7

Unsupervised Anomaly Detection and Temporal Segmentation of Long-Term Action Sequences

In Chapter 5, we saw that performances from multiple experts can be equally good candidates for template matching, but can differ in terms of feedback for the same test sequence. This is due to differences in the templates that come because of speed or little variations in postures. In the last chapter we provided a more generalized solution compared to template matching. To compensate for minute variations in poses we used a K-means encoding technique over pose features and to handle variations in speed we proposed an autoencoder based model that learned to construct all expert sequences. The reconstructed sequences from this model was an expert rendering adapted to the speed of the performance given as input. This approach outperformed template matching approach in skill assessment. However, a challenge still remains: the encoding approach used in the previous chapter forces the anomalous poses to be a part of one of the key clusters and hence the anomalous pose also gets identified as a known pose. This leads to an under-performance in skill determination. We need a segmentation algorithm that can divide a long sequence into sub-action segments such that the counts of these sub-segments is not pre-specified. Such a technique would naturally cluster frames and can identify anomalous sub-segments as separate clusters. To this end, we present a novel community detection-based human action segmentation algorithm. Our work marks the existence of community structures in human action videos where the consecutive frames around the key poses group together to form communities, similar to social networks.

7.1 INTRODUCTION

Temporal segmentation of complex human action videos into action primitives plays a pivotal role in building models for human action understanding. Studies in the past have introduced unsupervised frameworks for deriving a known number of motion primitives from action videos.

Previous works on human motion segmentation can be divided into three categories - a) Supervised techniques, where the ground truth labels are available and models are built over these labeled video datasets, e.g. Bag-of-Features approach [Duchenne *et al.* [2009a]], Template matching based segmentation approach [Jain and Harit [2018]] and Hidden Markov Model (HMM) [Borzeshi *et al.* [2013a]; Jain and Harit [2016]]; b) Unsupervised approaches that model the video sequences without their ground-truth labels and have an explicit training phase, e.g. probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA) [Niebles *et al.* [2008a]]; c) Unsupervised segmentation techniques which identify distinct actions in human activity videos without the need of training. Techniques in the first set cluster individual poses from all frames to extract meaningful action primitives [Turaga *et al.* [2009]De la Torre *et al.* [2007]] e.g. Spectral clustering [Ng *et al.* [2002]]. Techniques in the second set like ACA [Zhou *et al.* [2008]] and HACA [Zhou *et al.* [2013]] solve for variability in the temporal scales of human actions. These techniques partition the video data into m segments of variable lengths and cluster these segments to one of the k action clusters. Common to all these works is the prior knowledge about number of action types. But its not always necessary that we have this count such as in anomalous action detection, where we may not have the count of anomalies.

Our work focuses towards answering a question : Given a set of videos with humans performing

Technique	Supervised	Training Required	No of actions required
Bag of words HMM	YES	YES	YES
PLSA / LDA	NO	YES	YES
ACA / HACA / SC	NO	NO	YES
Community Detection	NO	NO	NO

Figure 7.1 : Our community detection based segmentation technique is unsupervised and requires no training and can segment without knowledge of count of action categories

an activity, can the action primitives be derived from them without specifying any prior knowledge about the count for the constituting sub-actions categories?

We develop an unsupervised human action segmentation approach that requires no training phase and which can segment human actions without explicitly specifying the count of the sub-actions in the videos. For example, in a sequence of a person who first runs for some time, then stops and then bends to pick up something, we aim to segment the entire performance into running segment and bending segment, without providing the count of actions, $k = 2$. Figure 7.1 gives an overview of our work.

To address our problem of unsupervised segmentation, we adopt ideas from social networking theory that concern with the properties of community structures, i.e., they naturally divide into groups of nodes with denser connections inside each group and fewer connections across groups. Human action videos display similar nature. Each action has a representative pose (key pose). Poses attained for a given action exhibit similarity to the key pose and hence constitute a single community, similar to friend groups in Facebook or professional connections in LinkedIn.

Community detection algorithms Lancichinetti and Fortunato [2009] naturally divide such social networks into communities, with denser connections in the community and sparser connections outside the community. Network-based concepts towards activity understanding were recently used in Fallahzadeh and Ghasemzadeh [2017], where activities in the test video were labeled using community detection based algorithms and the network parameters, like optimal similarity threshold were learned from the training data. In contrast, the network in case of our work is a self learned network and directly applies over the test data, and requires no parameter learning from training videos.

Community detection and clustering are closely related to each other. They both share the same objective of partitioning nodes into groups. Clustering algorithms take as input, the number of clusters that need to be identified whereas the community detection algorithms do not require the number of communities to be pre-specified. Our proposed methodology of human action segmentation using community detection presents three advantages :

- It is an unsupervised technique with no training phase.
- No prior knowledge about the count of sub-action types 'k' required.
- It is computationally efficient

We discuss our technique of unsupervised human action segmentation followed by the task of anomlaous action discovery using this technique.

