International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:13, No:12, 2019

# Online Pose Estimation and Tracking Approach with Siamese Region Proposal Network

Cheng Fang, Lingwei Quan, Cunyue Lu

***Abstract*—**Human pose estimation and tracking are to accurately identify and locate the positions of human joints in the video. It is a computer vision task which is of great significance for human motion recognition, behavior understanding and scene analysis. There has been remarkable progress on human pose estimation in recent years. However, more researches are needed for human pose tracking especially for online tracking. In this paper, a framework, called PoseSRPN, is proposed for online single-person pose estimation and tracking. We use Siamese network attaching a pose estimation branch to incorporate Single-person Pose Tracking (SPT) and Visual Object Tracking (VOT) into one framework. The pose estimation branch has a simple network structure that replaces the complex upsampling and convolution network structure with deconvolution. By augmenting the loss of fully convolutional Siamese network with the pose estimation task, pose estimation and tracking can be trained in one stage. Once trained, PoseSRPN only relies on a single bounding box initialization and producing human joints location. The experimental results show that while maintaining the good accuracy of pose estimation on COCO and PoseTrack datasets, the proposed method achieves a speed of 59 frame/s, which is superior to other pose tracking frameworks.

***Keywords*—**Computer vision**,** Siamese network, pose estimation, pose tracking.

## I. Introduction

HUMAN pose estimation is a fundamental step for people's behavior understanding and scene analysis in images. The task of pose estimation is to accurately recognize and locate the important keypoints of the human body. In order to understand the behavior of people in the video, pose tracking has become a new task in computer vision. Pose tracking is a large-scale benchmark for human pose estimation and articulated tracking in video.

Similar to many other computer vision tasks, the research of pose estimation has made significant progress with the development of deep learning and the advent of publicly available pose dataset. The works in [1], [2] increased the mean Average Precision (mAP) metric to 77.0 and 73.7 for COCO human pose benchmark [3]. The performance on the MPII benchmark [4] has been saturated in recent years. The Percentage of Correct Keypoints (PCKH-0.5) metric has increased from 80% to more than 90%, and it has already reached 93.9% [1], [5], [6]. However, there has not been much work for pose tracking. Moreover, the speeds of these systems are not satisfactory in most existing works. These methods for pose estimation and tracking are offline. Human detection, pose estimation and pose tracking are divided into sequential stages. The pose of future frames should be pre-computed in the procedure. They separately trained pose estimation and human pose matching as two modules. They are more focused on Multi-Object Tracking Accuracy (MOTA) criterion than Frame Per Second (FPS) criterion.

For the VOT task, the problem to be solved is to give the position of the target to be tracked in the first frame and predict the position of the target in subsequent frames. The object position is generally described by bounding box. Due to the outstanding work of Wang et al. [9], the two tasks of VOT and Video Object Segmentation (VOS) are unified into one framework, and the object location can even be accurately described by binary segmentation mask. As for human pose tracking, not only the position of the tracking object but also the position of the body joints of the human body needs to be estimated in frames.

In this paper, we consider the problem of estimating and tracking an arbitrary person's keypoints in video. In order to narrow the gap between VOT and human pose tracking, we propose a multi-task learning approach that combines pose estimation and pose tracking in one stage. Inspired by the works in [7]-[9], we use Siamese network with Region Proposal Network (RPN) module for human tracking in videos. Based on the publicly available large dataset PoseTrack which contains explicit information on human keypoints and the success of fast-tracking methods with fully convolutional Siamese networks, we can achieve offline trainability and retaining online speed of these methods with satisfactory pose estimation results. To the best of our knowledge, this is the first paper to perform pose tracking using Siamese network. Once Siamese network performs the task of the human keypoints estimation in the process of tracking the target person, human pose tracking task is completed.

There has been rapid progress on pose estimation with the emergence of CNN-based methods. The works in [2], [10] have demonstrated that a simple network structure can achieve very competitive pose estimation performance. Our proposed architecture can be represented by three branches. As in the framework proposed by Li et al. [8], proposal extraction is performed on the correlation feature maps of template branch and detection branch. The Siamese network learns the similarity between the target and multiple candidate by a sliding window operation to determine the position of the template image in the larger search image. In order to predict the keypoints of the target in the search picture, we added a

Cheng Fang is under postgraduate in Instrument Science and Engineering Department, Shanghai Jiao Tong university. Shanghai, China (phone: 086-18721213671; e-mail: chengfang@sjtu.edu.cn).

