A Cluster-Based Method for Action Segmentation Using Spatio-Temporal and Positional Encoded Embeddings

Anonymous Author(s)

ABSTRACT

A crucial task to overall video understanding is the recognition and localisation in time of different actions or events that are present along the scenes. To address this problem, *action segmentation* must be achieved. Action segmentation consists of temporally segmenting a video by labeling each frame with a specific action. In this work, we propose a novel action segmentation method that requires no prior video analysis and no annotated data. Our method involves extracting spatio-temporal features from videos in samples of 0.5s using a pre-trained deep network. Data is then transformed using a positional encoder and finally a clustering algorithm is applied with the use of a silhouette score to find the optimal number of clusters where each cluster presumably corresponds to a different single and distinguishable action. In experiments, we show that our method produces competitive results on *Breakfast* and *Inria Instructional Videos* dataset benchmarks.

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; *Cost-sensitive learning*.

KEYWORDS

Action segmentation, Action recognition, Positional encoding

ACM Reference Format:

Anonymous Author(s). 2018. A Cluster-Based Method for Action Segmentation Using Spatio-Temporal and Positional Encoded Embeddings. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In recent years, video streaming platforms and services have been growing immensely. A Cisco report[1] suggests that by 2021 approximately 80% of the Internet's traffic would be made by video. To effectively extract information from this massive amount of data, better *video understanding* methods are needed. In particular, a crucial task of video understanding is the recognition and localisation in time of different actions or events that are present along the scenes [26, 32] in the so called *action recognition* task, which has attracted much of the researchers attention nowadays. Initial efforts focused on the classification of trimmed videos with a single action [5, 11, 36, 41]. Early methods were primarily based on the extraction of hand-crafted features [41] but, more recently, deep learning methods became end-to-end learning models with automatic feature extraction [5, 11, 36], achieving state-of-the-art results.

Since in real-life situations, videos are not always trimmed, the research community started to address the complex problem of *action segmentation*. Action segmentation consists of temporally segmenting a video by labeling each frame with a specific action. The performance achieved by fully supervised methods for this task is encouraging. Nevertheless, these solutions require frame-level or scene-level annotations that are incredibly laborious. For this reason, researchers started focusing on methods with less supervision, such as weakly-supervised [4, 22, 29, 37] and unsupervised methods [23, 33, 40].

Nevertheless, with the exception of the recent work from Sarfraz et al. [33], in which they view temporal action segmentation as a grouping problem and propose a new clustering algorithm that can generate segmentation for a single video, most unsupervised or weakly-supervised methods for action segmentation rely on datasets with many examples of the selected activities.

In this work, we propose a novel segmentation method that requires no prior video analysis and no annotated data. Our method involves extracting spatio-temporal features from videos in samples of 0.5 s, using a pre-trained deep network. Data is then transformed using a positional encoder and finally, a clustering algorithm is applied with the use of a silhouette score to find the optimal number of clusters. On the final result, each produced cluster presumably corresponds to a different single and distinguishable action. In experiments, we show that our method produces competitive results on *Breakfast*[2] and *Inria Instructional Videos*[3] datasets benchmark.

This remainder of this paper is organized as follows. Section 2 summarizes how some recent related works have been successfully applying DL-based methods for action recognition and segmentation. In Section 3 we introduce our unsupervised method for action segmentation, followed by Section 4 where we describe the experiments conducted to evaluate the effectiveness of our proposal. Section 5 is devoted to our final remarks and conclusions.

2 RELATED WORK

The action recognition methods strongly rely on feature extraction. Therefore, we first summarize modern methods for video feature extraction in section 2.1). Then, we present works with focus on the action recognition task in section 2.2) followed by works on temporal action segmentation in section 2.3).

[1] https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[2] https://serre-lab.clps.brown.edu/resource/breakfast-actions-dataset

[3] https://hal.inria.fr/hal-01171193

Webmedia '21, November 05–12, 2021, Belo Horizonte, MG

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

2.1 Video feature extraction

Unlike images, videos have not only visual but also audible and temporal information. Current methods for video classification are generally divided into two stages: (1) Convolutional Neural Networks (CNNs) (that within this context are called *backbones*) are used to extract the audio-visual features from the video content; (2) after the feature-extraction, sophisticated models for aggregation, such as NetVLAD [3] and NetFV [27], can be applied to undermine audio-visual features and perform classification.

