Semi-Supervised 2D Human Pose Estimation Driven by Position Inconsistency Pseudo Label Correction Module (Supplementary Material)

Linzhi Huang[1, 2 *] Yulong Li[2] Hongbo Tian[1, 2*] Yue Yang[2] Xiangang Li[2]

Weihong Deng[1 †] Jieping Ye[2]

[1]Beijing University of Posts and Telecommunications, [2]Beike

{huanglinzhi, tianhongbo, whdeng}@bupt.edu.cn

{liyulong008, yangyue092, lixiangang002, yejieping}@ke.com

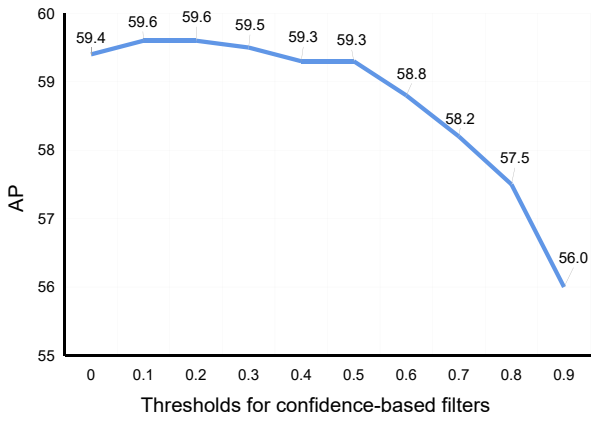

Figure 1. Hyper-parameter analysis of the thresholds for confidence-based filters.

1. The Thresholds for Confidence-Based Filters

We performed the ablation experiment by adjusting the confidence threshold of the confidence-based filters. If the maximum response of a heatmap (pseudo label) is larger than a threshold, we use it as supervision. As shown in the Figure. 1, the best result is obtained when the threshold is 0.1, and the larger the threshold is, the worse the result is. Because category confidence is not positively correlated with positioning quality, the higher the confidence threshold, the more useful data will be deleted.

*This work was done when the authors were visiting Beike as interns.

†Corresponding author.

Table 1. AP of different methods on BKFisheye when different numbers of labels are used. Backbone is ResNet18 [2].

Methods	Dataset	1K	2K	3K	All
Supervised [6]	BKFisheye	31.6	41.4	51.7	65.2
Cons [7]	BKFisheye	50.7	55.9	61.2	—
Dual [7]	BKFisheye	53.4	57.0	61.7	—
Ours	**BKFisheye**	**54.7**	**58.9**	**63.3**	—

2. Algorithm Flow

Algorithm 1 describes the algorithm process of our method. (1) $\mathcal{M}_A^{'}$, $\mathcal{M}_B^{'}$ and $\mathcal{M}_C^{'}$ trains on labeled data and get the supervised loss L_{sup}. (2) Generate easy augmentation I_e and hard augmentation I_h. (3) Then the easy augmentation is fed into two networks ($\mathcal{M}_A^{'}$ and $\mathcal{M}_B^{'}$) to produce targets HM_{e1}, HM_{e2}. (4) Input unlabeled data I_u into the SSCO module to get the hard sample I_h with occlusion, and hard data augmentation Aug_h is performed on it. (5) I_h is fed into two networks ($\mathcal{M}_A^{'}$ and $\mathcal{M}_B^{'}$) to produce student predictions HM_{h1}, HM_{h2}. (6) Use PCM module to get the corrected pseudo labels HM_{Final}. (7) I_h is fed into $\mathcal{M}_C^{'}$ to produce student predictions HM_{h3}. (8) Use pseudo labels and student predictions to get the unsupervised loss L_{unsup}. (9) Use L_{sup} and L_{unsup} to update $\mathcal{M}_A^{'}$, $\mathcal{M}_B^{'}$ and $\mathcal{M}_C^{'}$ parameters.

3. Supplementary content of Fisheye Datasets

In actual scenes, we often deploy different types of sensors to collect data under different conditions, such as indoor overhead fisheye cameras. At present, there are several public datasets related to human detection in overhead fisheye images (HABBOF [3], CEPDOF [1], and WEPDTOF

Algorithm 1 SSPCM (Ours)

Input: $\mathcal{L} = \{(I_l, H_l)\}_{l=1}^{N}$: Batch of labeled data.
Input: $\mathcal{U} = \{(I_u\}_{u=1}^{M}$: Batch of unlabeled data.
Input: $\mathcal{M}_A$, $\mathcal{M}_B$, $\mathcal{M}_C$: Pre-trained model parameters.
Input: $HM_{e1\to h1}^{last}$, $HM_{e2\to h2}^{last}$: Pseudo label of $\mathcal{M}_A$ and $\mathcal{M}_B$ output in the last epoch.
Output: $\mathcal{M}_A^{'}$, $\mathcal{M}_B^{'}$, $\mathcal{M}_C^{'}$: Updated model parameters.