Lingwei Quan is under postgraduate and Cunyue Lu is working as an assistant professor in Instrument Science and Engineering Department, Shanghai Jiao Tong University. Shanghai, China (e-mail: kathyquan1995@163.com, lucunyue@sjtu.edu.cn).

pose estimation branch to the feature map of the search image. The pose estimation branch shares the same CNN features with search image branch. The keypoint labels are only needed during offline training to compute the keypoint heatmap loss and not required during tracking. Each branch of the proposed architecture contributes towards the final loss. All the three branches are trained end-to-end under the supervision of RPN.

We verify the effectiveness of our approach over two benchmark datasets: the COCO keypoints detection dataset [3] and the PoseTrack dataset [11]. The results show that our approach runs at about 62 FPS, which is superior to other existing frameworks, while preserving competitive mAP of detected keypoints.

## II. Related Works

### *A. Single Person Pose Estimation and Tracking*

Since deep convolutional neural networks are used for human keypoints detection, there are two mainstream methods for single person pose estimation. One is directly regressing to the position of keypoints [12], [13], and the other is predicting the keypoints heatmap [14]-[17], which is generated by applying 2D Gaussian centered on the ground truth joint location. The coordinates of the highest heat values are the location of human joints. In fact, the final task of pose estimation is to output the coordinates of the predicted joint in the image. However, it is an extremely nonlinear process to directly let the network output two-dimensional coordinates for optimization learning. Besides, the loss function has weaker constraints on the weight of the neural network. Therefore, most modern methods perform pose estimation by predicting keypoints heatmap. There are several advantages for heatmap prediction: (1) The neural network can be fully-convolutional since the output is a two-dimensional image and does not need a fully connected layer. (2) There is a strong correlation between the human joints (such as head and chest, neck and shoulders). However, this correlation cannot be expressed and utilized during the regression of the coordinate of each joint. On the contrary, each joint of the human body corresponds to a response heatmap. The heatmaps corresponding to an input image contains this correlativity of human joints, which can be used to guide the network to learn. In short, the regression of the head joint can help return to the chest position, and the regression of the neck joints can also help the regression of left and right shoulders, and vice versa. (3) The heatmap also captures the contrast between the foreground (human joints) and the background, and can also be used to guide the network to learn.

Pose tracking is a new topic after the emergence of the MPII Video Pose dataset [18] and PoseTrack dataset. A generic light-weight framework was proposed in [19], and both the pose estimation part and the Re-ID part of the work can be flexibly replaced. Xiu et al. [20] proposed a time-space-based method. The information of the preceding and succeeding frames is used for recovery for fuzzy and occluded frames. These works are excellent contributions in the field of multi-person pose tracking, even if their tracking speed can be further improved. However, there is not too much work for SPT. Due to the great success of the Siamese network for single object tracking; we explored the use of Siamese networks for human pose tracking.

### *B. Siamese Network Series*

Before the deep neural network is used for VOT, a large number of trackers is improved based on the correlation filter algorithm. The classic algorithms include Kernelized Correlation Filter (KCF) [21], Discriminative Scale Space Tracking (DSST) [22] and so on. With the development of deep learning algorithm, scholars try to apply deep neural network methods to the field. The Siamese tracker represented by SiamFC [7] stands out and has received enthusiastic attention from researchers. The main reason is that the SiamFC shows us an ultra-fast tracking speed and preserving good tracking accuracy. Currently, the tracking field is mainly divided into two main lines, based on correlation filtering and Siamese networks.

There are two branches in a Siamese network: template branch and detection branch. The offline-trained fully-convolutional network which compares the exemplar image *z* and the larger search image *x* obtains the similarity response of the two images. The two input images use the same CNN network $\varphi_\theta$ to extract features. The combining feature map can be obtained by cross-correlation:

$$g_\theta(z, x) = \varphi_\theta(z) * \varphi_\theta(x) \tag{1}$$

The SiamRPN [8] improves the detection accuracy and tracking speed of SiamFC by introducing RPN [23], which is used to generate anchor boxes in the object detection task. The RPN module consists of two branches. One is the classification branch for distinguishing positive and negative anchors and the other is the regression branch for producing corresponding bounding box regression coefficients. With the introduction of RPN, the Siamese network has the capability of multi-scale detection by generating different sizes of anchor mechanisms to cover various sizes, and can accurately regress to the position and size of the tracking object. The main idea of SiamMask [9] is to add a mask branch to SiamRPN for generating segmentation mask. The mask branch makes the object detection results more accurate. Moreover, SiamMask creatively unifies the VOT and VOS tasks in a framework that can simultaneously acquire the object's bounding box and segmentation mask. Based on these efforts, we propose the PoseSRPN method, which aims to unify VOT and human pose estimation tasks, so that the algorithm simultaneously outputs the human body's bounding box and keypoints.

