testing should be identical. In the second setting, the curve has small fluctuations, which illustrates the temporal stride has a limited effect on the final performance. Besides, the network obtains the best performance when the temporal stride sets 1, which shows that the TCE module can better capture the temporal consistency when the temporal stride is small.

5. Conclusion and future work

In this work, we have presented a unified deep network for estimating human poses in videos. To efficiently explore the temporal consistency in videos, we proposed the novel TCE module that captures geometric transformations between frames at the feature level. On the basis of the TCE module, we further integrated it with the spatial pyramid to explore time consistency at multi-scale spatial levels. Finally, we designed a video-based pose estimation network by extending the encoder-decoder architecture with the multi-scale TCE module. The experimental results showed that our model achieves better performance than recent video-based approaches on two popular video datasets. Moreover, we showed the effectiveness and efficiency of our model through a detailed ablative analysis.

Our current method focuses on the problem of 2D human pose estimation in single-person videos. For future work, we will explore the following two aspects. We will first consider designing a unified framework integrating the multi-scale TCE module with the person tracking technique to improve the performance of 2D pose estimation in multi-person videos. Besides, our method can be extended to the problem of 3D human pose estimation. We will extend the multi-scale TCE module to predict three-dimensional offsets and generate temporally enhanced features for 3D human pose estimation in videos.

Declaration of Competing Interest

None.

Acknowledgements

This work was supported in part by Beijing Natural Science Foundation (No. L181010, No. 4172054) and Graduate Technological Innovation Project of Beijing Institute of Technology (No. 2018CX20026).

References

[1] L. Huang, Y. Huang, W. Ouyang, L. Wang, Part-aligned pose-guided recurrent network for action recognition, Pattern Recognit. 92 (2019) 165–176.
[2] D. Zecha, M. Einfalt, C. Eggert, R. Lienhart, Kinematic pose rectification for performance analysis and retrieval in sports, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 1791–1799.
[3] K. Wang, R. Zhao, Q. Ji, Human computer interaction with head pose, eye gaze and body gestures, in: Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition, 2018. 789–789
[4] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2d human pose estimation: new benchmark and state of the art analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3686–3693.
[5] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: common objects in context, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 740–755.
[6] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[7] S. Qi, W. Wang, B. Jia, J. Shen, S.-C. Zhu, Learning human-object interactions by graph parsing neural networks, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 401–417.
[8] R. Girdhar, G. Gkioxari, L. Torresani, M. Paluri, D. Tran, Detect-and-track: efficient pose estimation in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 350–359.
[9] Y. Luo, J.S.J. Ren, Z. Wang, W. Sun, J. Pan, J. Liu, J. Pang, L. Lin, LSTM pose machines, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5207–5215.
[10] G. Gkioxari, A. Toshev, N. Jaitly, Chained predictions using convolutional neural networks, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 728–743.
[11] T. Pfister, J. Charles, A. Zisserman, Flowing convnets for human pose estimation in videos, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1913–1921.
[12] J. Charles, T. Pfister, D. Magee, D. Hogg, A. Zisserman, Personalizing human video pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3063–3072.
[13] J. Song, L. Wang, L. Van Gool, O. Hilliges, Thin-slicing network: a deep structured model for pose estimation in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5563–5572.
[14] B. Xiao, H. Wu, Y. Wei, Simple baselines for human pose estimation and tracking, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 466–481.
[15] B.K. Horn, B.G. Schunck, Determining optical flow, Artif. Intell. 17 (1–3) (1981) 185–203.
[16] X. Chu, W. Ouyang, H. Li, X. Wang, Structured feature learning for pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4715–4723.
[17] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, J. Sun, Cascaded pyramid network for multi-person pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7103–7112.
[18] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7291–7299.
[19] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 483–499.
[20] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, M.J. Black, Towards understanding action recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3192–3199.
[21] W. Zhang, M. Zhu, K.G. Derpanis, From actemes to action: a strongly-supervised representation for detailed action understanding, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2248–2255.
[22] M. Andriluka, S. Roth, B. Schiele, Pictorial structures revisited: people detection and articulated pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1014–1021.
[23] Y. Tian, C.L. Zitnick, S.G. Narasimhan, Exploring the spatial hierarchy of mixture models for human pose estimation, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 256–269.
[24] S. Sedai, M. Bennamoun, D.Q. Huynh, Discriminative fusion of shape and appearance features for human pose estimation, Pattern Recognit. 46 (12) (2013) 3223–3237.
[25] X. Lan, D.P. Huttenlocher, Beyond trees: common-factor models for 2d human pose recovery, in: Proceedings of the IEEE International Conference on Computer Vision, 2005, pp. 470–477.
[26] L. Sigal, M.J. Black, Measure locally, reason globally: occlusion-sensitive articulated pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2041–2048.
[27] Y. Wang, G. Mori, Multiple tree models for occlusion and spatial constraints in human pose estimation, in: Proceedings of the European Conference on Computer Vision, 2008, pp. 710–724.
[28] S.-E. Wei, V. Ramakrishna, T. Kanade, Y. Sheikh, Convolutional pose machines, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4724–4732.
[29] V. Belagiannis, A. Zisserman, Recurrent human pose estimation, in: Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition, 2017, pp. 468–475.
[30] Y. Zhao, Z. Luo, C. Quan, D. Liu, G. Wang, Cluster-wise learning network for multi-person pose estimation, Pattern Recognit. 98 (2020) 107074.
[31] W. Yang, S. Li, W. Ouyang, H. Li, X. Wang, Learning feature pyramids for human pose estimation, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1281–1290.
[32] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[33] T.-Y. Lin, P. Dollár, R.B. Girshick, K. He, B. Hariharan, S.J. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 936–944.
[34] W. Yang, W. Ouyang, H. Li, X. Wang, End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3073–3082.
[35] J.J. Tompson, A. Jain, Y. LeCun, C. Bregler, Joint training of a convolutional network and a graphical model for human pose estimation, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 1799–1807.
[36] H. Fang, Y. Xu, W. Wang, X. Liu, S. Zhu, Learning pose grammar to encode human body configuration for 3d pose estimation, in: Proceedings of the AAAI Conference on Artificial Intelligence.

