Leveraging triplet loss for unsupervised action segmentation

Elena Belén Bueno-Benito ⋆ ebueno@iri.upc.edu
Biel Tura Vecino*† bieltura@amazon.co.uk
Mariella Dimiccoli ⋆ mdimiccoli@iri.upc.edu

⋆ Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain
† Amazon Alexa AI, UK, Cambridge

Abstract

In this paper, we propose a novel fully unsupervised framework that learns action representations suitable for the action segmentation task from the single input video itself, without requiring any training data. Our method is a deep metric learning approach rooted in a shallow network with a triplet loss operating on similarity distributions and a novel triplet selection strategy that effectively models temporal and semantic priors to discover actions in the new representational space. Under these circumstances, we successfully recover temporal boundaries in the learned action representations with higher quality compared with existing unsupervised approaches. The proposed method is evaluated on two widely used benchmark datasets for the action segmentation task and it achieves competitive performance by applying a generic clustering algorithm on the learned representations. [1]

1. Introduction

Unconstrained videos capturing real-world scenarios are usually long, untrimmed and contain a variety of actions which can be effortlessly divided by a human observer into semantically homogeneous units. The task of action segmentation, which we target in this work, is the process of identifying the boundaries of an action, i.e. *pour water*, in an untrimmed video of an activity, i.e. *making tea*, even when temporally adjacent actions may have very small visual variance between them. This problem has been traditionally tackled through supervised learning approaches [9, 13]. More recently, weakly-supervised and semi-supervised approaches have shown to be an effective way to reduce the labelling effort [6, 14, 15, 19]. However, these approaches are still data-hungry and computationally expensive. Unsupervised approaches have developed following two different research lines [1, 11, 12]. Most of them focus on grouping actions across videos and rely on the use of activity labels [10, 18, 20, 22], therefore putting more emphasis on the quality of the representation. A few of them, the most computationally efficient, act on a single video to recover clusters [16] or detect temporal boundaries [7] and do not require any manual annotation.

Our approach follows this latter research line. We assume that the atomic actions can effectively be modelled as clusters in an underlying representational space and we propose a novel framework that maps the initial feature space of a video into a new one, where the temporal-semantic clusters corresponding to atomic actions are unveiled. Similarly to other unsupervised approaches that rely on similar assumptions [10], our focus is on representation learning. However, we operate at video level instead of activity level. Similarly to other fully unsupervised approaches [7, 16], our method has considerable practical advantages for downstream applications since it can be in principle applied to any video no matter the dataset it belongs to nor if there exist videos having a similar temporal structure, as well as being more reliable.

Our technical contribution is a novel approach to action representation learning that uses a shallow network and a triplet loss operating on similarity distributions with a novel triplet selection strategy based on a downsampled temporal-semantic similarity weighting matrix. Our approach outperforms the state-of-the-art in action segmentation on *Breakfast* and *Youtube INRIA Instructional* benchmark datasets.

2. Related work

Fully supervised approaches. Action Segmentation has been traditionally tackled as a supervised learning problem, where existing approaches differ mainly in the way temporal information is taken into account [21]. Traditional supervised methods for action segmentation require significant amounts of labelled data for training, which restricts their applicability beyond pre-segmented datasets in large-scale domains [9, 13].

Weakly and semi-supervised approaches. To alleviate the need for large annotated datasets, weakly supervised techniques for video segmentation involve using transcripts, visual similarities, and audio information to generate pseudo-labels for training [8]. Some approaches use machine learn-

*Work done during an internship at the IRI.