## III. Proposed Method

### *A. PoseSRPN Framework*

The Siamese network in PoseSRPN is based on SiamMask [9]. In SiamMask, the simple cross-correlation of (1) is replaced with depth-wise cross-correlation [24]. The system performs depth-wise cross-correlation between the exemplar and search image feature map in a sliding window way. For

Siamese networks, an important trick is that the input image pair is cropped around the tracking object. We obtain the input image pairs by extracting exemplar and search images that are centered on the object. Since the sub-windows that are most difficult to distinguish from the correct location of the target and the sub-windows that have the greatest impact on tracker's performance are those adjacent to the target, we consider search images centered on the target. Therefore, the exemplar image *z* and the search image x are fixed-size images obtained by affine transformation centering on the target bounding box. The image preprocessing process is shown in Fig. 1. The left side is the original image, and the right is the image centered on the object and transformed by affine transformation. The image size after preprocessing is fixed, and the corners of the image are padded with mean values. The keypoints connection lines in both images are just to show that after processing the picture, the human joints coordinates in person keypoints annotation are also based on the new images. The pre-processed image is filled with a mean value, and the coordinates (*x', y'*) in exemplar image or search image is transformed by the point (*x, y*) in the original image. The affine transformation can be expressed by (2):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 \; a_2 \; t_x \\ a_3 \; a_4 \; t_y \\ 0 \;\; 0 \;\; 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2}$$

where $(t_x,t_y)$ represents the amount of translation, and parameter $a_i$ reflects the rotation, scaling, etc. of the image.

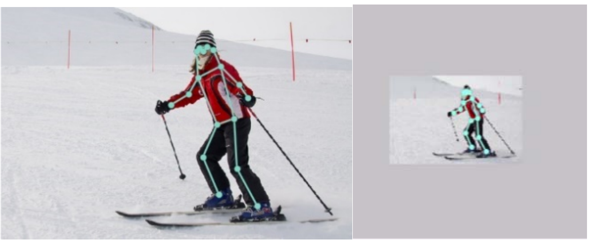
(a) original image (b) preprocessed image

Fig. 1 Image preprocessing.

The overall structure of PoseSRPN is shown in Fig. 2. There are three parts in PoseSRPN. Siamese network is used for feature extraction for exemplar image and search image; RPN module has two branches producing correct bounding box classification and location. In Fig. 2 is pose estimation branch for recognizing and locating human keypoints in search image. The preprocessed image pairs are processed by the same CNN $\varphi_\theta$ and then passed through (1) to generate a response map. The depth-wise cross-correlation function is represented as $\star d$ in Fig. 3. Each spatial element of the response map (left side of (1)) covers the similarity between the example image *z* and the corresponding spatial positions of search image *x*. For RPN module, there are k anchors of different scales and aspect ratios generated in each spatial element of response map. The response map is run through two 1×1 kernel convolutional layers to produce background/ foreground class scores and probabilities and corresponding bounding box regression coefficients. The classification loss uses cross entropy loss and the regression loss uses smooth *L1* loss to penalize incorrectly bounding box. In the following we refer to them as $L_{cls}$ and $L_{loc}$ respectively.