To extract the visual features from video, CNNs (e.g. Inception [38], ResNet [13]) pre-trained in the *dataset* ImageNet [8] can be used. Analogously, for extracting features from audio, CNNs adapted for the audio domain, such as AudioVGG [14] or WaveNet [28] pre-trained in dataset AudioSet [12] can be used. In addition, other models are focused on classifying video at the segment level. Deep recurrent models such as LSTMs [15] and GRUs [7] are commonly used for video segment classification as they are well suited to extract temporal features across time.

2.2 Action recognition

Action recognition on trimmed videos has been widely studied. First, methods were based on hand-crafted features such as the work proposed by Wang and Schmid [41], in which they present one of the most used hand-crafted feature descriptors, the *Improved Dense Trajectories* (IDT). However, following the impressive results of deep convolutional neural networks proposed by Krizhevsky *et al.* [16] at the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [31], deep learning methods started to flourish for video classification. For instance, Simonyan *et al.* [36] were the first to propose a two-way stream for action recognition where one stream aims to incorporate spatial information and the other temporal information.

Carreira *et al.* [5] proposed a two-stream architecture, but now with inflated 3D ConvNets. By inflating 2D ConvNets into 3D they were able to extract better Spatio-temporal features. Besides that, they also bootstrapped the parameters of the 3D filters and layers with inflated 2D parameters from pre-trained ImageNet models. Moreover, they also learned that feeding RGB to one stream and optical flow to the other would also greatly improve results. Feichtenhofer *et al.* [11] proposed a two-pathway model, where one pathway is designed to capture semantic and spatial information by operating at low frame rates and a second pathway operating at a high temporal rate designed to capture motion information. By handling input video with different temporal rates, their method with two pathways, each with their own expertise on video modeling, achieved state-of-the-art results.

2.3 Temporal Action Segmentation

Temporal action segmentation has been getting increasing attention in recent years. Fully supervised methods have achieved impressive results but at the cost of widely annotated data that is prohibitive for many real-case scenarios. Therefore, our focus is on weakly supervised and unsupervised setups.

Methods for weakly supervised action segmentation use the actions ordering, termed as *transcripts*, and the video-level activity as weak supervision. One of the first works from Bojanowski *et al.* [4] proposed a method where they use the information on the ordering of actions to train a discriminative clustering approach for action alignment to perform segmentation. Richard *et al.* [30] proposed a learning algorithm with a Viterbi-based loss that directly leverages transcripts allowing online and incremental learning.

A recent proposal by Li *et al.* [22] contributed with a constrained discriminative forward loss (CDFL) to train an HMM and GRU network under weak supervision considering all paths, where each path is a candidate for labeling frames. Kuehne *et al.* [20] contributed with a method where an RNN is used to recognize and classify small temporal clips, this way learning local temporal information. By classifying these small clips, they model complex action classes with subactions. These subactions allow their model to learn fine-grained movements but still capture mid and long-range temporal information frames. Another very recent proposal, by Souri *et al.* [37], utilizes a two-branch network where both try to predict the segmentation and to train it. They propose a novel mutual consistency loss (MuCon) to enforce consistency between the two predictions.

3 METHOD

Our proposal consists of a clustering-based method. We consider a video as a sequence $V = \{f_i\}_{i=1}^{N}$ of N samples. First, we extract spatio-temporal embeddings for each sample $f_i \in V$. This results in a matrix $M_{N \times d_{model}}$ where d_{model} is the embeddings's dimensionality.

Since clustering algorithms do not use temporal frame ordering information, following the approach proposed by Vaswani *et al.* [39], we use the positional encoding approach to inject positional information into the model. Then, we apply the cluster-based heuristic proposed by Mendes *et al.* [25] to find the optimal number of clusters in which each cluster is expected to represent a different action. Figure 1 illustrates the method overview. In the remainder of this section, we detail each step involved in our proposal.

