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

Linzhi Huang¹,² * Yulong Li² Hongbo Tian¹,²* Yue Yang² Xiangang Li²
Weihong Deng¹† Jieping Ye²

¹Beijing University of Posts and Telecommunications, ²Beike
{huanglinzhi, tianhongbo, whdeng}@bupt.edu.cn
{liyulong008, yangyue092, lixiangang002, yejieping}@ke.com

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

<table>
<thead>
<tr>
<th>Methods</th>
<th>Dataset</th>
<th>1K</th>
<th>2K</th>
<th>3K</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised [6]</td>
<td>BKFisheye</td>
<td>31.6</td>
<td>41.4</td>
<td>51.7</td>
<td>65.2</td>
</tr>
<tr>
<td>Cons [7]</td>
<td>BKFisheye</td>
<td>50.7</td>
<td>55.9</td>
<td>61.2</td>
<td>—</td>
</tr>
<tr>
<td>Dual [7]</td>
<td>BKFisheye</td>
<td>53.4</td>
<td>57.0</td>
<td>61.7</td>
<td>—</td>
</tr>
<tr>
<td>Ours</td>
<td>BKFisheye</td>
<td>54.7</td>
<td>58.9</td>
<td>63.3</td>
<td>—</td>
</tr>
</tbody>
</table>

2. Algorithm Flow

Algorithm 1 describes the algorithm process of our method. (1) \( M_A', M_B' \) and \( M_C' \) 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 \( (M_A' \) and \( M_B') \) 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 \( (M_A' \) and \( M_B') \) 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 \( M_C' \) 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 \( M_A', M_B' \) and \( M_C' \) 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 fish-eye images (HABBOF [3], CEPDOF [1], and WEPDTOF

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

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.
Algorithm 1 SSSPCM (Ours)

Input: $\mathcal{L} = \{(I_i, H_i)\}_{i=1}^N$: 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: $H_{M_{\text{last} + h_1}}, H_{M_{\text{last} + h_2}}$: 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.

$L_{\text{sup}} = 0, L_{\text{unsup}} = 0, L_{\text{Final}} = 0$

for each $(I_i, H_i) \in \mathcal{L}$ do
    Calculate supervised loss
    $HM_{s_1} = \mathcal{M}_A(I_i), HM_{s_2} = \mathcal{M}_B(I_i), HM_{s_3} = \mathcal{M}_C(I_i)$
    $L_s = L_s + \|HM_{gt} - HM_{s_1}\|^2 + \|HM_{gt} - HM_{s_2}\|^2 + \|HM_{gt} - HM_{s_3}\|^2$
end for

for each $I_u \in \mathcal{U}$ do
    Randomly sample augmentations: $I_e = \text{Aug}_e(I_u)$, $I_h = \text{Aug}_h(I_u)$
    Compute teacher predictions: $HM_{e_1} = \mathcal{M}_A(I_e), HM_{e_2} = \mathcal{M}_B(I_e)$

    **Semi-supervised Cut-Occlude based on pseudo keypoint perception (SSCO):**
    $ID_{\text{other}_1} = \text{Rand}(1, \text{batch size}) \Rightarrow (I_{\text{other}_1}, HM_{\text{other}_1})$
    $ID_{\text{other}_2} = \text{Rand}(1, \text{batch size}) \Rightarrow (I_{\text{other}_2}, HM_{\text{other}_2})$
    Post-processing: $HM_{e_1} \Rightarrow (x_1, y_1), HM_{e_2} \Rightarrow (x_2, y_2), HM_{\text{other}_1} \Rightarrow (x_{o_1}, y_{o_1}), HM_{\text{other}_2} \Rightarrow (x_{o_2}, y_{o_2})$
    Clip local limb images:
    $h = \text{Rand}(1, H_{\text{max}})$ and $w = \text{Rand}(1, W_{\text{max}})$
    $I_{\text{limb}_1} = I_{\text{other}_1} | y_{o_1} - h: y_{o_1} + h, x_{o_1} - w: x_{o_1} + w |$
    $I_{\text{limb}_2} = I_{\text{other}_2} | y_{o_2} - h: y_{o_2} + h, x_{o_2} - w: x_{o_2} + w |$
    Paste: $(I_{\text{limb}_1}, I_h, (x_1, y_1)) \Rightarrow I_{h_1}', (I_{\text{limb}_2}, I_h, (x_2, y_2)) \Rightarrow I_{h_2}'$
    Generate targets: $HM_{e_1-h_1} = \text{Aug}_e(I_{h_1}(HM_{e_1})), HM_{e_2-h_2} = \text{Aug}_e(I_{h_2}(HM_{e_2}))$
    Compute student ($\text{Network}_A$ and $\text{Network}_B$) predictions: $HM_{h_1} = \mathcal{M}_B(I_{h_1}), HM_{h_2} = \mathcal{M}_A(I_{h_2})$