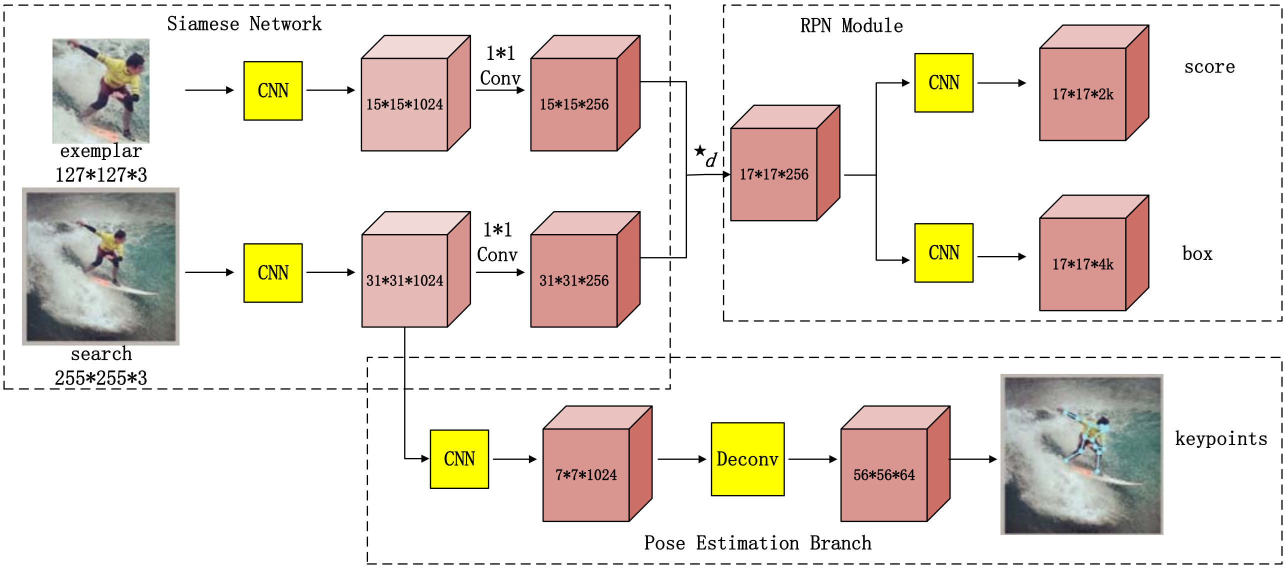

Fig. 2 Framework of PoseSRPN

The ground truth bounding box is described as $(G_x, G_y, G_w, G_h)$ which denotes center point and shape of the box. The anchor box can be correspondingly marked as $(A_x, A_y, A_w, A_h)$. The normalized distance is as (3) shows:

$$d_0 = \frac{G_x - A_x}{A_w},\ d_1 = \frac{G_y - A_y}{A_h}$$
$$d_3 = \ln\frac{G_w}{A_w},\quad d_4 = \ln\frac{G_h}{A_h} \tag{3}$$

We can define smooth *L1* loss function as below:

$$L_{1smooth}(x,\sigma) = \begin{cases} 0.5\sigma^2 x^2, & |x| < \frac{1}{\sigma^2} \\ |x| - \frac{1}{2\sigma^2}, & |x| \geq \frac{1}{\sigma^2} \end{cases} \tag{4}$$

Then the loss of regression branch is:

$$L_{loc} = \sum_{i=0}^{3} L_{1smooth}(d_i, \sigma) \tag{5}$$

The total loss of RPN module can be optimized as:

$$L_{RPN} = L_{cls} + \alpha L_{loc} \tag{6}$$

where $\alpha$ is parameter to balance the two parts.

Unlike the object detection task in Faster-Rcnn [23], in general, the same target in two adjacent frames in pose tracking task does not change too much. Therefore, in our pose tracking task, it is enough to generate anchors for one scale with different scales. The anchors generated in each spatial element of response map are less than the anchors generated in the RPN module of Faster-Rcnn. This trick makes the execution speed of the RPN module greatly improved. The anchor scale is set as 8 and the anchor ratios are set as [0.33,0.5,1,2,3] in our model. The strategy to distinguish positive and negative training samples follows the selection strategy in SiamRPN [8].

TABLE I
BACKBONE ARCHITECTURE

| block | exemplar output size | search output size | backbone |
|---|---|---|---|
| conv1 | 61×61 | 125×125 | 7×7, 64, stride 2 |
| | | | 3×3 max pool, stride 2 |
| conv2_x | 31×31 | 63×63 | $\begin{bmatrix} 1\times 1, 64 \\ 3\times 3, 64 \\ 1\times 1, 256 \end{bmatrix}\times 3$ |
| conv3_x | 15×15 | 31×31 | $\begin{bmatrix} 1\times 1, 128 \\ 3\times 3, 128 \\ 1\times 1, 512 \end{bmatrix}\times 4$ |
| conv4_x | 15×15 | 31×31 | $\begin{bmatrix} 1\times 1, 256 \\ 3\times 3, 256 \\ 1\times 1, 1024 \end{bmatrix}\times 6$ |
| adjust | 15×15 | 31×31 | 1×1,256 |
| xcorr | 17×17 | | depth-wise correlation |

We use ResNet [25] as backbone network for image feature extraction in Siamese network and pose estimation branch. The backbone structure is shown in Table I. We perform depth-wise correlation after the ResNet conv4, which is called C4 for Siamese network. We adjusted the stride of C4 to 1 so the output feature size after performing C4 is unchanged. Inspired by [2], we use the simplest but efficient network to generate keypoints heatmaps from deep and low-resolution features. Finally, while tracking the target using the Siamese network, we estimate the location of the keypoints of the tracked object in the current frame.