3.1 Video Embeddings Extraction

The first step in our method is the feature-descriptors generation for tiny clips of a video. We use the I3D [5] pre-trained model in the Kinetics400[4] dataset with RGB and optical flow inputs to generate the spatio-temporal embeddings. Each embedding is in the $\mathbb{R}^{2048}$ feature space. The extraction is performed in video snippets of 0.5 s frames, the minimum temporal window possible to input in the I3D architecture. We slide this 10 frames window for every frame, generating a video with N frames, $N - 10$ feature vectors. To fill the matrix with the same number of feature vectors and frames, we simply repeat the last feature vector until the dimensionality of our feature vector matrix M is $N \times d_{model}$ where $d_{model} = 2048$.

3.2 Positional Encoding

In our method, we aim to segment a video by clustering frame embeddings. However, common clustering algorithms make no use of the temporal frame ordering, which is important for temporally segmenting a video. This temporal information is crucial to establish the sub-actions that form more complex actions, which of course are close to each other in the time dimension; the same happens

[4] https://deepmind.com/research/open-source/kinetics

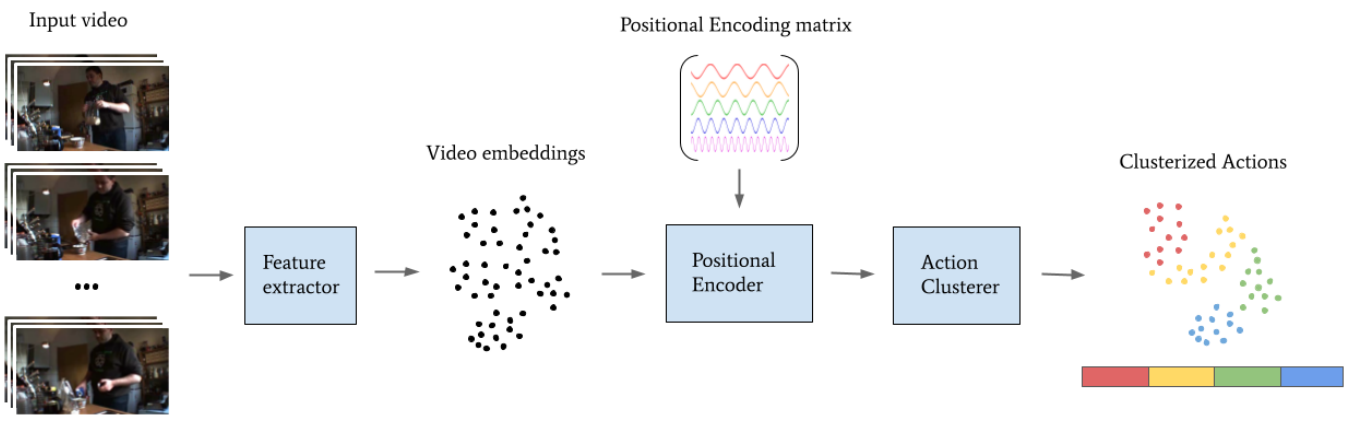

Figure 1: Method Overview.

with frames of the same action. To address this problem, some previous works included frame-wise feature vectors into a temporal embedding. For instance, VidalMata et al.[40] trained a temporal embedding model as a Multilayer Perceptron with the learning goal of predicting the relative timestamp t of a given frame.

Despite of this trend in the literature, we propose a simpler way that requires no training to store temporal information in the frame embeddings. We opted to use the *positional encoding method* proposed by Vaswani et al.[39] in which we produce an encoding matrix, $PE_{N \times d_{model}}$, where N is the number of frames in the input video and d_{model} is the dimensionality of each frame feature descriptor. Likewise, we use the positional encoding with the sine and cosine functions:

$$PE_{(pos,2i)} = sin(\frac{pos}{10000^{2i/d_{model}}})$$

$$PE_{(pos,2i+1)} = cos(\frac{pos}{10000^{2i/d_{model}}})$$

After constructing PE we sum it to the video representation in the feature space resulting in a representation with positional information.