$L_{sup} = 0$, $L_{unsup} = 0$, $L_{Final} = 0$
for each $(I_l, H_l) \in \mathcal{L}$ **do**
  Calculate supervised loss
  $HM_{s1} = \mathcal{M}_A(I_e), HM_{s2} = \mathcal{M}_B(I_e), HM_{s3} = \mathcal{M}_C(I_e)$
  $L_s = L_s + \|HM_{gt} - HM_{s1}\|^2 + \|HM_{gt} - HM_{s2}\|^2 + \|HM_{gt} - HM_{s3}\|^2$
end for
for each $I_u \in \mathcal{U}$ **do**
  Randomly sample augmentations: $I_e = Aug_e(I_u), I_h = Aug_h(I_u)$
  Compute teacher predictions: $HM_{e1} = \mathcal{M}_A(I_e), HM_{e2} = \mathcal{M}_B(I_e)$
  Semi-supervised Cut-Occlude based on pseudo keypoint perception (SSCO):
    $ID_{other1} = Rand(1, batch\_size) \Rightarrow (I_{other1}, HM_{other1})$
    $ID_{other2} = Rand(1, batch\_size) \Rightarrow (I_{other2}, HM_{other2})$
    Post-processing: $HM_{e1} \Rightarrow (x1, y1), HM_{e2} \Rightarrow (x2, y2), HM_{other1} \Rightarrow (x_{o1}, y_{o1}), HM_{other2} \Rightarrow (x_{o2}, y_{o2})$
    Clip local limb images:
      $h = Rand(1, H_{max}) \quad and \quad w = Rand(1, W_{max})$,
      $I_{limb1} = I_{other1}[y_{o1} - h : y_{o1} + h, x_{o1} - w : x_{o1} + w]$,
      $I_{limb2} = I_{other2}[y_{o2} - h : y_{o2} + h, x_{o2} - w : x_{o2} + w]$
    Paste: $(I_{limb1}, I_h, (x1, y1)) \Rightarrow I_{h1}^{'}$, $(I_{limb2}, I_h, (x2, y2)) \Rightarrow I_{h2}^{'}$
  Generate targets: $HM_{e1\to h1} = Aug_{e1\to h1}(HM_{e1}), HM_{e2\to h2} = Aug_{e2\to h2}(HM_{e2})$
  Compute student ($NetworkA$ and $NetworkB$) predictions: $HM_{h1} = \mathcal{M}_B(I_{h1}^{'}), HM_{h2} = \mathcal{M}_A(I_{h2}^{'})$
  Position inconsistency pseudo label correction module (PCM):
  for j **in** keypoint number **do**
    $Dist_1 = \frac{argmax(HM_{e1,j}) - argmax(HM_{e2,j})}{L_{HM}}$,
    $Dist_2 = \frac{argmax(HM_{e1,j}^{last}) - argmax(HM_{e2,j}^{last})}{L_{HM}}$,
    $Dist_3 = \frac{argmax(HM_{e1,j}) - argmax(HM_{e2,j}^{last})}{L_{HM}}$,
    $Dist_4 = \frac{argmax(HM_{e1,j}^{last}) - argmax(HM_{e2,j})}{L_{HM}}$
    $Dist_{min} = min(Dist_1, Dist_2, Dist_3, Dist_4) \Rightarrow (HM_{e1,j}^{min}, HM_{e2,j}^{min})$
    $HM_{Final} = 0.5 \cdot (HM_{e1,j}^{min} + HM_{e2,j}^{min})$
    Semi-supervised Cut-Occlude based on pseudo keypoint perception (SSCO):
    $ID_{other3} = Rand(1, batch\_size) \Rightarrow (I_{other3}, HM_{other3})$
    Post-processing: $HM_{Final} \Rightarrow (x3, y3), HM_{other3} \Rightarrow (x_{o3}, y_{o3})$
    Clip local limb images: $h = Rand(1, H_{max}) \quad and \quad w = Rand(1, W_{max})$,
    $I_{limb3} = I_{other3}[y_{o3} - h : y_{o3} + h, x_{o3} - w : x_{o3} + w]$
    Paste: $(I_{limb3}, I_h, (x3, y3)) \Rightarrow I_{h3}^{'}$
    Compute student ($NetworkC$) predictions: $HM_{h3} = \mathcal{M}_C(I_{h3}^{'})$
    Calculate unsupervised loss L_{unsup} by
    $L_{unsup} = L_{unsup} + \|HM_{e1\to h1} - HM_{h1}\|^2 + \|HM_{e2\to h2} - HM_{h2}\|^2 + \|HM_{Final} - HM_{h3}\|^2$
  end for
end for
$L_{Final} = L_{sup} + L_{unsup}$
update $\mathcal{M}_A$, $\mathcal{M}_B$ and $\mathcal{M}_C$ by minimizing L_{Final}