### *B. Pose Estimation Approach*

The network structure of the pose estimation branch is shown in Fig. 3. Notably, when performing the 1×1 convolution network after ResNet conv5, which is called C5, only the center features were extracted. Therefore, the 15×15 feature map became a 7×7 feature map. After that, we add three deconvolutional layers. We generate a heatmap by adding the two-dimensional Gaussian function to the ground truth coordinate of the keypoints; so that the network outputs predicted heatmaps, training to make the latter approach the former. Each location of human join $p$ $(x, y)$ in the image is mapped to the location $\tilde{p}\,(\left\lfloor \frac{x}{n} \right\rfloor, \left\lfloor \frac{y}{n} \right\rfloor)$ in the heatmaps, where n is the downsampling factor generated by output stride. The Gaussian kernel $Y_{xyc}$ can be described as (7):

$$Y_{xyc} = \exp\left(-\frac{(x-\tilde{p}_x)^2 + (y-\tilde{p}_y)^2}{2{\sigma_p}^2}\right) \tag{7}$$

where $\sigma_p$ is a standard deviation [26] which adapts to target size.

We use focal loss [27] to train keypoint heatmaps. The design of focal loss reduces the weight of easily categorized samples, allowing model training to focus more on samples that are difficult to classify. The loss function for training keypoint heatmaps is as follows:

$$L_{hm} = -\frac{1}{N}\sum_{xyc}\begin{cases} (1-\hat{Y}_{xyc})^{\alpha}\log(\hat{Y}_{xyc}) & if\ Y_{xyc} = 1 \\ (1-Y_{xyc})^{\beta}(\hat{Y}_{xyc})^{\alpha}\log(1-\hat{Y}_{xyc}) & otherwise \end{cases} \tag{8}$$

where $\hat{Y}_{xyc}$ is predicted heatmap. $\hat{Y}_{xyc}$=1 corresponds to a detected keypoint and $\hat{Y}_{xyc}$=0 represents for background. $\alpha$ and $\beta$ are hyper-parameters of the focal loss. We set $\alpha = 2$ and $\beta = 4$ following [26].

In order to compensate for the discretization error caused by the output stride, a local offset is predicted for pose estimation. When the keypoint locations in the heatmaps are remapped to the search image, the offset can be described as:

$$O_p = \left(\frac{x}{n} - \left\lfloor \frac{x}{n} \right\rfloor, \frac{y}{n} - \left\lfloor \frac{y}{n} \right\rfloor\right) \tag{9}$$

where $O_p$ is the offset of keypoint $p(x, y)$. We train the offset with an L1 loss

$$L_{off} = \frac{1}{N}\sum_{p}\left|\hat{O}_{\tilde{p}} - O_p\right| \tag{10}$$

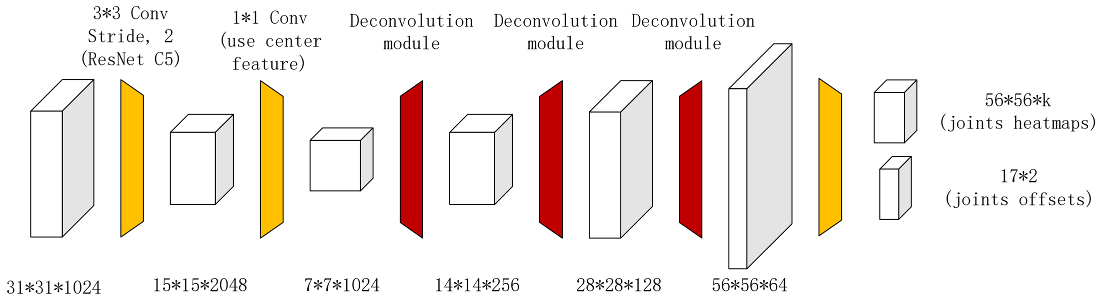

Fig. 3 Pose estimation branch network

As a result, there are two outputs in this part. First output is the predicted k (k = 17 for COCO and PoseTrack) human keypoint heatmaps. The second is local offset for recovering the discretization error. The loss function $L_{kp}$ for human keypoints estimation task is the sum of $L_{hm}$ and $L_{off}$:

$$L_{kp} = L_{hm} + L_{off} \tag{11}$$

The proposed network can be trained end-to-end on the datasets that contain the annotation of human keypoints like COCO keypoints dataset and PoseTrack dataset. The total losses for PoseSRPN can be defined as:

$$L = L_{RPN} + \beta L_{kp} = L_{cls} + \alpha \cdot L_{loc} + \beta L_{kp} \tag{12}$$

We did not set $\alpha$, $\beta$ as hyper-parameters and just set $\alpha$=1.2, $\beta$= 1.