3.3 Action Clustering

For this step, we considered two clustering algorithms: (1) k-Means [24] a centroid-based method and one of the most used unsupervised learning algorithms, and (2) FINCH [34] a state-of-the-art hierarchical agglomerative clustering method, also used in [33].

Combined with k-Means, we apply the method proposed by Mendes *et al.* that uses the *Silhouette Score* to find the optimal number of clusters. The *Silhouette Score* corresponds to the mean of the *Silhouette Coefficient* of all samples, which is calculated by the following equation:

$$S = \frac{b - a}{max(a, b)} \quad (1)$$

where a is the mean distance from a sample to all other samples in the same cluster, and b is the mean distance from a sample to all other samples in the closest cluster to that sample.

That way, the best value is 1, and the worst is -1. Values close to 0 indicate overlapping clusters, whereas negative values usually indicate that a sample has been assigned to the wrong cluster since a different cluster is more similar. In this strategy, we try to increase the number of clusters until the maximum Silhouette Score does not increase for more than t times in a row or until it reaches the maximum number of clusters, which is the number of data points. When it stops, we return the clustering configuration with the highest Silhouette Score. Each cluster corresponds to a different action.

While with KMeans, we used the *Silhouette Score* to propose an automatic way of finding the optimal number of clusters, FINCH does not need that. Most clustering methods are based on the direct distance between samples. However, in high dimensional spaces, distances are less informative. For this reason, Sarfraz et al.[34] proposed FINCH, a clustering method based on the intuition that semantic relations are indirect relations that are not sensitive for high dimensional spaces. They observed that the first neighbor of each data point is sufficient to discover linking chains in the data. So with a recursive approach, they generate a first neighbors adjacency matrix, representing the clusters for the first partition. For each new partition, they use mean vectors of the previous partitions to generate the next one. At the end of this process, they generate a hierarchical structure where each successive partition is a superset of all previous partitions.

4 EXPERIMENTS

This section evaluates the effectiveness of our method by comparing it with state-of-the-art models in two benchmark datasets. Since our proposal consists of an unsupervised method, we restrict the model list in both benchmarks only to models of the unsupervised or weakly supervised type for a fair comparison. It is worth pointing out that our implementations of the networks, training module, saved models, and datasets can be obtained in our public git repository.[5]

[5] blind-review

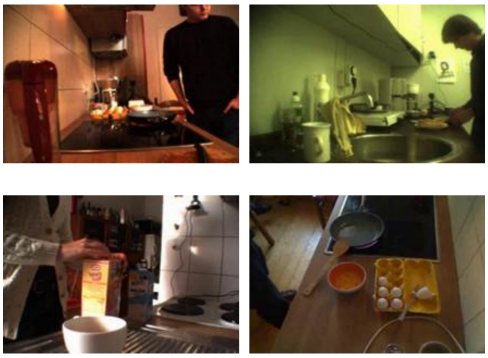

Figure 2: Samples from the "Tea" and "Coffee" activities from the Breakfast Dataset

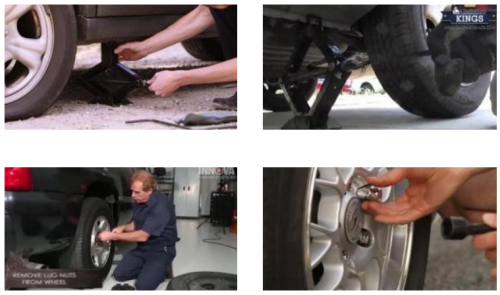

Figure 3: Samples from "Change car tire" action in INRIA Dataset

The remainder of this Section is structured as follows. First, in section 4.1 describes the two datasets used in the experiment, followed by section 4.2, where we describe our experimental setup. The selected metrics used in our evaluation are discussed in section 4.3 and, finally, our empirical findings and results are registered in section 4.4.