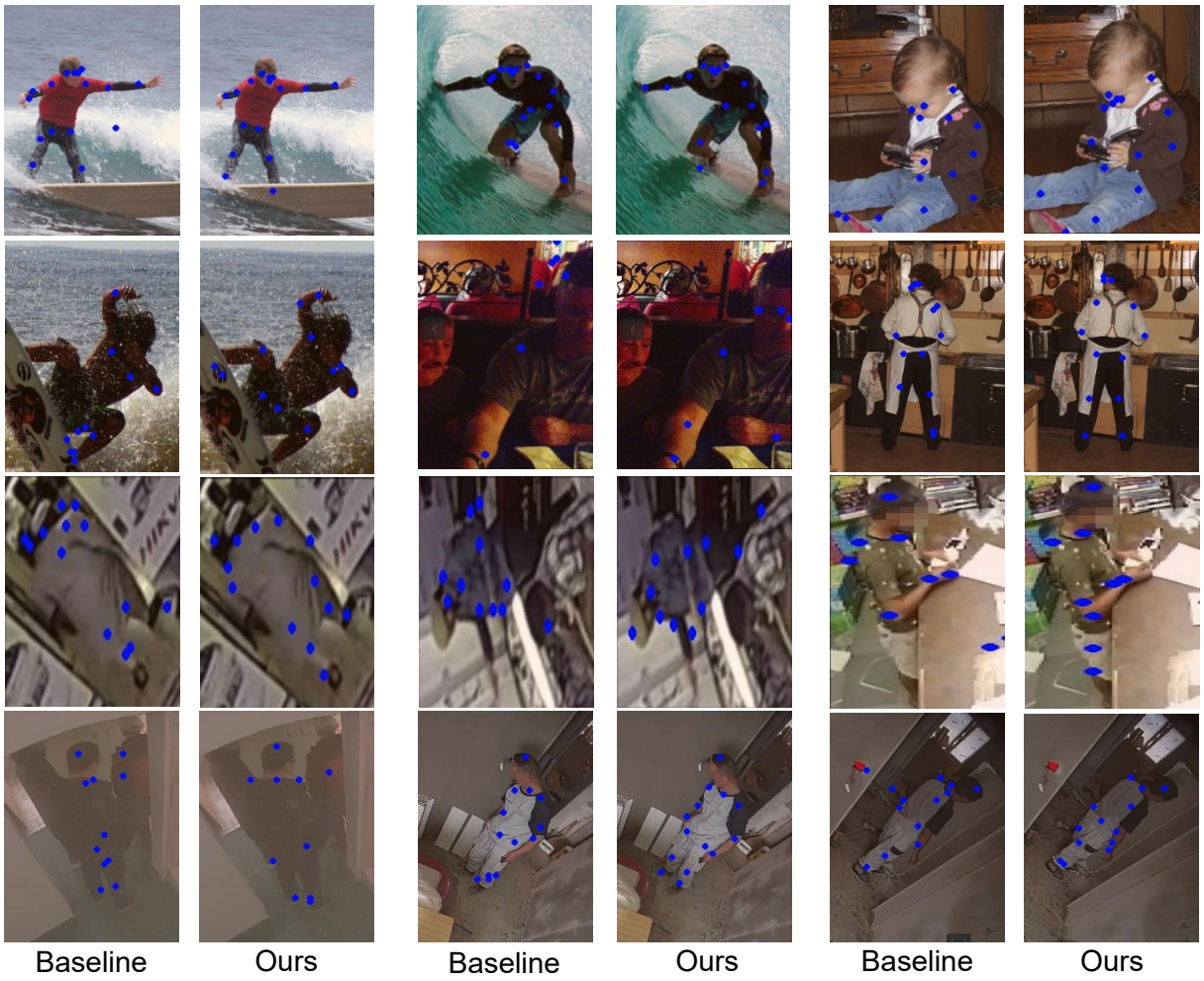

Figure 2. Qualitative results on COCO [4], WEPDTOF-Pose and BKFisheye datasets. The 1, 3, 5 column is the result of baseline (ResNet18) and the 2, 4, 6 column is the result of SSPCM (Ours).

[5]). However, these datasets do not have human keypoint labels. Therefore, we propose a new human keypoint dataset WEPDTOF-Pose based on the WEPDTOF [5] dataset. We also perform experiments on a real site scene dataset (BKFisheye) after removing sensitive information to prove the effectiveness of our method.

WEPDTOF-Pose dataset is a new human body keypoint dataset based on the WEPDTOF [5] dataset. It is an indoor dataset collected by an indoor overhead fisheye camera.

First, let's introduce the WEPDTOF dataset. The videos in WEPDTOF have been collected from YouTube and represent natural human behavior. This is important for assessing an algorithm's performance in real-world situations. This dataset is composed of 16 clips, 14 scenes, 188 distinct people, 10544 frames. Each video has 1-35 people.

Since the WEPDTOF is video dataset, and the repeatability between adjacent frames is high, we conducted 10 times down-sampling of the original dataset, and filtered person instances whose height or width is less than 50 pixels. Then, we annotate the processed images, and there are 14 keypoints in total: right shoulder, right elbow, right wrist, left shoulder, left elbow, left wrist, right hip, right knee, right ankle, left hip, left knee, left ankle, head, and lower neck. It consists of WEPDTOF-Pose $TRAIN$ (12 clips, 4688 person instances) and WEPDTOF-Pose $TEST$ (4 clips, 1179 person instances). The full amount of WEPDTOF-Pose $TRAIN$ is used as labeled data, and the CEPDOF [1] dataset is used as unlabeled data for experiments. The metric of mAP (Average AP over 10 OKS thresholds) [4] is reported.

we use the ResNet18 [2] model to conduct experiments on the BKFisheye dataset. We used 1K, 2K, and 3K labeled

data for the experiment, as shown in Table. 1. The results of supervised training using only labeled data are the worst. Our method outperforms the best semi-supervised 2D human pose estimation method in 1K, 2K, and 3K settings, and improves 1.3 mAP, 1.9 mAP, and 1.6 mAP respectively.