[1] https://github.com/elenabbbuenob/TSA-ActionSeg

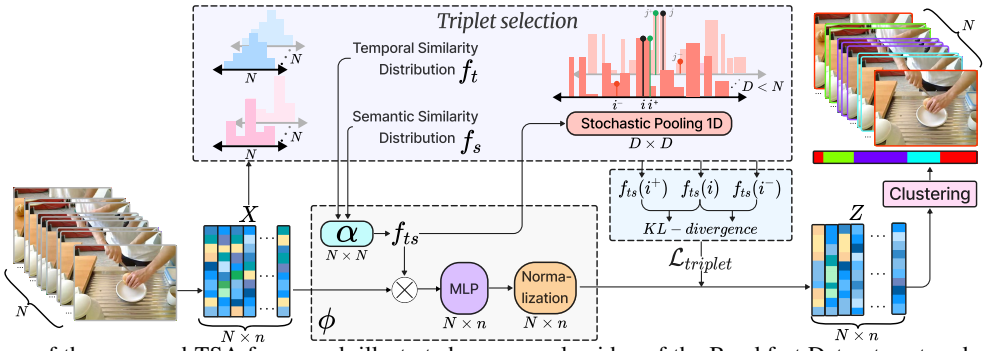


Figure 1. Overview of the proposed TSA framework illustrated on a sample video of the Breakfast Dataset: network architecture transforming the initial features X into the learned features Z through a shallow network with a novel triplet selection strategy and a triplet loss based on similarity distributions.

ing models to infer the segments of the video [14]. Other approaches, such as those based on frame-to-frame visual similarities, self-attentions mechanism [15] or iterative soft boundary assignment [6], enforce consistency between the video and labels without the need for temporal supervision. Recent work has proposed a semi-supervised approach that uses unsupervised and supervised training to improve performance [5, 19]. These methods only apply to videos with transcripts and cannot be extended to unconstrained videos.

Unsupervised learning approaches. Unsupervised learning approaches typically learn action representation in a self-supervised fashion and then apply a clustering algorithm to obtain the action segmentation (assuming that the number of clusters is known). Some methods solve the temporal action relations globally over a collection of videos of the same activity [10–12, 18, 22]. Even if these approaches do not require labelled data, they are data-hungry and are not suitable for transferring the learned knowledge to a dataset with a different distribution and they demonstrated modest performance for the task of action segmentation at the video level. In contrast, we aim at unveiling the clusters underlying a single video. There is limited work in the literature on learning action representations in a self-supervised manner within a single video. [1] proposes a model based on an encoder LSTM architecture with Adaptive Learning (LSTM+AL) that minimizes the prediction error of future frames and assigns segmentation boundaries based on the prediction error of the next frame. Recent works suggested learning event representations [3] and graph structures from a single sequence (DGE) [4], but only low-temporal resolution image sequences have been used for testing.

Fully unsupervised approaches. Clustering methods, which generate a partition of the input data based on a specific similarity metric, have been poorly investigated within the field of action segmentation. However, very recent work [16] has shown that simple clustering approaches, i.e. K-means, are instead a strong baseline for action segmentation. They hence proposed a new clustering approach called Temporally-Weighted FINCH (TW-FINCH), which is similar in spirit to the clustering approach named FINCH [17] but takes into account temporal proximity in addition to semantic similarity. Recently, [7] proposed to detect action boundaries (ABD) by measuring the similarity between adjacent frames based on the insight that actions have internal consistency within and external discrepancy across actions. We based our approach on the same insight that we modelled via a deep metric learning approach.

3. Methodology

We assume that the representational clustering grounding action segmentation encodes both temporal and semantic similarity, based on two observations: (*i*) temporal adjacent frames are likely to belong to the same action. (*ii*) frames corresponding to the same action (but not necessarily temporal adjacent) should have similar representation, encoding the common underlying semantic. Formally, let $X \in R^{N \times n}$ denote the matrix of n-dimensional feature vectors for a given sequence of N frames. We aim at learning a parametric function ϕ such that given the input feature matrix X, new Temporal-Semantic Aware (TSA) representations $Z \in \mathbb{R}^{N \times n}$ are obtained as $Z = \phi(X)$.

