Online Action Tube Detection

The task is to determine what action is occurring in a video, and to localise its spatial and temporal extent.

- Action Tube: a set of linked bounding boxes spanning an individual action instance
- Online: the action tube should be constructed incrementally, on a frame by frame basis

Action Micro-tubes for Action Detection with Two frames

- Action micro-tube detection based on two frames separated by 3 frames, Saha et al. [1]
- Similarly, action tubules composed by 6 frames in Kalogeiton et al. [2]
- Micro-tubes are linked up to create whole action-tubes, based on Singh et al. [4]

The Problem with Anchors

- In both Saha et al. [7] and Kalogeiton et al. [8] anchors are cuboidal in shape – as a result, anchors for detection prediction may be wrong when detections significantly shift in time (a)
- Indeed, all possible associations are $O(n^2)$, as opposed to $O(n)$, where $n$ is the number of anchors per frame, and # is the number of considered frames
- Our proposal generator, instead, can flexibly explore the space of all micro-tubes (b)

Hidden Markov Models for Learning Anchor Transitions

Our Hidden Markov Model formulation:

- Observations: sequences $O = \{a_1, a_2, ...\}$ of ground truth bounding boxes $a_i$ in time
- Hidden State: anchor location $q = \{x, y, x\text{-width}, y\text{-height}\}$ within a pre-defined set (grid) $G$
- Transition matrix $A$: encodes the probability of transition from one anchor location to other in two successive frames
- Solved for HMM parameters (transition matrix $A$, means and covariances) via EM

As HMMs are inefficient to train with N=8732 states, we propose an approximation

- Compute matrices separately for each level of the feature pyramid in the SSD detector
- Count state transitions based on the highest IoU overlap, adding 1 in correspondence
- Normalise transition matrices row-wise

(a) Transition matrix for a $5 \times 5$ feature map grid for different values of $A$ (frame interval)
(b) Monte-Carlo sampling of transition hypotheses vs our anchor sampling scheme

Proposed Architecture (TraMNet)

- Replaces anchor cuboids with anchors sampled from the learnt Transition Matrix
- Network is configurable, as the configuration of the pooling layer (d) depends on the TM
- From the micro-tube predictions produced by TraMNet, we can build online tubes as in [6]

Effect of Training and Testing Gaps ($\Delta$) on UCF-101

- Performance drops with increase in testing gap, $4 \times$ speedup with $\Delta = 4$
- Performance is resilient to training step change: TraMNet can handle sparse annotations

Results on the UCF-24 dataset (24 classes, annotation every frame)

<table>
<thead>
<tr>
<th>Method</th>
<th>Train $\Delta$</th>
<th>Test $\Delta$</th>
<th>BOVW0.5</th>
<th>BOVW0.75</th>
<th>BOVW0.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-CNN [3]</td>
<td>NA</td>
<td>NA</td>
<td>41.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MB-TS [7]</td>
<td>NA</td>
<td>NA</td>
<td>73.5</td>
<td>02.7</td>
<td>07.3</td>
</tr>
<tr>
<td>Saha et al [6]</td>
<td>NA</td>
<td>NA</td>
<td>73.2</td>
<td>46.3</td>
<td>15.0</td>
</tr>
<tr>
<td>SSD [4]</td>
<td>NA</td>
<td>NA</td>
<td>73.2</td>
<td>46.3</td>
<td>15.0</td>
</tr>
<tr>
<td>AMTNet [1]</td>
<td>1.23</td>
<td>6.03</td>
<td>33.1</td>
<td>00.5</td>
<td>10.7</td>
</tr>
<tr>
<td>ACT [2]</td>
<td>1.1</td>
<td>76.2</td>
<td>49.2</td>
<td>19.7</td>
<td>23.4</td>
</tr>
<tr>
<td>Gu et al [8] [9]</td>
<td>NA</td>
<td>NA</td>
<td>59.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SSD-L with trimming</td>
<td>NA</td>
<td>NA</td>
<td>76.2</td>
<td>45.5</td>
<td>16.4</td>
</tr>
<tr>
<td>SSD-L</td>
<td>NA</td>
<td>NA</td>
<td>76.8</td>
<td>48.2</td>
<td>17.0</td>
</tr>
<tr>
<td>AMTNet-L</td>
<td>1.1</td>
<td>77.9</td>
<td>50.8</td>
<td>19.8</td>
<td>23.9</td>
</tr>
<tr>
<td>AMTNet-L</td>
<td>5</td>
<td>77.5</td>
<td>49.5</td>
<td>17.3</td>
<td>22.5</td>
</tr>
<tr>
<td>TraMNet (ours)</td>
<td>1</td>
<td>79.0</td>
<td>50.9</td>
<td>21.3</td>
<td>23.9</td>
</tr>
<tr>
<td>TraMNet (ours)</td>
<td>5</td>
<td>77.6</td>
<td>49.7</td>
<td>18.4</td>
<td>22.8</td>
</tr>
<tr>
<td>TraMNet (ours)</td>
<td>21</td>
<td>75.2</td>
<td>47.8</td>
<td>17.4</td>
<td>22.3</td>
</tr>
</tbody>
</table>

Results on Sparse DALY (10 classes, max 5 frames annotated per instance)

<table>
<thead>
<tr>
<th>Method</th>
<th>Untrimmed Videos</th>
<th>Trimmed Videos</th>
</tr>
</thead>
<tbody>
<tr>
<td>weissapfel et al. [5]</td>
<td>13.9</td>
<td>63.9</td>
</tr>
<tr>
<td>SSD-L without-trimming</td>
<td>0.611</td>
<td>61.5</td>
</tr>
<tr>
<td>SSD-L</td>
<td>14.6</td>
<td>56.2</td>
</tr>
<tr>
<td>AMTNet-L</td>
<td>3.121</td>
<td>63.9</td>
</tr>
<tr>
<td>AMTNet-L</td>
<td>3.144</td>
<td>62.0</td>
</tr>
<tr>
<td>TraMNet (ours)</td>
<td>3.134</td>
<td>67.0</td>
</tr>
</tbody>
</table>

Contributions

- Efficient and flexible anchor hypothesis generation framework (third figure (c) on the left)
- Number of anchors is based on the cardinality of the transition matrix, rather than $af$
- Allows using different anchors at train and test time to capture variability in the test data
- Handles significant spatial movement in dynamic actions
- Allows training on sparse annotations, unlike [6,8]

Effect of Training and Testing Gaps ($\Delta$) on UCF-101

- Performance drops with increase in testing gap, $4 \times$ speedup with $\Delta = 4$
- Performance is resilient to training step change: TraMNet can handle sparse annotations