SSCAP: Self-supervised Co-occurrence Action Parsing for Unsupervised Temporal Action Segmentation

Zhe Wang, Hao Chen, Xinyu Li, Chunhui Liu, Yuanjun Xiong, Joseph Tighe, Charless Fowlkes
Amazon AWS

WACV WAIKOLOA, HI JAN 4-8

Introduction

Temporal action segmentation is a task to classify each frame in the video with an action label.

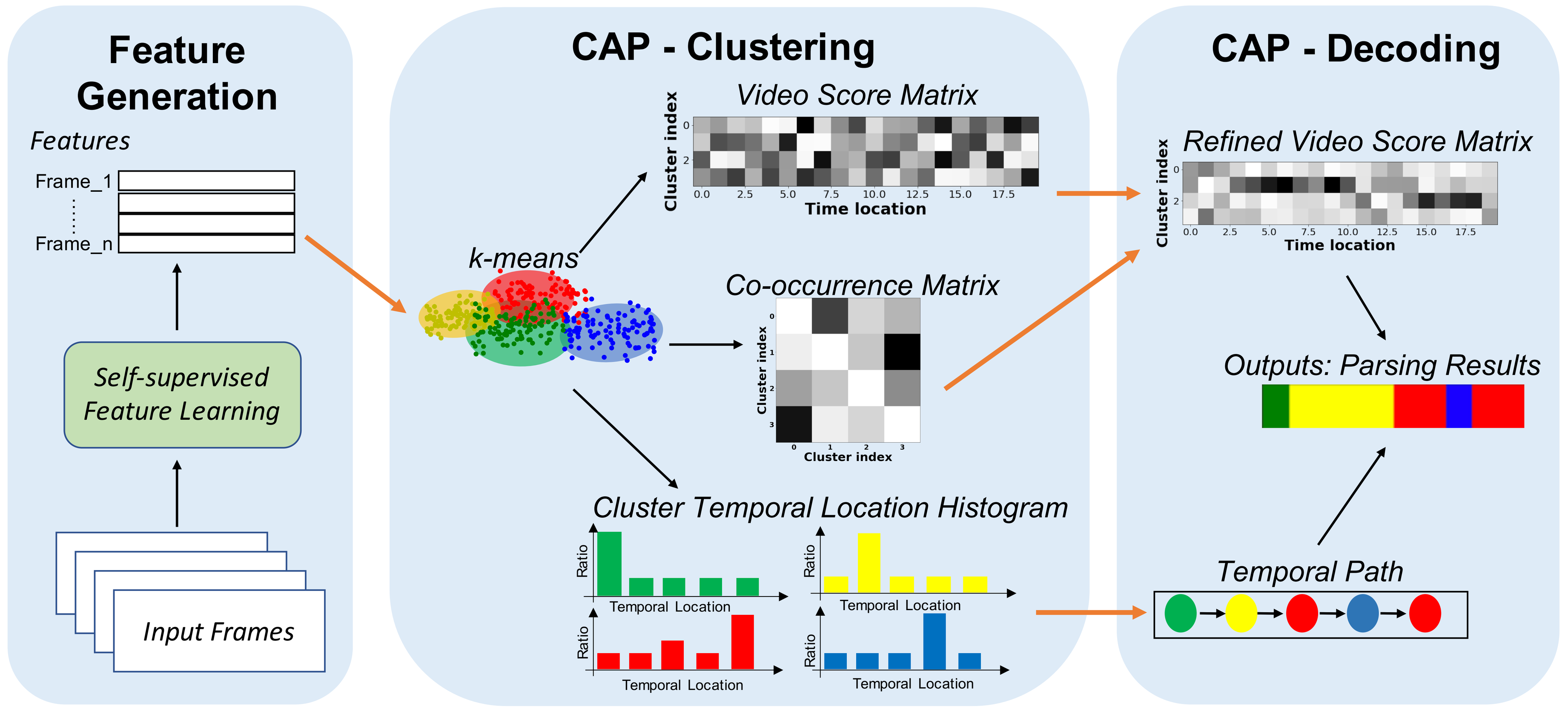
Full frame level supervision in untrimmed video is expensive
Weak supervision reduces the cost, but still heavily relies on non-trivial expertise in annotation

Can we do it in an ***unsupervised*** manner?