Triplet loss and triplet selection. To learn ϕ, we minimize a triplet loss function that implements an original approach to select the triplets appropriately by relying on temporal-semantic similarity distributions f_{ts} obtained as the weighted sum of the temporal and the semantic similarity distributions, say f_t and f_s, $f_{ts} = \boldsymbol{\alpha} \cdot f_t + (1 - \boldsymbol{\alpha}) \cdot f_s$, where $\boldsymbol{\alpha} \in [0, 1]^{N \times 1}$ a vector of learning parameters of the function ϕ.

To define $f_{\mathcal{S}}$, we assume that the set of most similar frames in the original feature space of an anchor i is very likely to be part of the same action. The self-similarity of an anchor i to all other frames is defined element-wise via a pairwise similarity, upon normalization to the total unit weight, $f_{\mathcal{S}} = w_{ij}/W$, with $W = \sum_{i,j) \in E} w_{ij}$ and $w_{ij} = \exp(-(1 - d(x_i, x_j))/h)$, and where E is the set of pairwise relations, $d(\cdot, \cdot)$ is the cosine distance and h is the filtering parameter of the exponential function. The pairwise similarities are normalized to represent joint probability distributions between pairs of elements in the sequence.

To define f_t, we assume that as we move away from the anchor i, the likelihood of a feature vector $x_{j\neq i}$ to represent the same action as frame i decreases. To model this behaviour, we define a weight function $w(\cdot)$ that depends on the temporal frame distance d from the given frame as $w(d) = -1 + 2\exp(-\frac{1}{\beta}d)$ where β is a constant that controls the slope of the weight function and d is the temporal distance between frames. By imposing that $w(L/2) = 0$, and then solving for β, we get that the constant β can be expressed in terms of the positive window length, that is: $\beta = -L/(2\ln(\frac{1}{2}))$.

The temporal and semantic matrices are downsampled to reduce computational costs using stochastic pooling during the training. An anchor index is randomly selected from the set of downsampled indices $i \in D$. Its set of positive samples $\mathcal{P}_i$ is taken as the 5% of the frames with the highest similarity values in i-row of the temporal-semantic affinity matrix f_{ts}. We define the negative set $\mathcal{N}_i$ as the frames whose i-row f_{ts} is between the mean and the sum of the mean and standard deviation of the similarity metric. The triplet loss is defined as:

$$\mathcal{L}_{triplet} = \frac{1}{D}\sum_{i\in D}\max(0,\ KL(f_{ts}(i)||f_{ts}(i^-)) - KL(f_{ts}(i)||f_{ts}(i^+)) \quad (1)$$

where KL represent the KL-divergence of the temporal-semantic similarity distribution f_{ts}. For each loss term, given an anchor index $i \in D$ with $D < N$, we define the triplet $\{i,\ i^+,\ i^-\}$ where $i^+ \in \mathcal{P}_i$ and $i^- \in \mathcal{N}_i$ that they are the sets of positive and negative indices, respectively.

Model architecture. We used a shallow neural network consisting in our case of a multi-layer perceptron with a single hidden layer followed by a ReLu activation function. This makes our approach (Figura 1) easy to train and more suited for practical applications than existing approaches consisting of multiple convolutional layers and/or recurrent networks. Empirical experiments showed that using a single hidden layer was easier and faster to train than deeper models while achieving similar performance. The results are also invariant to the number of units in the hidden layer.

4. Experimental evaluation

Input features. We use the same datasets and input features for the frame-level initial representations as in [1,7,10, 16, 18, 20]. For BF, we use the Improved Dense-Trajectory (IDT) features. For YII, we use a set of frame-level representations given by a concatenation of HOF descriptors and features extracted from the VGG16-*conv5* network.

Model training. The parameter L used in this paper is the average number of action classes for a specific dataset, being 6 and 9 for BF and YII, respectively. Empirical experiments showed that using a single hidden layer was easier and faster to train than deeper models while achieving similar performance. The reported results are also invariant to the number of units in the hidden layer. The architecture used to obtain our features is a multi-layer perceptron with as many units as the input feature dimensionality, n, although this could be changed to obtain the desired output dimensionality. The batch size is equivalent to the downsampling and the number of batches will be the quotient of the number of frames and the batch size. We define the distance hyperparameter as the minimum threshold ε that the difference of the last two losses should take. This hyperparameter is set to track early stops with a patience of 2 times. The minimum and maximum number of training epochs are fixed at 2 and 50, respectively. The initial learning rate depends on each dataset and follows an exponential learning decay rate of 0.3 and a weight decay L_2 of 10^{-3} as the regularisation parameter.

