
Guillaume VAUDAUX-RUTH\(^1,2\), Adrien CHAN-HON-TONG\(^1,3\), Catherine\(^2\) ACHARD

\(^1\)ONERA \quad \(^2\)Sorbonne Université \quad \(^3\)Université Paris-Saclay

Action spotting has recently been proposed as an alternative to action detection [1] and key frame extraction. However, the current state-of-the-art method of action spotting requires an expensive ground truth.

In this work, we propose to use a reinforcement learning algorithm to perform efficient action spotting using only the temporal segments from action detection annotations.

**Context**

Action spotting has recently been proposed as an alternative to action detection [1] and key frame extraction. However, the current state-of-the-art method of action spotting requires an expensive ground truth.

In this work, we propose to use a reinforcement learning algorithm to perform efficient action spotting using only the temporal segments from action detection annotations.

**Action Spotting Framework**

In a first stage, the CNN backbone encodes the frame into a feature vector which is then forwarded to a GRU layer. The resulting hidden state vector is then individually processed by (SF), (CL), (BROW) and (crit). The (SF) stage deals with the decision of turning the current frame into a spot frame or skip it. The (CL) stage predicts the action class related to the spot frame and the (BROW) stage outputs the next video frame to look at. The (crit) stage is only used to ensure better convergence in the reinforcement learning framework.

**Action Spotting Optimization**

The global policy has to maximize the final mAP of the video being processed. Thus our local reward at step \( n \) is the difference of AP between step \( n \) and \( n-1 \) plus an entropy term:

\[
r_{\pi,n} = \text{mAP}(V_n) - \text{mAP}(V_{n-1}) + \rho H(\pi(n))
\]

and, the cumulative discounted reward is:

\[
R_{\pi,n} = \sum_{k=0}^{N-n-1} \gamma^k r_{\pi,k+n}
\]

Then, the global loss is:

\[
L_{global} = L_{cls} + \lambda_1 L_{critic} - \lambda_2 J_{actor}
\]

with \( L_{cls} \) the loss of the classification network (CL) which is a cross-entropy loss, \( L_{critic} \) is the MSE between the estimation of the value function of \( \text{(crit)} \) and the real one.

As \( J_{actor} \) objective is non differentiable, we use REINFORCE to derive the expected gradient:

\[
\nabla J_{actor} = \mathbb{E} \left[ \sum_{n=1}^{N} \log(\pi(n))(R_{\pi,n} - \mathbb{E}[R_{\pi,n} | h_n]) \right]
\]

**Results**

<table>
<thead>
<tr>
<th>Approach</th>
<th>Detection mAP</th>
<th>Spoting mAP</th>
<th>ActivityNet-1.12</th>
<th>Spoting mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>THUMOS14</td>
<td>0.5 0.7 0.95</td>
<td>Arg</td>
<td>18.6 18.6 18.3</td>
<td>18.6 18.6 18.6</td>
</tr>
<tr>
<td>Visc4 [45]</td>
<td>37.0 34.6 34.6</td>
<td>18.6 18.6 18.6</td>
<td>- 18.6 18.6 18.6</td>
<td></td>
</tr>
<tr>
<td>SSN-SW [35]</td>
<td>37.3 32.4 27.9</td>
<td>18.6 18.6 18.6</td>
<td>- 18.6 18.6 18.6</td>
<td></td>
</tr>
<tr>
<td>IC-Net [44]</td>
<td>37.6 32.4 27.9</td>
<td>18.6 18.6 18.6</td>
<td>- 18.6 18.6 18.6</td>
<td></td>
</tr>
<tr>
<td>PFPN [37]</td>
<td>38.0 32.4 27.9</td>
<td>18.6 18.6 18.6</td>
<td>- 18.6 18.6 18.6</td>
<td></td>
</tr>
<tr>
<td>SSN [39]</td>
<td>38.0 32.4 27.9</td>
<td>18.6 18.6 18.6</td>
<td>- 18.6 18.6 18.6</td>
<td></td>
</tr>
<tr>
<td>SSN [49]</td>
<td>44.0 32.4 27.9</td>
<td>18.6 18.6 18.6</td>
<td>- 18.6 18.6 18.6</td>
<td></td>
</tr>
<tr>
<td>BSN [49]</td>
<td>50.4 32.4 27.9</td>
<td>18.6 18.6 18.6</td>
<td>- 18.6 18.6 18.6</td>
<td></td>
</tr>
<tr>
<td>One (CNN backbone)</td>
<td>-</td>
<td>18.6 18.6 18.6</td>
<td>- 18.6 18.6 18.6</td>
<td></td>
</tr>
<tr>
<td>One (C3D backbone)</td>
<td>-</td>
<td>18.6 18.6 18.6</td>
<td>- 18.6 18.6 18.6</td>
<td></td>
</tr>
</tbody>
</table>

**Comparison with methods designed to detect actions.**

**References**