Temporal-Relational CrossTransformers for Few-Shot Action Recognition

T. Perrett, A. Masullo, T. Burghardt, M. Mirmehdi, D. Damen
University of Bristol
CVPR 2021

CV/DL reading group on Transformers
April 6th 2021
Sofia Broomé
In a nutshell

• Few-shot action recognition approach building on tuple matching between query and support set

  • Constructs query-specific class prototypes using CrossTransformer method applied to video frames instead of image parts

  • Main result: SotA on action recognition few-shot benchmarks (Kinetics, Something-something, HMDB51 and UCF101)
Few-shot learning (action recognition setting) preliminaries

- You have an (action recognition) dataset with, say, 100 classes
- Train on e.g. 64 of those, validate on 12, test on 24 (i.e. the classes are new and unseen at test-time)
- For each class you have a small labeled support set (K videos, e.g. 5)
- Given a query video, match it to one of the classes
- Overall goal: Train the system to quickly (with few attempts) learn how to match a query to a support class
- Compute query-class distances and pass through cross-entropy loss

- During training, sample X tasks (64 choose 5 = 7M, 24 choose 5 = 42k), In this paper: X=10 000. Average the results at test time
What are CrossTransformers?

• "CrossTransformers: spatially-aware few-shot transfer" by C. Doersch, A. Gupta and A. Zisserman (NeurIPS 2020)

• Idea: Transformer attention between spatial locations in query image and spatial locations in support set images

(hence cross (?))
What are Temporal-Relational CrossTransformers?

• Apply CrossTransformer idea to video frames instead
• Sample *tuples* from query video and from all videos in support set, compute attention between these

• Flexible-length $\omega$ for tuples (although only 8 frames are sampled from videos here), so $\omega \leq 8$

• Different $\omega$ can be combined into $\Omega$ set for final classification
• One Transformer per $\omega$
• $\Omega = \{2, 3\}$ works best in this paper
How is this done concretely? For $\omega = 2$

- Query representation $Q_p$ for pair $p = (p_1, p_2)$
  
  $$Q_p = [\Phi(q_{p_1}) + \text{PE}(p_1), \Phi(q_{p_2}) + \text{PE}(p_2)] \in \mathbb{R}^{2 \times D}$$

- One pair $m = (m_1, m_2)$ from support set representation, similarly:
  
  $$S_{km}^c = [\Phi(s_{km_1}^c) + \text{PE}(m_1), \Phi(s_{km_2}^c) + \text{PE}(m_2)] \in \mathbb{R}^{2 \times D}$$

$\Phi$ is a CNN, $c$ is the class, $k$ is the video

**Whole support set for one class:**

$$S^c = \{S_{km}^c : (1 \leq k \leq K) \land (m \in \Pi)\}$$
How is this done concretely? For $\omega = 2$

• Will use these representations in a Transformer query/key/value setting

• To compute query-specific prototypes for each class $t^c_p$
How is this done concretely? For $\omega = 2$

• Calculate query-specific class prototypes (next slide), $t_p^c$
  and then: distance between prototype and query representation ($u_p$)

$$T(Q_p, S^c) = \| t_p^c - u_p \|$$

• Consider multiple pairs from query video, take average distance

$$Q = \{ Q_p : p \in \Pi \}$$

$$T(Q, S^c) = \frac{1}{|\Pi|} \sum_{p \in \Pi} T(Q_p, S^c)$$
Scaled dot-product attention to obtain $t_p^c$

- Query $\gamma$, key $\Gamma$ and value $\Lambda$ linear maps (shared across classes)
  Also query $\gamma$ and key $\Gamma$ are always shared within a class

$$\gamma, \Gamma : \mathbb{R}^{2 \times D} \rightarrow \mathbb{R}^{d_k} \quad \text{and} \quad \Lambda : \mathbb{R}^{2 \times D} \rightarrow \mathbb{R}^{d_v}$$

- Correspondence between query $p$ and support set tuple $km$, dot product between key and query:

$$a_{kmp}^c = (\Gamma \cdot L(S_{km}^c)) \cdot (\gamma \cdot L(Q_p))$$

$L$ is layer normalization.

- Finally, scaling and softmax:

$$\tilde{a}_{kmp}^c = \frac{\exp(a_{kmp}^c) / \sqrt{d_k}}{\sum_{l,n} \exp(a_{lnp}^c) / \sqrt{d_k}}$$
Scaled dot-product attention to obtain $t_p^c$ continued.