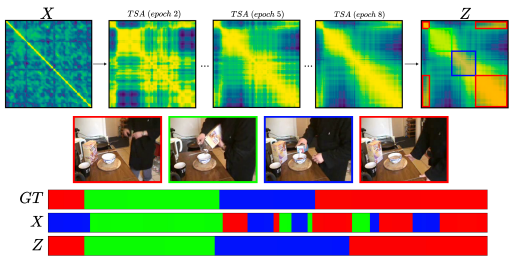


Figure 2. Example of training and result obtained by using TSA. **(a)** Cosine similarity affinity matrix for X and evolution of Z for different training epochs. Actions are highlighted as neighbour communities referring to the segmentation of the video when a clustering algorithm is applied. **(b)** Segmentation plots showing the ground truth, X and Z.

Model study. The results of the training process are visualized in Figure 2, which shows how the initial features change to uncover clusters in the new representational space. As the training progresses, the clusters become more transparent and more visible along the diagonal. To obtain the final segmentation, three different clustering algorithms were applied: K-means, Spectral clustering, and FINCH [17]. The performance by our method with or without a layer combination of f_t and f_s (being f_{ts} when both are marked) is also compared in Table 2.

Experimental results. We report the results of existing unsupervised methods for comparison, by applying Hungarian matching at the video-level [2]. We report three widely used metrics: (*i*) accuracy of the segmentation and action identification, computed as the *Mean over Frames (MoF)* metric. (*ii*) Similarity and diversity of the predicted segments, calculated as the *Intersection over Union (IoU)* metric. (*iii*) The *F1-score* computed across the predicted segments and the known ground truth to evaluate the quality of the action segmentation. We used the code made publicly available by the authors [2] to compute the performance of DGE on the considered datasets, since this approach, similar to

[2]https://github.com/mdimiccoli/DGE

Breakfast Action Dataset				
Baselines	**MoF**	**IoU**	**F1**	**T**
Equal Split*	34.8	21.9	-	✗
Spectral*	55.5	44.6	-	✗
Kmeans*	42.7	23.5	-	✗
FINCH* [17]	51.9	28.3	-	✗
Unsupervised				
LSTM+AL [1]	42.9	46.9	-	✓
VTE [20]	52.2	-	-	✓
DGE* [4] (Kmeans)	58.8	47.8	51.6	✗
DGE* [4] (Spectral)	59.5	48.5	51.7	✗
TW-FINCH [16]	62.7	42.3	49.8	✗
ABD [7]	64.0	-	52.3	✗
Ours* (Kmeans)	63.7	**53.3**	**58.0**	✗
Ours* (Spectral)	63.2	52.7	57.8	✗
Ours* (FINCH)	**65.1**	52.1	54.6	✗

Youtube INRIA Instructional Dataset			
Baselines	**F1**	**MoF**	**T**
Equal Split*	27.8	30.2	✗
Spectral*	44.6	55.1	✗
K-means*	29.4	38.5	✗
FINCH* [17]	35.4	44.8	✗
Unsupervised			
LSTM+AL [1]	39.7	-	✓
DGE* [4] (Kmeans)	47.0	42.1	✗
DGE* [4] (Spectral)	48.9	44.8	✗
TW-FINCH [16]	48.2	56.7	✗
ABD [7]	49.2	**67.2**	✗
Ours* (Kmeans)	**55.3**	59.7	✗
Ours* (FINCH)	54.7	62.4	✗