4.1 Datasets

We evaluate our method using two benchmark datasets: the Breakfast [18], and Inria Instructional Videos [2]. The Breakfast is a large-scale dataset with 1,712 videos comprising ten different complex cooking activities with, on average, six actions per video. Moreover, video duration can vary significantly, ranging from 30 seconds to a few minutes; The Inria Instructional Videos (INRIA) dataset comprises 150 videos with 2 minutes of duration on average, containing nine actions per video on average, and has a very high ratio of background frames. Figures [**?**] and [**?**] illustrate examples of the Breakfast and INRIA datasets, respectively.

4.2 Setup

Our method was tested with a 6 cores i7 2.60 GHz CPU and a RTX-2070 Max-Q Design GPU. We had to set only 2 hyperparameters: the number of times, t, that the Silhoutte Score does not increase which we used $t = 2$; the maximum number of clusters, C that the Silhoute Score heuristic could reach we set it to $C = 15$.

4.3 Metrics

In order to evaluate the temporal segmentation, we need a one-to-one mapping between the method's output and the ground-truth labels. To generate such a mapping, following [21][35][33], we use the Hungarian method. After generating such a mapping for each video, we report the accuracy as the mean over frames (MoF) metric for both datasets.

4.4 Results

As can be seen from Table 1, our proposed approach outperforms almost all unsupervised and weakly supervised methods in the Breakfast dataset. By comparing our method with the baselines established by Sarfraz et al.[33] we can see that the positional encoding is very helpful to generate better segmentation. However, as mentioned by [33] one of the limitations that greatly affects this method is that the same actions occurring in temporally distant moments are always assigned to different clusters. This is especially true to our method as well, due to the effect of positional encoding. For instance, two frames are injected with positional information that is increasingly different as their temporal distance increases. This effect can cause over-segmentation, which may be especially detrimental for datasets that have actions which appear multiple times in distant temporal instants. This over-segmentation effect can be seen in the results of table 2. The Inria dataset has a very high percentage of background frames, so multiple segments that should be assigned to the same cluster are instead assigned to different clusters.

Despite of achieving competitive results, TW-FINCH is still very much ahead when it comes to accuracy. We believe this is true due to the way they encode temporal information while building the clusters. As briefly explained in Subsection 3.3, FINCH utilizes a one nearest neighbor graph to generate clusters recursively, and TW-FINCH is very similar. The difference is that when TW-FINCH builds its graph, it does by computing the spatio-temporal feature space distance and modulating the frame's features with their respective temporal position.

Another interesting observation we make is that K-Means combined with *Silhouette Score* is performing better than FINCH. We think that this is because FINCH does not use the feature space the same way as k-Means does, as it focus on semantic relations rather than on distances. Therefore, we believe that the positional information injection is altering the semantic structure of the spatio-temporal embeddings, causing this performance deterioration in FINCH.

Figure 4 illustrates examples of predictions made by our best model. The bars represents the entire video timeline, while each color represents the video segments. On the top bar are the predicted actions, and on the bottom bar are the ground truth actions.

5 FINAL REMARKS

In this work, we proposed a novel unsupervised method for action recognition. Our method consists of extracting spatio-temporal features from videos in samples of 0.5 s, using the I3D model. Then the data is transformed using the positional encoder. Finally, we apply the k-Means algorithm combined with the *Silhouette Score* to find the optimal number of clusters in which each cluster corresponds to