[37] W. Wang, Y. Xu, J. Shen, S.-C. Zhu, Attentive fashion grammar network for fashion landmark detection and clothing category classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4271–4280.
[38] W. Wang, Z. Zhang, S. Qi, J. Shen, Y. Pang, L. Shao, Learning compositional neural information fusion for human parsing, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 5703–5713.
[39] D. Park, D. Ramanan, N-best maximal decoders for part models, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 2627–2634.
[40] B.X. Nie, C. Xiong, S. Zhu, Joint action recognition and pose estimation from video, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1293–1301.
[41] U. Iqbal, M. Garbade, J. Gall, Pose for action - action for pose, in: Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition, 2017, pp. 438–445.
[42] N.-G. Cho, A.L. Yuille, S.-W. Lee, Adaptive occlusion state estimation for human pose tracking under self-occlusions, Pattern Recognit. 46 (3) (2013) 649–661.
[43] P. Kaliamoorthi, R. Kakarala, Parametric annealing: a stochastic search method for human pose tracking, Pattern Recognit. 46 (5) (2013) 1501–1510.
[44] A. Cherian, J. Mairal, K. Alahari, C. Schmid, Mixing body-part sequences for human pose estimation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2361–2368.
[45] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S.C. Hoi, H. Ling, Learning unsupervised video object segmentation through visual attention, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3064–3074.
[46] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, F. Porikli, See more, know more: unsupervised video object segmentation with co-attention siamese networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3623–3632.
[47] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.
[48] H. Song, W. Wang, S. Zhao, J. Shen, K.-M. Lam, Pyramid dilated deeper convLSTM for video salient object detection, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 715–731.
[49] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, arXiv:1708.04896 (2017).
[50] I. Sárándi, T. Linder, K.O. Arras, B. Leibe, How robust is 3d human pose estimation to occlusion?, arXiv:1808.09316 (2018).
[51] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 764–773.
[52] X. Peng, R.S. Feris, X. Wang, D.N. Metaxas, A recurrent encoder-decoder network for sequential face alignment, in: Proceedings of the European conference on computer vision, 2016, pp. 38–56.
[53] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv:1412.6980 (2014).
[54] Y. Yang, D. Ramanan, Articulated human detection with flexible mixtures of parts, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2012) 2878–2890.
[55] S. Johnson, M. Everingham, Learning effective human pose estimation from inaccurate annotation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1465–1472.
[56] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, Flownet 2.0: evolution of optical flow estimation with deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2462–2470.
[57] S. Baker, D. Scharstein, J. Lewis, S. Roth, M.J. Black, R. Szeliski, A database and evaluation methodology for optical flow, Int. J. Comput. Vis. 92 (1) (2011) 1–31.
[58] X. Sun, B. Xiao, F. Wei, S. Liang, Y. Wei, Integral human pose regression, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 529–545.

Yang Li is a Ph.D. candidate working towards the dual doctoral degree at both the Beijing Institute of Technology and University of Technology Sydney. His research interests include computer vision and deep learning.

Kan Li is currently a Professor in the School of Computer at Beijing Institute of Technology. He has published over 50 technical papers in peerreviewed journals and conference proceedings. His research interests include machine learning and pattern recognition.

Xinxin Wang is currently working toward the M.S. degree under the supervision of Prof. Kan Li in the Department of Computer Science at Beijing Institute of Technology, China. Her research interests include computer vision and deep learning.

Richard Yi Da Xu received the B.Eng. degree in computer engineering from the University of New South Wales, Sydney, NSW, Australia, in 2001, and the Ph.D. degree in computer sciences from the University of Technology at Sydney (UTS), Sydney, NSW, Australia, in 2006. He is currently an Associate Professor of School of Electrical and Data Engineering, UTS. His current research interests include machine learning, deep learning, data analytics and computer vision.