Table 1. Action Segmentation results on the BF and YII dataset. T denotes whether the method has a training stage on target activity/videos. The dash indicates "not reported". * denotes results computed by ourselves. The best and second-best results are marked in bold and underlined, respectively

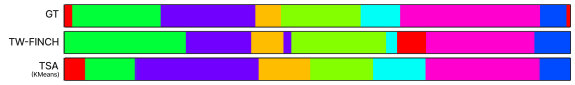
		BF (kmeans)		**BF** (FINCH)		**YII** (kmeans)		**YII** (FINCH)	
f_t	f_s	**F1**	**MoF**	**F1**	**MoF**	**F1**	**MoF**	**F1**	**MoF**
✗	✓	38.6	44.4	35.3	49.0	46.8	52.8	44.1	53.7
✓	✗	57.7	63.5	54.0	64.6	54.8	59.4	53.5	62.2
✓	✓	**58.0**	**63.7**	**54.6**	**65.1**	**55.3**	**59.7**	**54.7**	**62.4**

Table 2. Ablation study the BF and YII datasets, showing the importance of modelling both temporal and semantic information.

ours, computes a video representation suitable for the task of temporal/action segmentation.

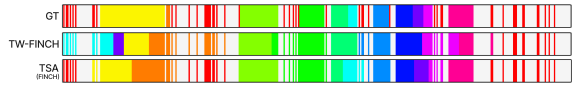
The left-hand table 1 reports the resulting metrics for the BF dataset, obtained with a learning rate 0.051, distance 0.032 and batch size 128. Our method significantly outperforms all other existing approaches. Special attention on F1, which is considerably better in our results, which tells us better quality and less over-segmentation in our method. These results are consistent with all three clustering approaches considered for obtaining the final segmentation of our learned features. We can therefore conclude that TSA outperforms SoTA approaches for the downstream task of action segmentation. Examples of segmentation results on a few videos for this dataset can be seen in Figure 2 (b) and Figures 3 (a)-(b).

The right-hand table 1 reports the resulting metrics for the YII dataset, obtained with a learning rate 0.403, distance 0.892 and batch size 12. This dataset is particularly challenging because of the nature of the annotations, where most of the frames in each video are labelled as background frames. To enable direct comparison, we follow the same procedure used in previous work [7, 16, 18] and report results by removing the ratio ($\tau = 75\%$) of the background frames from the video sequence and then report the performance. To capture the temporal dependencies in the time window, we compute the temporal similarity matrix before subtracting a ratio of the background frames. Our method improves the best *F1* metrics from the literature with a large margin, which indicates the quality of the segmentation in our method on both datasets, as the MoF does not reflect the quality, especially when the whole sequence is dominated by some very long segments. A segmentation output sample of our method for this dataset is plotted in Figures 3 (c). Also for this dataset, our results are consistent with all two clustering approaches.



(a) *P45_Scrambleegg,* BF - TSA $(77.7, 72.5)$ and TW-FINCH $(77.9, 68.1)$.

(b) *P18_Friedeggs,* BF - TSA $(90.8, 85.6)$ and TW-FINCH $(73.7, 61.9)$.



(c) *Changing_Tire_02,* YII - TSA $(76.1, 74.5)$ and TW-FINCH $(57.4, 53.2)$.

Figure 3. Segmentation output comparisons on two sample videos from BF and YII. Each caption shows the name of the video and the results (x, y) which are (MoF, F1) for each example.

5. Conclusions

This paper introduced a novel fully unsupervised approach for learning action representations in complex activity videos that solely operates on a single unlabelled input video. Our key contributions are a shallow architecture and a triplet-based loss with a triplet-based selection mechanism based on similarity distribution probabilities to model temporal smoothness and semantic similarity within and across actions. Experimental results on the BF and the YII datasets demonstrated that the learned representations, followed by a generic clustering algorithm, achieve SoTA performance. Furthermore, it has the advantage of not requiring human annotations, is easy to train and does not present domain adaptation issues. Future work will focus on improving the representation and making it more general to match videos at an activity-level.