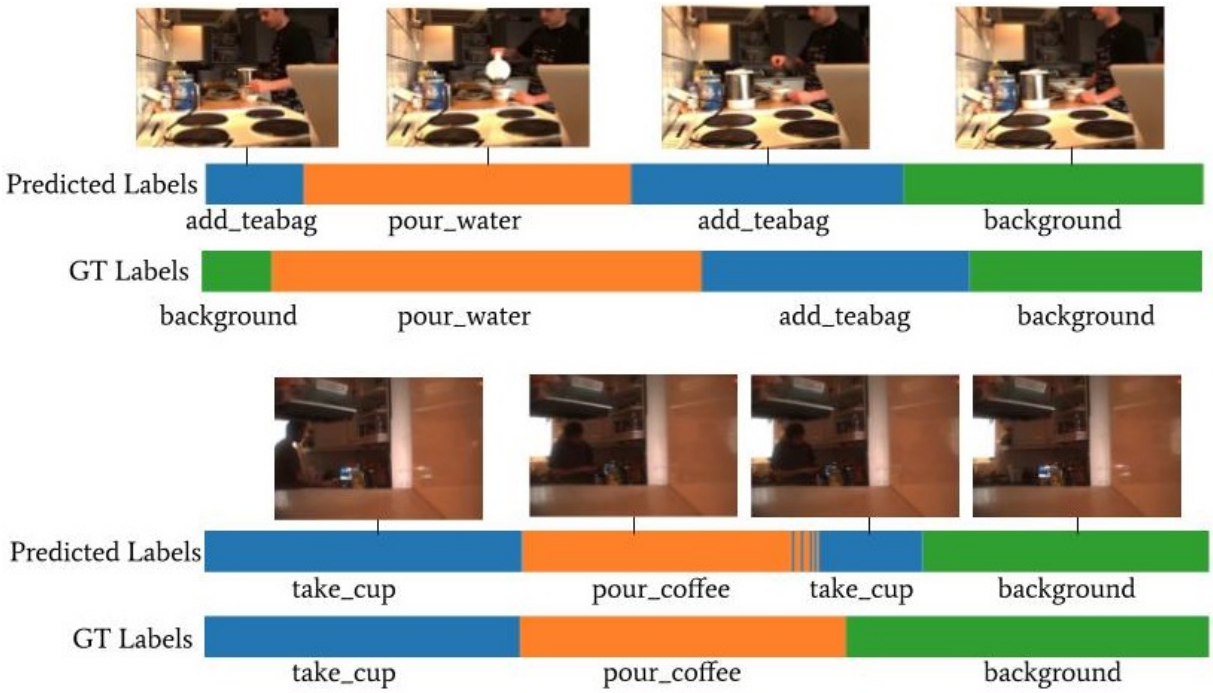

Figure 4: Action predictions produced by the I3D+KMeans+SS+PE model.

Table 1: Comparison on the Breakfast dataset

#	Method	Type	MoF
01	TW-FINCH [33]	Unsupervised	62.7
02	**I3D+KMeans+SS+PE (Ours)**	**Unsupervised**	**58.6**
03	**I3D+FINCH+PE (Ours)**	**Unsupervised**	**55.6**
04	VTE-UNET [40]	Unsupervised	52.2
05	CDFL [22]	Weakly Sup.	50.2
06	MuCon [37]	Weakly Sup.	49.7
07	D3TW [6]	Weakly Sup.	45.7
08	NN-vit [30]	Weakly Sup.	43.0
09	LSTM+AL [1]	Unsupervised	42.9
10	CTE [21]	Unsupervised	41.8
11	TCFPN [9]	Weakly Sup.	38.4
12	RNN+HMM [20]	Weakly Sup.	36.7
13	Mallow [35]	Unsupervised	34.6
14	RNN-FC [29]	Weakly Sup.	33.3
15	SCT [10]	Weakly Sup.	30.4
16	GMM+CNN [19]	Weakly Sup.	28.2

Table 2: Comparison on the Inria Instructional Videos dataset

#	Method	Type	MoF
01	TW-FINCH [33]	Unsupervised	58.6
02	**I3D+KMeans+SS+PE (Ours)**	**Unsupervised**	**52.1**
03	CTE [21]	Unsupervised	39.0
04	**I3D+FINCH+PE (Ours)**	**Unsupervised**	**33.8**
05	Mallow [35]	Unsupervised	27.8

a different action. We evaluated our method using two well-know benchmark datasets and obtained results that were competitive with the state-of-the-art.

We must highlight that, despite our competitive results, TW-FINCH is still ahead when it comes to accuracy. We believe it is due to the way TW-FINCH encode temporal information while building the clusters. To overcome this disadvantage we think that utilizing more advanced models to extract the clips features such as [11] could be helpful. Moreover, we could use other modalities present in videos like audio and use models such as [42]. Another future work we pretend to make is to attack the problem of over-segmentation. We believe by interpreting each segment as a time-series data point and applying a post-processing method to cluster these segments generated with some time-series focused clustering method like Dynamic Time Warping (DTW) proposed by [17] could be helpful.