SSCAP – an unsupervised solution:

- **Uses Self-Supervised (SS) learning to extract features that are more temporal distinguishable;**
- **Uses Co-occurrence Action Parsing (CAP) to capture the correlations among sub-actions, and handle complex structures and recurrences of sub-actions;**
- **Achieves SOTA result on Breakfast, Salad, and FineGym (with more complex action structures), even outperforms weakly-supervised solutions.**

SSCAP



Algorithm 1: Generating the Co-occurrence Matrix and the Refined Video Score Matrix.

Data: Video score matrix S_m for video X_m, with $S_m[k,n]_{k=1,n=1}^{K,N_m}$, number of clusters: K.

Result: Co-occurrence matrix: $P(i,j)_{i,j=1}^{K}$, Refined video score matrix R_m, with scores $R_m[k,n]$.

Generate Co-occurrence Matrix

- Iterate all the videos, count the times each cluster appears $C(i)_{i=1}^{K}$, and the times different clusters co-occur together $C(i,j)_{i,j=1}^{K}$. Normalize to make it as conditional probability $P(j|i) = C(i,j)/C(i)$

Refined Video Score Matrix

- Initialization: $\mathcal{G} \leftarrow k_0$ (k_0 is the cluster with the largest ratio of frames $r(k_0)$ in current video).

- $k^* = k_0$

while $len(\mathcal{G}) \leq K$ *and* $r(k^*)>0$ **do**

1. For each remaining cluster $j \notin \mathcal{G}$:
 - update the video score matrix conditioned on the previous selected cluster k^*:
 $R_m[j,n] = P(j|k^*) \cdot S_m[j,n]$
2. Select the next cluster: $k^* \leftarrow \arg\max_j r(j)$.
3. Update: $\mathcal{G} \leftarrow \mathcal{G} \cup \{k^*\}$

end

- Return: P, R_m.

Feature Generation

Self-supervised learning:

- **SpeedNet[1,2]**: predict different frame rates;
- **ShuffleLearn[3]**: predict whether a clip is shuffled or not;
- **RotationNet[4]**: predict the degrees of the rotation.

[1] Dave Epstein et. al. "Oops! predicting unintentional action in video". In CVPR, 2020.
[2] Sagie Benaim et. al. "Speednet: Learning the speediness in videos". In CVPR, 2020.
[3] Ishan Misra et. al. "Shuffle and learn: Unsupervised learning using temporal order verification". In ECCV, 2016.
[4] Spyros Gidaris et. al. "Un-supervised representation learning by predicting image rotations". In ICLR, 2018.

CAP - Clustering

Clustering: cluster all frame-level features using k-means;

Video Score Matrix $S \in \mathbb{R}^{K \times N}$ **:**

- Capture the score of each frame belonging to a cluster, $s_{(n,k)} = p(x_n|k) = \mathcal{N}(x_n; \mu_k, \Sigma_k)$

Co-occurrence Matrix $C \in \mathbb{R}^{K \times K}$ **:**

- Capture the correlations among sub-actions based on the times they co-occur

Cluster Temporal Location Histogram $H(t_n, k)$:

- Estimate when each cluster happens in a video;
- $t_n = \frac{n}{N}$, the relevant timestamp in the video;
- For recurrence of the sub-action, the histogram contains multiple significant bins

CAP - Decoding

Refined Video Score Matrix $R \in \mathbb{R}^{K \times N}$ **:**

- Is refined using the co-occurrence matrix;
- Capture correlation information among sub-actions and global structures;
- Carefully select clusters k to avoid over-segmenting the video

Temporal Path Estimation and Decoding:

- Select top-K bins from temporal location histogram for cluster, and then concatenate all of them into an ordered sequence;
- Capture the multi-occur sub-actions;
- Decoding: Viterbi algorithm[1]

[1] T. Quach and M Farooq. Maximum likelihood track formation with the Viterbi algorithm. In IEEE Conference on Decision and Control, 1994.