## IV. Experiments

### A. Implementation Details

We train our network using COCO train2017 and PoseTrack 2018 dataset. The COCO dataset contains over 250,000 person instances labeled with 17 keypoints. PoseTrack is a large-scale benchmark for human pose estimation and articulated tracking in video. PoseTrack train2018 dataset contains over 790 video sequences and labeled with 17 keypoints for each person instances. The exemplar and search image are cropped as the size of 127×127 and 255×255 respectively. The network backbone is pretrained on the ImageNet [28] classification task. During training, a first warmup phase in which the learning rate increases linearly from $10^{-2}$ to $5\times10^{-2}$ for first 5 epochs is used. Finally, the learning logarithmically decreases to $10^{-3}$.

During inference time, PoseSRPN is simply evaluated once per frame without online adaptation. We find the best estimation bounding box which has the highest-scoring in RPN module. We use the most accurate bounding box as reference to crop the next frame search region and find the human keypoints in the search region. Our training phase and testing experiments are implemented on a PC with an Intel i7-8700K CPU, 16 G RAM and Nvidia GTX 2080 GPU.

### B. Pose Estimation Results

To evaluate our approach on pose estimation task, we use the COCO val2017 and PoseTrack val2018 dataset. For COCO benchmark, we use the average precision (AP) and average recall (AR) as evaluation metrics. These standard evaluation metrics are based on Object Keypoint Similarity (OKS) which plays the same role as the IoU (Intersection-over-Union) in object detection:

$$OKS = \frac{\sum_{i}\exp(-d_i^2 / 2s^2k_i^2)\delta(v_i > 0)}{\sum_{i}\delta(v_i > 0)} \tag{13}$$

where $v_i$ is the mark of the visibility of ground truth keypoint, $v_i$ = 1 represents the keypoint is visible; $d_i$ is the Euclidean distance between the ground truth value of each keypoint and the detected keypoint; $s$ is the object scale which is calculated according to the ground truth bounding box; $k_i$ represents the normalization factor of the keypoint, which reflects the degree of influence of the current keypoint pair and the overall human pose.

We validated the AP and AR metrics of the proposed algorithm on the COCO val2017 dataset. The evaluation metrics are shown in Table II. The $AP^{50}$ in the table represents the average accuracy when the OKS threshold is 0.5. The AP is the mean precision average under 10 thresholds (OKS = 0.50, 0.55... 0.90, 0.95), $AP^M$ represents the average accuracy of the medium size target, and $AP^L$ represents the average accuracy of the large size target. The algorithm proposed in this paper is compared with the existing human pose estimation methods including the pose estimation algorithm CPM [29] based on single-person ground truth position, using the object detector SSD [30] combined with CPM multi-person pose estimation algorithm and the CMU-Pose [31].

It is shown that the average accuracy of the pose estimation part of the proposed algorithm is 0.6% higher than that of the single-person pose estimation algorithm CPM. Especially when the OKS threshold is lower, the average precision is 7.4%

higher than the CMU-Pose algorithm. The visualization of the pose estimation of proposed algorithm in this paper on the COCO dataset is shown in Fig. 4. The first image on the left is the visualization of all the keypoints, and the seven images on the right with blue shadows show the heatmaps of the seven keypoints of the human body. From left to right, these seven keypoints are human joints of nose, left eye, right eye, left ear, right ear, left shoulder and right shoulder.

TABLE II
EVALUATION METRICS COMPARISON BETWEEN POSESRPN AND OTHER METHODS ON COCO VAL2017 DATASET

| Methods | AP | $AP^{50}$ | $AP^{75}$ | $AP^{M}$ | $AP^{L}$ | AR | $AR^{50}$ | $AR^{75}$ |
|---|---|---|---|---|---|---|---|---|
| GT+CPM | 62.7 | 86.0 | 69.3 | 58.5 | 70.6 | — | — | — |
| CPM+SSD | 52.7 | 71.1 | 57.2 | 47.0 | 64.2 | — | — | — |
| CMU-Pose | 65.3 | 85.2 | 71.3 | 62.2 | 70.7 | — | — | — |
| PoseSRPN | 63.3 | 92.6 | 69.8 | 59.3 | 71.3 | 76.2 | 96.1 | 83.5 |