REFERENCES

[1] Sathyanarayanan N. Aakur and Sudeep Sarkar. 2019. A Perceptual Prediction Framework for Self Supervised Event Segmentation. *arXiv:1811.04869 [cs]* (April 2019). http://arxiv.org/abs/1811.04869 arXiv: 1811.04869.

[2] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. 2016. Unsupervised Learning from Narrated Instruction Videos. In *CVPR2016 - 29th IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas, United States. https://hal.inria.fr/hal-01171193

[3] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5297–5307.

[4] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. 2014. Weakly Supervised Action Labeling in Videos Under Ordering Constraints. *arXiv:1407.1208 [cs]* (July 2014). http://arxiv.org/abs/1407.1208 arXiv: 1407.1208.

[5] Joao Carreira and Andrew Zisserman. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. 6299–6308. https://openaccess.thecvf.com/

content_cvpr_2017/html/Carreira_Quo_Vadis_Action_CVPR_2017_paper.html

[6] Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. 2019. D3TW: Discriminative Differentiable Dynamic Time Warping for Weakly Supervised Action Alignment and Segmentation. *arXiv:1901.02598 [cs]* (April 2019). http://arxiv.org/abs/1901.02598 arXiv: 1901.02598.

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. (2009).

[9] Li Ding and Chenliang Xu. 2018. Weakly-Supervised Action Segmentation with Iterative Soft Boundary Assignment. *arXiv:1803.10699 [cs]* (March 2018). http://arxiv.org/abs/1803.10699 arXiv: 1803.10699.

[10] Mohsen Fayyaz and Juergen Gall. 2020. SCT: Set Constrained Temporal Transformer for Set Supervised Action Segmentation. *arXiv:2003.14266 [cs]* (March 2020). http://arxiv.org/abs/2003.14266 arXiv: 2003.14266.

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. SlowFast Networks for Video Recognition. 6202–6211. https://openaccess.thecvf.com/content_ICCV_2019/html/Feichtenhofer_SlowFast_Networks_for_Video_Recognition_ICCV_2019_paper.html

[12] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 776–780.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[14] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. 2017. CNN Architectures for Large-Scale Audio Classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://arxiv.org/abs/1609.09430

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997).

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (May 2017), 84–90. https://doi.org/10.1145/3065386

[17] JB Kruskal and Mark Liberman. 1983. The symmetric time-warping problem: From continuous to discrete. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison* (Jan. 1983).

[18] Hilde Kuehne, Ali Arslan, and Thomas Serre. 2014. The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 780–787. https://doi.org/10.1109/CVPR.2014.105 ISSN: 1063-6919.

[19] Hilde Kuehne, Alexander Richard, and Juergen Gall. 2017. Weakly supervised learning of actions from transcripts. *arXiv:1610.02237 [cs]* (June 2017). http://arxiv.org/abs/1610.02237 arXiv: 1610.02237.

[20] Hilde Kuehne, Alexander Richard, and Juergen Gall. 2020. A Hybrid RNN-HMM Approach for Weakly Supervised Temporal Action Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 4 (April 2020), 765–779. https://doi.org/10.1109/TPAMI.2018.2884469 arXiv: 1906.01028.

[21] Anna Kukleva, Hilde Kuehne, Fadime Sener, and Jurgen Gall. 2019. Unsupervised Learning of Action Classes With Continuous Temporal Embedding. 12066–12074. https://openaccess.thecvf.com/content_CVPR_2019/html/Kukleva_Unsupervised_Learning_of_Action_Classes_With_Continuous_Temporal_Embedding_CVPR_2019_paper.html

[22] Jun Li, Peng Lei, and Sinisa Todorovic. 2019. Weakly Supervised Energy-Based Learning for Action Segmentation. *arXiv:1909.13155 [cs]* (Sept. 2019). http://arxiv.org/abs/1909.13155 arXiv: 1909.13155.