Experimental Results

Comparing to SOTA

Breakfast	MoF	F1 score
Unsupervised setting		
GMM [48]	0.346	-
LSTM + AL [1]	0.429*	-
CTE [30]	0.418	0.264
VTE-UNET [53]	0.481	-
ASAL [35]	0.525	0.379
Our SSCAP	0.511	**0.392**
Weakly-supervised setting		
Action Sets [45]	0.284	-
NNviterbi [46]	0.430	-
SCT [16]	0.304	-
SetViterbi [34]	0.408	-
EnergySeg [33]	0.630	-
Fully-supervised setting		
HTK [28]	0.259	-
GTRM [24]	0.650	-
MS-TCN [15]	0.663	-
BCN [60]	0.704	-

50Salads	MoF	F1 score
Unsupervised setting		
LSTM + AL [1]	0.606*	-
CTE [30]	0.355	-
VTE-UNET [53]	0.306	-
ASAL [35]	0.392	-
Our SSCAP	**0.414**	**0.303**
Weakly-supervised setting		
NNviterbi [46]	0.494	-
EnergySeg [33]	0.547	-
Fully-supervised setting		
HTK [28]	0.247	
GTRM [24]	0.826	-
MS-TCN [15]	0.734	-
BCN [60]	0.844	-

FineGym	MoF	F1 score
Baseline [30]	0.294	0.167
Our SSCAP	**0.666**	**0.297**

- Achieves SOTA on both Breakfast and 50Salads, even outperforms weakly-supervised solutions;
- Achieves significant improvement on FineGym, demonstrating the effectiveness in handling videos with more complex structures.

Visualization

SSCAP suppresses the over-segmentation issue using the co-occurrence of sub-actions, and improves the segmentation quality.

Ablation #1 – Self-supervised Features

	Feature Description	MoF	F1
Baseline			
(a)	IDT [30]	0.316	-
(b)	K400 I3D [30]	0.251	-
(c)	CTE [30]	0.418	0.264
Self-supervised on K400			
(d)	K400 SpeedNet	**0.508**	**0.391**
(e)	K400 RotationNet	0.328	0.317
(f)	K400 shuffleLearn	0.339	0.328
Self-supervised on Breakfast			
(g)	Breakfast SpeedNet	0.344	0.327
(h)	Breakfast RotationNet	0.307	0.319
(i)	Breakfast shuffleLearn	0.315	0.309
Self-supervised first on K400, then on Breakfast			
(j)	K400, Breakfast, SpeedNet	0.501	0.337
(k)	K400, Breakfast, RotationNet	0.279	0.290
(l)	K400, Breakfast, shuffleLearn	0.292	0.318

1. Self-supervised features always perform better than the I3D feature;

2. RotationNet performs worse than SpeedNet and ShuffleLearn, indicating that self-supervised from temporal augmentation is important;

3. SpeedNet performs the best;

4. Larger dataset like Kinetics can help build better self-supervised representation;

5. It's not needed to use target dataset to get a good feature representation.

Ablation #2 – Module Design

Dataset	SS	C-Matrix	M-T-Path	MoF	F1
Breakfast				0.418	0.264
	✓			0.508	0.391
	✓	✓		0.511	0.392
	✓	✓	✓	**0.511**	**0.392**
50Salads				0.355	-
	✓			0.372	0.281
	✓	✓		0.378	0.290
	✓	✓	✓	**0.414**	**0.303**
FineGym				0.294	0.167
	✓			0.425	0.246
	✓	✓		0.442	0.248
	✓	✓	✓	**0.666**	**0.297**

- ***Self-supervised learning (SS)*** always helps;
- ***Co-occurrence matrix (C-Matrix)*** always helps, while on FineGym the improvement is more notable, indicating the importance of it in handling more complex scenarios;
- ***Multi-occur temporal path (M-T-Path)*** helps 50Salads and FineGym, but not Breakfast, as most of the sub-actions only occur once in Breakfast. The improvement on FineGym is significant.

Contact: Hao Chen (hxen@amazon.com), Zhe Wang (zwang15@uci.edu)