References

[1] Sathyanarayanan N. Aakur and Sudeep Sarkar. A perceptual prediction framework for self supervised event segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 4

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4575–4583, 2016. 3

[3] Catarina Dias and Mariella Dimiccoli. Learning event representations by encoding the temporal context. In *Workshops Computer Vision (ECCV)*, volume 11131, pages 587–596, 2018. 2

[4] Mariella Dimiccoli and Herwig Wendt. Learning event representations for temporal segmentation of image sequences by dynamic graph embedding. *IEEE Transactions on Image Processing*, 30:1476–1486, 2021. 2, 4

[5] Guodong Ding and Angela Yao. Leveraging action affinity and continuity for semi-supervised temporal action segmentation. In *European Conference Computer Vision (ECCV)*, 2022. 2

[6] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6508–6516, 2018. 1, 2

[7] Zexing Du, Xue Wang, Guoqing Zhou, and Qing Wang. Fast and unsupervised action boundary detection for action segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3313–3322, 2022. 1, 2, 3, 4

[8] Mohsen Fayyaz and Jürgen Gall. Sct: Set constrained temporal transformer for set supervised action segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 498–507, 2020. 1

[9] Hilde Kuehne, Juergen Gall, and Thomas Serre. An end-to-end generative framework for video segmentation and recognition. In *Winter Applications for Computer Vision(WACV)*, 2016. 1

[10] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. Unsupervised learning of action classes with continuous temporal embedding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12066–12074, 2019. 1, 2, 3

[11] Sateesh Kumar, Sanjay Haresh, Awais Ahmed, Andrey Konin, M. Zeeshan Zia, and Quoc-Huy Tran. Unsupervised action segmentation by joint representation learning and online clustering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20174–20185, 2022. 1, 2

[12] Jun Li and Sinisa Todorovic. Action shuffle alternating learning for unsupervised action segmentation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12623–12631, 2021. 1, 2

[13] Shi-Jie Li, Yazan AbuFarha, Yun Liu, Ming-Ming Cheng, and Juergen Gall. Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[14] Zijia Lu and Ehsan Elhamifar. Weakly-supervised action segmentation and alignment via transcript-aware union-of-subspaces learning. In *International Conference on Computer Vision (ICCV)*, pages 8085–8095, 2021. 1, 2

[15] Yan Bin Ng and Basura Fernando. Weakly supervised action segmentation with effective use of attention and self-attention. *Computer Vision and Image Understanding*, 213:103298, 2021. 1, 2

[16] M. Saquib Sarfraz, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. Temporally-weighted hierarchical clustering for unsupervised action segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11225–11234, 2021. 1, 2, 3, 4

[17] M. Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. Efficient parameter-free clustering using first neighbor relations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2019. 2, 3, 4

[18] Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8368–8376, 2018. 1, 2, 3, 4

[19] Dipika Singhania, Rahul Rahaman, and Angela Yao. Iterative contrast-classify for semi-supervised temporal action segmentation. In *Conference on Artificial Intelligence (AAAI)*, 2022. 1, 2

[20] Rosaura G. VidalMata, Walter J. Scheirer, Anna Kukleva, David D. Cox, and Hilde Kuehne. Joint visual-temporal embedding for unsupervised learning of actions in untrimmed sequences. In *Winter Conference on Applications of Computer Vision (WACV)*, pages 1237–1246, 2021. 1, 3, 4

[21] Dong Wang, Di Hu, Xingjian Li, and Dejing Dou. Temporal relational modeling with self-supervision for action segmentation. In *Conference on Artificial Intelligence (AAAI)*, pages 2729–2737, 2021. 1

[22] Zhe Wang, Hao Chen, Xinyu Li, Chunhui Liu, Yuanjun Xiong, Joseph Tighe, and Charless Fowlkes. Sscap: Self-supervised co-occurrence action parsing for unsupervised temporal action segmentation. In *Winter Conference on Applications of Computer Vision (WACV)*, 2022. 1, 2