    **Position inconsistency pseudo label correction module (PCM):**
    for $j$ in keypoint number do
        $Dist_1 = \arg\max_{\mathcal{L}_{HM}}(HM_{e_1,j}) - \arg\max_{\mathcal{L}_{HM}}(HM_{e_2,j})$
        $Dist_2 = \arg\max_{\mathcal{L}_{HM}}(HM_{\text{other}_1,j}) - \arg\max_{\mathcal{L}_{HM}}(HM_{\text{other}_2,j})$
        $Dist_3 = \arg\max_{\mathcal{L}_{HM}}(HM_{e_1,j}) - \arg\max_{\mathcal{L}_{HM}}(HM_{\text{other}_1,j})$
        $Dist_4 = \arg\max_{\mathcal{L}_{HM}}(HM_{\text{other}_1,j}) - \arg\max_{\mathcal{L}_{HM}}(HM_{\text{other}_2,j})$
        $Dist_{min} = \min(Dist_1, Dist_2, Dist_3, Dist_4) \Rightarrow (HM_{\text{min}_{e_1,j}}, HM_{\text{min}_{e_2,j}})$
        $HM_{F_{\text{Final}}} = 0.5 \cdot (HM_{\text{min}_{e_1,j}} + HM_{\text{min}_{e_2,j}})$
    end for

    **Semi-supervised Cut-Occlude based on pseudo keypoint perception (SSCO):**
    $ID_{\text{other}_3} = \text{Rand}(1, \text{batch size}) \Rightarrow (I_{\text{other}_3}, HM_{\text{other}_3})$
    Post-processing: $HM_{F_{\text{Final}}} \Rightarrow (x_3, y_3), HM_{\text{other}_3} \Rightarrow (x_{o_3}, y_{o_3})$
    Clip local limb images: $h = \text{Rand}(1, H_{\text{max}})$ and $w = \text{Rand}(1, W_{\text{max}})$
    $I_{\text{limb}_3} = I_{\text{other}_3} | y_{o_3} - h: y_{o_3} + h, x_{o_3} - w: x_{o_3} + w |$
    Paste: $(I_{\text{limb}_3}, I_h, (x_3, y_3)) \Rightarrow I_{h_3}'$
    Compute student ($\text{Network}_C$) predictions: $HM_{h_3} = \mathcal{M}_C(I_{h_3})$
    Calculate unsupervised loss $L_{\text{unsup}}$ by
    $L_{\text{unsup}} = L_{\text{unsup}} + \|HM_{e_1-h_1} - HM_{h_1}\|^2 + \|HM_{e_2-h_2} - HM_{h_2}\|^2 + \|HM_{F_{\text{Final}}} - HM_{h_3}\|^2$
end for

$L_{\text{Final}} = L_{\text{sup}} + L_{\text{unsup}}$
update $\mathcal{M}_A, \mathcal{M}_B$ and $\mathcal{M}_C$ by minimizing $L_{\text{Final}}$
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