Apply same value-transform $\Lambda$ to query and support set tuple

$$\gamma, \Gamma : \mathbb{R}^{2 \times D} \rightarrow \mathbb{R}^{d_k} \quad \text{and} \quad \Lambda : \mathbb{R}^{2 \times D} \rightarrow \mathbb{R}^{d_v}$$

$$v_{km}^c = \Lambda \cdot S_{km}^c$$

$$t_p^c = \sum_{km} \tilde{a}_{km}^c v_{km}^c$$

$$T(Q_p, S^c) = \|t_p^c - u_p\|$$
Final version for all "cardinalities": \( T^\Omega \)

- Each TRX \( T^\omega \) includes query, key and value linear maps corresponding to the dimensionality of \( \omega \)

\[
\gamma^\omega, \Gamma^\omega : \mathbb{R}^{\omega \times D} \rightarrow \mathbb{R}^{d_k} \quad \text{and} \quad \Lambda^\omega : \mathbb{R}^{\omega \times D} \rightarrow \mathbb{R}^{d_v}
\]

- Accumulate distances from the different TRXs:

\[
T^\Omega(Q, S^c) = \sum_{\omega \in \Omega} T^\omega(Q^\omega, S^{c\omega})
\]

- Query is assigned to class with lowest \( T^\Omega \)
Experiments

• **Flagship result:** State-of-the-art on 5-way, 5-shot benchmarks

<table>
<thead>
<tr>
<th>Method</th>
<th>Kinetics</th>
<th>SSv2$^\dagger$</th>
<th>SSv2$^*$</th>
<th>HMDB</th>
<th>UCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMN [31]</td>
<td>78.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CMN-J [32]</td>
<td>78.9</td>
<td>48.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TARN [3]</td>
<td>78.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ARN [27]</td>
<td>82.4</td>
<td>-</td>
<td>-</td>
<td>60.6</td>
<td>83.1</td>
</tr>
<tr>
<td>OTAM [4]</td>
<td>85.8</td>
<td>-</td>
<td>52.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TRX (Ours)</td>
<td><strong>85.9</strong></td>
<td><strong>59.1</strong></td>
<td><strong>64.6</strong></td>
<td><strong>75.6</strong></td>
<td><strong>96.1</strong></td>
</tr>
</tbody>
</table>

Table 1: Results on 5-way 5-shot benchmarks of Kinetics (split from [32]), SSv2 ($^\dagger$: split from [32], $^*$: split from [4]), HMDB51 and UCF101 (both splits from [27]).
Experiments

• TRX with different $\Omega$ values

<table>
<thead>
<tr>
<th>Cardinalities</th>
<th>Num tuples</th>
<th>Kinetics</th>
<th>SSv2$^\dagger$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Omega={1}$</td>
<td>-</td>
<td>85.2</td>
<td>53.3</td>
</tr>
<tr>
<td>$\Omega={2}$</td>
<td>28</td>
<td>85.0</td>
<td>57.8</td>
</tr>
<tr>
<td>$\Omega={3}$</td>
<td>56</td>
<td>85.6</td>
<td>58.8</td>
</tr>
<tr>
<td>$\Omega={4}$</td>
<td>70</td>
<td>84.5</td>
<td>58.9</td>
</tr>
<tr>
<td>$\Omega={2, 3}$</td>
<td>84</td>
<td><strong>85.9</strong></td>
<td><strong>59.1</strong></td>
</tr>
<tr>
<td>$\Omega={2, 4}$</td>
<td>98</td>
<td>84.4</td>
<td>58.4</td>
</tr>
<tr>
<td>$\Omega={3, 4}$</td>
<td>126</td>
<td>85.3</td>
<td><strong>59.1</strong></td>
</tr>
<tr>
<td>$\Omega={2, 3, 4}$</td>
<td>154</td>
<td>85.3</td>
<td>58.9</td>
</tr>
</tbody>
</table>

Table 2: Comparing all values of $\Omega$ for TRX, noting the number of tuples for each model, given by $\sum_{\omega \in \Omega} |\Pi^\omega|$. 

• Skipping remaining experiments since they mostly are there to ensure that the method is behaving as desired
Conclusion for CV/DL group

• This was another example of how to apply Transformers-principles in a vision setting

• Videos are represented by ordered tuples of frames at different positions, allowing flexibility in what speeds and temporal offsets we match query with support set

• Quite thorough experimentation shows that the method seems to be doing what it should (qualitative examples, uses multiple support videos, etc)
Discussion points

• Thorough paper, following standard recipe for a CV-paper (currently popular task, slightly new idea, extensive experimentation with ablations, beats SotA, well-written)

• ”Temporality”: Representing videos, first by only 8 frames, and then with pairs and triplets, with frame-wise down-sampling

• Toy-flavor of few-shot task and datasets?
  • A robot that encounters unseen objects. How will it find 5 labels? (Usually N is small in N-way k-shot classification)
  • Videos are short snippets (temporally trimmed)

• Trained end-to-end with CNN. Missing fixed backbone ablation?
• Section 4.3.5: “Importantly, the method’s runtime scales linearly with the number of frames.”

Figure 8: SSv2$^\dagger$ accuracy (left y-axis) vs runtime analysis (right y-axis in seconds/task) for TRX $\Omega = \{2, 3\}$ as the number of sampled frames varies from 4 to 12 frames.
More discussion?