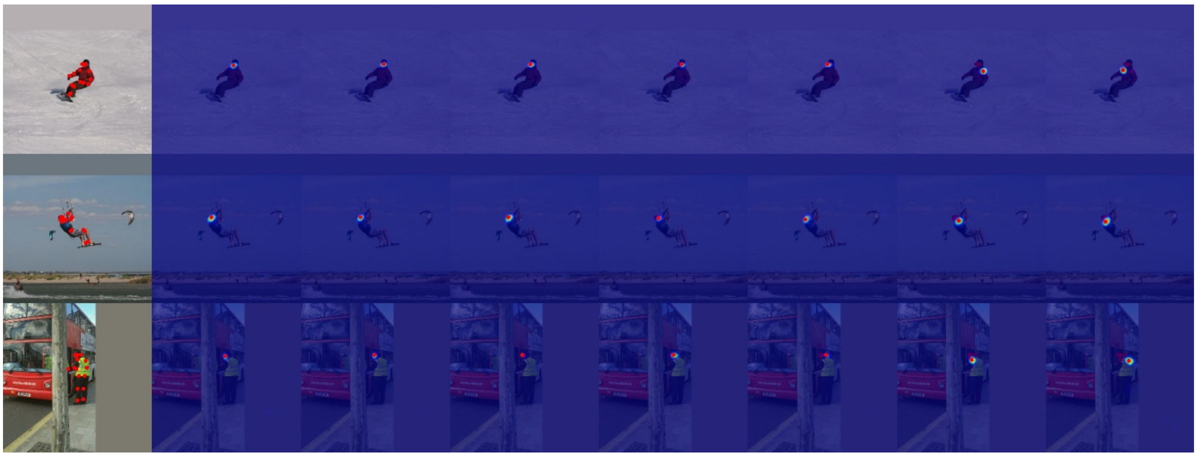

Fig. 4 Visualization of pose estimation on COCO dataset

TABLE III
EVALUATION METRICS COMPARISON BETWEEN POSESRPN AND OTHER METHODS ON POSETRACK DATASET

| Methods | Head mAP | Shoulder mAP | Elbow mAP | Wrist mAP | Hip mAP | Knee mAP | Ankle mAP | Total mAP |
|---|---|---|---|---|---|---|---|---|
| PoseTrack | 56.5 | 51.6 | 42.3 | 31.4 | 22.0 | 31.9 | 31.6 | 38.2 |
| PoseFlow | 64.7 | 65.9 | 54.8 | 48.9 | 33.3 | 43.5 | 50.6 | 51.7 |
| PoseSRPN | 67.8 | 68.3 | 59.5 | 49.9 | 35.7 | 42.6 | 54.8 | 54.1 |

The average accuracy of pose estimation for different keypoints was verified on the PoseTrack dataset. We compare the algorithm proposed in [11] and the PoseFlow [32], as shown in Table III. On the PoseTrack dataset, the mAP of each keypoint detection for the PoseSRPN algorithm proposed in this paper is 2.4% higher than that of the PoseFlow algorithm, which also verifies the effectiveness of our pose estimation algorithm. Pose estimation branch of PoseSRPN not only predicts the position of keypoints through heatmaps, but also increases the prediction of discrete error caused by convolution step, so that the accuracy of keypoints detection improved effectively.

### *C. Pose Estimation Results*

In order to test the object tracking performance of PoseSRPN, a tracking experiment was performed in the VOT2018 dataset. VOT is a test platform for single target tracking. The tracking performance evaluation indicators of VOT mainly include Expected Average Overlap (EAO), Accuracy, and Robustness. Table IV compares the tracking performance of PoseSRPN with several existing trackers.

TABLE IV
EVALUATION METRICS COMPARISON BETWEEN POSESRPN AND OTHER METHODS ON VOT2016

| Methods | Accuracy | Robustness | EAO |
|---|---|---|---|
| CSRDCF [33] | 0.466 | 0.318 | 0.263 |
| STRCF [34] | 0.523 | 0.215 | 0.345 |
| ECO [35] | 0.484 | 0.276 | 0.280 |
| PoseSRPN | 0.530 | 0.460 | 0.314 |

TABLE V
POSE TRACKING SPEED COMPARISON BETWEEN POSESRPN AND OTHER METHODS

| Methods | PoseFlow [32] | LightTrack [36] | PoseSRPN |
|---|---|---|---|
| fps | 10 | 0.76 | 52 |

As shown in Table V, PoseSRPN has a tracking speed of 59 frames/s, which is faster than the pose tracking framework PoseFlow and LightTrack. This is mainly due to the speed advantage of the Siamese network tracker compared to other trackers. Although the RPN module is combined in the Siamese network, the number of anchors generated in the RPN module is reduced. Therefore, the algorithmic computation is less than that of the object detection task in [23], but it can help the Siamese network to achieve multi-scale detection, which is beneficial to the detection accuracy of keypoints.