4. Qualitative Results

COCO [4], WEPDTOF-Pose and BKFisheye are 3 datasets used in our paper, which contain different scenarios. We use ResNet18 as the baseline to compare the qualitative results before and after using our method (SSPCM). The baseline method is easy to estimate the keypoint to the wrong object/human or wrong limb. SSPCM achieves more accurate estimations as shown in Fig. 2.

5. Limitation Analysis

Although our method improves the performance of 2D human pose estimation, our method still has some limitations. When training, our method requires more memory space and training time. The SSCO module belongs to difficult data augmentation and cannot improve model performance when supervised learning. For some difficult samples, our method may generate false pseudo labels with high confidence and low position inconsistency. However, these samples often have negative effects in training. How to filter them more accurately and effectively is a difficult task.

References

[1] Zhihao Duan, Ozan Tezcan, Hayato Nakamura, Prakash Ishwar, and Janusz Konrad. Rapid: Rotation-aware people detection in overhead fisheye images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 636–637, 2020. 1, 3

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 3

[3] Shengye Li, M Ozan Tezcan, Prakash Ishwar, and Janusz Konrad. Supervised people counting using an overhead fisheye camera. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8. IEEE, 2019. 1

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 3, 4

[5] Ozan Tezcan, Zhihao Duan, Mertcan Cokbas, Prakash Ishwar, and Janusz Konrad. Wepdtof: A dataset and benchmark algorithms for in-the-wild people detection and tracking from overhead fisheye cameras. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 503–512, 2022. 3

[6] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision*, pages 466–481, 2018. 1

[7] Rongchang Xie, Chunyu Wang, Wenjun Zeng, and Yizhou Wang. An empirical study of the collapsing problem in semi-supervised 2d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11240–11249, 2021. 1