7.2 UNSUPERVISED TEMPORAL ACTION SEGMENTATION AND ANOMALY DETECTION: PROPOSED METHODOLOGY

Our approach consists of four components: (a) feature extraction (b) frame-wise distance matrix computation (c) network construction from video frames d) key pose detection using community detection e) Anomlaous pose detection

7.2.1 Feature extraction

The framework treats human action videos as multivariate time-series. Each frame of the sequence is encoded as a feature vector $f \in \Re^d$. Vector f can stand for skeleton features that represent human joint locations, or a shape context feature for silhouette, or a Bag of feature representation. We deploy the method on two datasets, using two different feature formulations.

7.2.2 Frame wise distance matrix computation

Given an action sequence S of length l, we calculate the distance matrix $\mathbf{W}$ of the pair-wise distances of all frames of the sequence. Depending on the type of feature used, different distance functions can be used (e.g. Euclidean distance for motion data, χ^2 distance for histogram features, shape context cost, etc.)

7.2.3 Network Construction from video frames

The exact number of action segments in each sequence is unknown. We borrow the network based concepts for action segmentation which do not involve any additional guidance in terms of the number of clusters. Community detection algorithms help us for the same, where frames around the key poses are coherent and form communities that signify action categories.

To construct the network for an action sequence S, all frames $f_i, i = 1, 2, ...l$ of sequence S form nodes of the network. Next, an optimal distance threshold α is learned such that all nodes with distance less than α are connected. An optimal threshold is the one for which the resulting network is connected.

Spectral graph theory gives us the condition for connectedness of the network by checking the eigenvalues of the graph Laplacian. The graph Laplacian is the matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the diagonal matrix whose entries are the degree of each node and $\mathbf{A}$ is the adjacency matrix. The smallest eigenvalue of $\mathbf{L}$, that is λ_1, is always 0. The second smallest eigenvalue λ_2 gives us a measure of how well connected a graph is. A positive value of λ_2 signifies that the network is connected. Thus, we set α to a small value and check for connectivity of the network. The value of α is incremented in small steps the value for which the resulting network is connected is the final value. Algorithm 2 formally states network construction procedure.

7.2.4 Key Pose detection and Anomlay Detection using Commmunity Detection

Community detection [Fortunato [2010]] aims at grouping the nodes of the graph on the basis of the relationships that exists between them to generate strongly connected sub-graphs for a given graph.

Algorithm 2 Network Construction

```
procedure CONSTRUCT SIMILARITY NETWORK(W, α)
    A ← zeros(l,l);                                   ▷ Initialize Adjacency Matrix A
    connected ← false;
    do
        Construct a node for each frame f ∈ S;
        for each frame pair f_i, f_j ∈ S do
            if W(i,j) ≤ α then
                A(i,j) ← 1;
            end if
        end for
        if CHECK CONNECTIVITY(A) == false then
            α ← α + 0.005;
        else
            connected ← true;
        end if
    while connected ≠ true
    return A;                                         ▷ Adjacency matrix of Similarity Graph
end procedure
procedure CHECK CONNECTIVITY(A)
    Construct Degree Matrix D;
    L ← D − A;                                        ▷ Laplacian Matrix
    Z ←Sort ( Eigenvalues(L));                        ▷ Ascending order
    if Z(2) > 0 then
        return true;
    else
        return false;
    end if
end procedure
```

Communities constitute a non-overlapping group of clusters with dense intra-cluster links and sparser links between the groups. Mathematically, it is a list of non-empty node subsets $C = \{c_1, c_2, c_3 ..... c_n\}$, where $\bigcup_{i=1}^{n} c_i \subseteq V$, and n is the count of communities. Further, $c_i \cap c_j = \emptyset$.

Outliers - Community detection does not force every node to be a part of some group. These independent nodes that are not part of any community are referred to as outliers. Mathematically, $O = \{v | v \in V, \nexists c_i \in C \wedge v \in c_i\} = V - \bigcup_{i=1}^{n} c_i$. Outliers are the tiny groups whose size is less than a threshold, the valid size of a community.

Community detection algorithms partition the graph $G(V,E)$ into a set of communities C and outliers O such that the communities are closely connected group of nodes, and outliers are poles apart from these communities.

Community detection is an extensively researched topic with many algorithms designed for non-overlapping community division. These algorithms optimize a quality function called *modularity*. Modularity is the strength of division of a network into modules. Mathematically it is given as :

$$Q = \sum_{i=1}^{k} (e_{ii} - a_i^2)$$

where e_{ii} is probability that the edge is in module i and a_i is the probability that a random edge would fall in module i. A high value of modularity signifies more edges within the module that you expect by chance.