[23] Jun Li and Sinisa Todorovic. 2021. Action Shuffle Alternating Learning for Unsupervised Action Segmentation. *arXiv:2104.02116 [cs]* (April 2021). http://arxiv.org/abs/2104.02116 arXiv: 2104.02116.

[24] J. Macqueen. 1967. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*. 281–297.

[25] Paulo Renato C Mendes, Antonio José G Busson, Sérgio Colcher, Daniel Schwabe, Álan Lívio V Guedes, and Carlos Laufer. 2020. A Cluster-Matching-Based Method for Video Face Recognition. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*. 97–104.

[26] Paulo Renato C Mendes, Eduardo S Vieira, Pedro Vinicius A de Freitas, Antonio José G Busson, Álan Lívio V Guedes, Carlos de Salles Soares Neto, and Sérgio Colcher. 2020. Shaping the Video Conferences of Tomorrow With AI. In *Anais Estendidos do XXVI Simpósio Brasileiro de Sistemas Multimídia e Web*. SBC, 165–168.

[27] Antoine Miech, Ivan Laptev, and Josef Sivic. 2017. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905* (2017).

[28] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[29] Alexander Richard, Hilde Kuehne, and Juergen Gall. 2017. Weakly Supervised Action Learning with RNN based Fine-to-coarse Modeling. *arXiv:1703.08132 [cs]* (Oct. 2017). http://arxiv.org/abs/1703.08132 arXiv: 1703.08132.

[30] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. 2018. NeuralNetwork-Viterbi: A Framework for Weakly Supervised Video Learning. *arXiv:1805.06875 [cs]* (May 2018). http://arxiv.org/abs/1805.06875 arXiv: 1805.06875.

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575 [cs]* (Jan. 2015). http://arxiv.org/abs/1409.0575 arXiv: 1409.0575.

[32] Gabriel NP dos Santos, Pedro VA de Freitas, Antonio José G Busson, Álan LV Guedes, Ruy Milidiú, and Sérgio Colcher. 2019. Deep learning methods for video understanding. In *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web*. 21–23.

[33] M. Saquib Sarfraz, Naila Murray, Vivek Sharma, Ali Diba, Luc Van Gool, and Rainer Stiefelhagen. 2021. Temporally-Weighted Hierarchical Clustering for Unsupervised Action Segmentation. *arXiv:2103.11264 [cs]* (March 2021). http://arxiv.org/abs/2103.11264 arXiv: 2103.11264.

[34] M. Saquib Sarfraz, Vivek Sharma, and Rainer Stiefelhagen. 2019. Efficient Parameter-free Clustering Using First Neighbor Relations. (Feb. 2019). https://arxiv.org/abs/1902.11266v1

[35] Fadime Sener and Angela Yao. 2018. Unsupervised Learning and Segmentation of Complex Activities from Video. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, UT, 8368–8376. https://doi.org/10.1109/CVPR.2018.00873

[36] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 568–576. http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf

[37] Yaser Souri, Mohsen Fayyaz, Luca Minciullo, Gianpiero Francesca, and Juergen Gall. 2020. Fast Weakly Supervised Action Segmentation Using Mutual Consistency. *arXiv:1904.03116 [cs]* (March 2020). http://arxiv.org/abs/1904.03116 arXiv: 1904.03116.

[38] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv:1706.03762 [cs]* (Dec. 2017). http://arxiv.org/abs/1706.03762 arXiv: 1706.03762.

[40] Rosaura G. VidalMata, Walter J. Scheirer, Anna Kukleva, David Cox, and Hilde Kuehne. 2020. Joint Visual-Temporal Embedding for Unsupervised Learning of Actions in Untrimmed Sequences. *arXiv:2001.11122 [cs]* (Sept. 2020). http://arxiv.org/abs/2001.11122 arXiv: 2001.11122.

[41] Heng Wang and Cordelia Schmid. 2013. Action Recognition with Improved Trajectories. 3551–3558. https://openaccess.thecvf.com/content_iccv_2013/html/Wang_Action_Recognition_with_2013_ICCV_paper.html

[42] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. 2020. Audiovisual SlowFast Networks for Video Recognition. (Jan. 2020). https://arxiv.org/abs/2001.08740v2