Large-Scale Person Detection and Localization using Overhead Fisheye Cameras

Supplemental Material

Lu Yang1, Liulei Li2, Xueshi Xin1, Yifan Sun3, Qing Song1, Wenguan Wang4

1 Beijing University of Posts and Telecommunications 2 ReLER, AAII, University of Technology Sydney
3 Baidu 4 ReLER, CCAI, Zhejiang University

https://LOAFisheye.github.io/

Table S1: Detailed statistics of LOAF. # indicates the number of elements.

<table>
<thead>
<tr>
<th></th>
<th># Video</th>
<th># Image</th>
<th># People</th>
<th>Scene</th>
<th>Season</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Total</td>
<td>Avg.</td>
<td>Max</td>
<td>Total</td>
</tr>
<tr>
<td>train</td>
<td>51</td>
<td>29,569</td>
<td>135,262</td>
<td>10.6</td>
<td>65</td>
<td>35</td>
</tr>
<tr>
<td>val</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>seen</td>
<td>3</td>
<td>1,700</td>
<td>18,460</td>
<td>10.8</td>
<td>29</td>
<td>5</td>
</tr>
<tr>
<td>unseen</td>
<td>5</td>
<td>2,900</td>
<td>29,381</td>
<td>10.1</td>
<td>44</td>
<td>8</td>
</tr>
<tr>
<td>total</td>
<td>8</td>
<td>4,600</td>
<td>47,841</td>
<td>10.4</td>
<td>44</td>
<td>2</td>
</tr>
<tr>
<td>test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>seen</td>
<td>5</td>
<td>2,774</td>
<td>28,666</td>
<td>10.3</td>
<td>41</td>
<td>5</td>
</tr>
<tr>
<td>unseen</td>
<td>10</td>
<td>5,999</td>
<td>65,993</td>
<td>10.0</td>
<td>44</td>
<td>10</td>
</tr>
<tr>
<td>total</td>
<td>15</td>
<td>8,773</td>
<td>94,659</td>
<td>10.1</td>
<td>44</td>
<td>15</td>
</tr>
<tr>
<td>Total</td>
<td>74</td>
<td>42,942</td>
<td>457,762</td>
<td>10.5</td>
<td>65</td>
<td>30</td>
</tr>
</tbody>
</table>

A. Additional Dataset Analysis

More Statistics. LOAF is captured from multiple indoor/outdoor scenes (e.g., library, classroom, street, parking lot) across three seasons, we summarize the detailed statistics in Table S1, including the number of boxes, video sequences, etc. As seen, the majority of videos are collected from outdoor environments characterized by increased complexity, larger fields of view, and a higher number of human targets when compared to the indoor ones. These videos are divided into train, val, and test sets in the ratio of 7:1:2 respectively, while ensuring an roughly even distribution of attributes (e.g., season, time) across these sets.

B. More Implementation Details

Training Objective. We extend the Generalized IoU (GIoU) loss [4] utilized in vanilla DETR [1] for bounding box regression to the rotated setup. Concretely, Brute-force search is leveraged to compute the minimum enclosing box between two rotated bounding boxes. It is implemented in a fully differentiable manner and adapted for parallel processing on GPU, which merely defers the training speed by around 5% when compared to the axis-aligned setup.

C. Qualitative Evaluation

Visual Comparison. Fig. S1-S5 compare our method with existing work qualitatively. It is obvious that our proposed method consistently presents more accurate detection and localization results, regardless of the category of scenes. Notably, it is much more effective than existing work for targets that are relatively small or densely arranged.

D. Ethical Conducts

To protect the privacy of individuals and groups, we utilize Gaussian filters to blur all visible facial regions in LOAF. The proprietary data can only be accessed for non-commercial purposes to prevent inappropriate usage.

E. Pseudo Code

We offer the pseudo code for our proposed query-based rotation equivariant training strategy in Algorithm S1.
Algorithm S1  Pseudo-code for our proposed rotation equivariant training strategy.

```python
***
I: input image
gt: ground truth
angle: degree of clockwise rotation
λ: the balance factor
***
def rotat_equi_training(I, gt):
    # F(I)
    m1 = Encoder(I)
    angle = randint(0, 360)
    # F(g(I))
    m2 = Encoder(rotate(I, angle))

    # \{q_n\}_{n=1}^{N} = gen_proposal(m1)
    query1 = gen_proposal(m1)
    # \{q_{g_n}\}_{n=1}^{N} = gen_proposal(m2)
    query2 = gen_proposal(m2)

    # D(I, \{q_n\}_{n=1}^{N})
    det1 = Decoder(m1, query1)
    # D(g(I), \{q_{g_n}\}_{n=1}^{N})
    det2 = Decoder(rotate(m1, angle), query2)

    # \mathcal{L}_{det} (\{b_{q_n}\}_{n=1}^{N}, D(I, \{q_n\}_{n=1}^{N}))
    loss1 = det_loss(det1, gt)
    # \mathcal{L}_{det} (\{b_{q_{g_n}}\}_{n=1}^{N}, D(g(I), \{q_{g_n}\}_{n=1}^{N}))
    loss2 = det_loss(det2, label_rotate(gt, angle)

    return loss1 + λ*loss2
```

References


Figure S1: Visual comparison of detection results on the test set of LOAF. ◇ indicates targets missed by our method.
Figure S2: Visual comparison of detection results on the test set of LOAF. ◇ indicates targets missed by our method.
Figure S3: **Visual comparison of detection results** on the test set of LOAF. ◇ indicates targets missed by our method.
Figure S4: Visual comparison of detection results on the test set of LOAF. ◇ indicates targets missed by our method.
Figure S5: **Visual comparison of localization results** on the test set of LOAF. We selected three targets per frame for clear visualization.
Figure S6: A collage constituted from various scenes characterized by distinct attributes.