## V.Conclusion

In this paper, a multi-task approach for pose estimation and tracking based on Siamese network is proposed. To the best of our knowledge, this is the first paper to use Siamese network for pose tracking. A pose estimation branch is added to the search image branch of the Siamese network to enable real-time keypoint detection. The pose estimation branch replaces the complex upsampling and convolution network structure with deconvolution, and therefore has a simple network framework. We use a heatmap-based keypoint detection method, and it increases the prediction of the offset caused by the convolution step. The final predicted coordinates of keypoints are obtained by adding the coordinates of the heatmap estimation to the predicted offset. The RPN module introduces a multi-scale method for object detection, which improves the accuracy of object detection and further improves the accuracy of pose estimation. The experimental results show that while maintaining the good accuracy of pose estimation on COCO and PoseTrack datasets, the proposed algorithm achieves a speed of 59 frame/s, which is superior to other pose tracking frameworks.

## References

[1] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. In CVPR, 2019.

[2] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. ECCV, 2018.

[3] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In ECCV, 2014.

[4] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In CVPR, June 2014. 2

[5] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. Cascaded Pyramid Network for Multi-Person Pose Estimation. In CVPR, 2018.

[6] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In ECCV, pages 483–499. Springer, 2016.

[7] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional Siamese networks for object tracking. In ECCV Workshops, 2016

[8] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu. High performance visual tracking with siamese region proposal network. In CVPR, 2018

[9] Q. Wang, L. Zhang, L. Bertinetto, W. Hu. Fast online object tracking and segmentation: a unifying approach. In CVPR 2019.

[10] X. Zhou, D. Wang, P. Krähenbühl. Objects as points, *arXiv preprint arXiv:1904.07850*, 2019.

[11] U. Iqbal, A. Milan, and J. Gall. Posetrack: Joint multi-person pose estimation and tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 1, 2017.

[12] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In CVPR, pages 1653–1660, 2014.

[13] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In CVPR, pages 4733–4742, 2016.

[14] J. Tompson, A. Jain, Y. Lecun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. Eprint Arxiv, pages 1799–1807, 2014.

[15] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured feature learning for pose estimation. In CVPR, pages 4715–4723, 2016.

[16] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. In CVPR, pages 5669–5678, 2017.

[17] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In CVPR, pages 3073–3082, 2016.

[18] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele. Arttrack: articulated multiperson tracking in the wild. In CVPR, 2017.

[19] G. Ning, H. Huang. LightTrack: a generic framework for online top-down human pose tracking. *arXiv preprint arXiv: 1905.02822*, 2019.

[20] Y. Xiu, J. Li, H. Wang, Y. Fang, and C. Lu. Pose flow: Efficient online pose tracking. BMVC, 2018.

[21] J. Henriques, R. Caseiro, P. Martins, *et al*. High-speed tracking with kernelized correlation filters. PAMI 37(3) (2015) 583-596.

[22] M. Danelljan, G. Häger, F. Khan, M. Felsberg: Accurate scale estimation for robust visual tracking. In: BMVC 2014.

[23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. In International Conference on Neural Information Processing Systems, pages 91–99, 2015.

[24] L. Bertinetto, J. F. Henriques, J. Valmadre, P. H. S. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. In Advances in Neural Information Processing Systems, 2016.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[26] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In ECCV, 2018.

[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Doll´ar. Focal loss for dense object detection. ICCV, 2017.

[28] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al*. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 2015. 5.

[29] S. E. Wei, V. Ramakrishna, T. Kanade, *et al*. Convolutional Pose Machines (C)// The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4724-4732.

[30] W. Liu, D. Anguelov, D. Erhan, *et al*. SSD: Single Shot MultiBox Detector (C)// Leibe B, Matas J, Sebe N, Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9905. Cham: Springer, pp. 21-27.

[31] Z. Cao, T. Simon, S. E. Wei, et al. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields (C)// The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7291-7299

[32] Y. Xiu, J. Li, H. Wang, *et al*. Pose Flow: Efficient Online Pose Tracking (J). *arXiv preprint arXiv:1802.00977*, 2018.

[33] A. Lukezic, T. Vojir, L. C. Zajc, *et al*. Discriminative Correlation Filter with Channel and Spatial Reliability (C)// The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6309-6318.

[34] F. Li, C. Tian, W. Zuo, *et al*. Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking (C)// The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4904-4913.

[35] M. Danelljan, G. Bhat, F. S. Khan, *et al*. ECO: Efficient Convolution Operators for Tracking (C)// The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6638-6646.

[36] G. Ning, H. Huang. LightTrack: A Generic Framework for Online Top-Down Human Pose Tracking (J). *arXiv preprint arXiv:1905.02822*, 2019.