Detective: An Attentive Recurrent Model for Sparse Object Detection

Amine Kechaou, Manuel Martinez, Monica Haurilet, Rainer Stiefelhagen
Computer Vision for Human-Computer Interaction (cv:hci)
Karlsruhe Institute of Technology, Germany
cvhci.anthropomatik.kit.edu

Advantages
- Raw detections are final detections → No post-processing required
- Predictions are not made densely over the image → No more foreground / background class imbalance
- Objects are predicted one by one → Enable better understanding at the instance level
- Possibility for Ordered Object Detection → Objects are predicted in the right order in one pass

Challenges
- Assigning target objects to predictions: target objects do not have a predefined order
- Limited related work on sparse object detection
- Combination of classification and localization is challenging for RNNs

Detective is an end-to-end sparse object detector using a ConvLSTM and attention to detect one object each iteration, until the End of Sequence token is predicted.

Encoder-Decoder Architecture:

Dynamic Matching of Targets and Predictions During Training with the Hungarian Algorithm
- Challenge: How to assign target labels to predictions during training?
- Solution: Dynamic Matching using the Hungarian Algorithm
- Cost function: \[ f(t, p) = \mu_{\text{cls}} L_{\text{cls}}(t_{\text{cls}}, p_{\text{cls}}) + \mu_{\text{loc}} L_{\text{loc}}(t_{\text{loc}}, p_{\text{loc}}) \]
- The Hungarian Algorithm efficiently finds the matching with the minimal overall cost

Ablation Study on PASCAL VOC

<table>
<thead>
<tr>
<th>Model</th>
<th>Conv-LSTM</th>
<th>Attention</th>
<th>Positional embeddings</th>
<th>Background classification</th>
<th>mAP (%) on VOC07</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>40.7</td>
</tr>
<tr>
<td></td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>49.4</td>
</tr>
<tr>
<td></td>
<td>✘</td>
<td>✘</td>
<td>✘</td>
<td>✗</td>
<td>46.5</td>
</tr>
<tr>
<td></td>
<td>✘</td>
<td>✘</td>
<td>✗</td>
<td>✗</td>
<td>47.9</td>
</tr>
<tr>
<td></td>
<td>✗</td>
<td>✘</td>
<td>✗</td>
<td>✗</td>
<td>49.9</td>
</tr>
<tr>
<td>Detective</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>52.0</td>
</tr>
</tbody>
</table>

Examples, and their attention maps each iteration