We use the Fast Greedy Algorithm [Newman [2004]] for community detection for its computational efficiency. It is a bottom-up hierarchical approach to form communities and optimizes modularity in a greedy manner.

Initially, every vertex belongs to a separate community, and then iteratively, the communities

are merged such that there is an increase in the value of modularity after every iteration. The algorithm stops when it is not possible to increase the modularity any more, and outputs the final set of communities. Outcome of community detection is a set of communities. The communities can be large or small in size.

Larger communities - Frames with similar pose from multiple executions by different people get grouped together to form larger communities. These reflect the key poses of a complex action.

Smaller communities - These result when a person, for a small duration takes a strange pose very different from other poses seen in the execution instances. Errors in pose estimation also results in small communities getting formed. Further, while the person transits from one key pose to the other, the intermediate poses change throughout the transition period and form smaller communities.

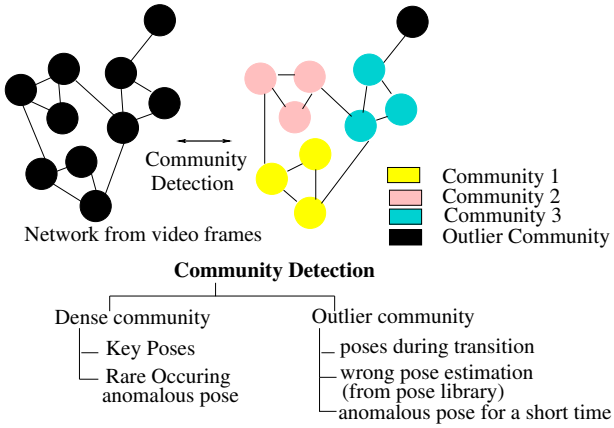


Figure 7.2 : Analysis of community detection results for video dataset

7.2.5 Filtering out anomalous poses from imperfect videos

We now analyze Community Detection applied to a set of complex action video dataset that comprises a mix of incorrect executions of the same complex action. A performer can perform a part of the activity correctly, but misses some poses or executes the poses out of order or can perform some anomalous poses. However, we assume that a particular pose will be performed correctly and in the right context by a majority of performers, i.e. the order of key poses remains partially correct in almost all video sequences.

Since the dataset contains incorrect executions, a few communities that are larger in size may get formed when there is a strange action taken by the performer for a longer period of time. However, it is easy to find such anomalous communities. Such communities contain frames from only few execution instances and therefore can be easily identified. These communities are considered anomalous and assigned a label *anomalous*. These are rejected for further consideration during sequence learning task.

It is important to make a contrast with the cited techniques of action segmentation here. The cited techniques whether they are unsupervisedNiebles *et al.* [2008a]Zhou *et al.* [2008]Zhou *et al.* [2013] or supervisedDuchenne *et al.* [2009a] require the prior knowledge about the number of key poses. They divide all frames into these set number of poses and thus cannot identify incorrect poses. Hence, imperfect videos are not used for such techniques.

In contrast, the additional capability of community detection algorithm to identify anomalies makes these imperfect videos an additional resource to gain more confidence in key pose detection. This is illustrated in the Experiments section where we see that as the number of videos (perfect

Algorithm 3 Key Pose Detection

procedure FIND KEY POSES(S) ▷ S : dataset of action sequences
  $\Delta \leftarrow 0.01$
  for each frame pair $f_i, f_j \; \forall \; V \in S$ **do**
    $\mathbf{W}(\mathbf{i},\mathbf{j}) \leftarrow \sum_{\forall \; joints}$ (distance between joint positions)
  end for
  $\mathbf{A} \leftarrow$ CONSTRUCT SIMILARITY NETWORK($\mathbf{W}, \Delta$)
  $C \leftarrow$ Community Detection Algorithm over $\mathbf{A}$
  for $c_i \in C$ **do** ▷ C is the set of communities
    if $|c_i| > \delta$ **then** ▷ $|c_i|$ is the size of community c_i
      ▷ larger communities
      ▷ wrong poses taken for a long duration
      if $P(c_i|videos) < 0.7$ **then** $Anomalies \leftarrow$ append c_i
      else $Key\_Poses \leftarrow$ append c_i
      end if
    else ▷ smaller communities
      ▷ wrong pose for short time or transitions
      if $|c_i| < \delta$ **then** $Anomalies \leftarrow$ append c_i
      end if
    end if
  end for
end procedure

or imperfect) increases the segmentation accuracy of the video increases. Figure 7.2 illustrates the community detection results : outliers, anomalous communities and key poses. Algorithm 3 formalizes the steps to find key poses and anomalies in action videos. Such a sequence analysis technique now results into different clusters of anomalies and key poses and thus a better performance of our autoencoder model.

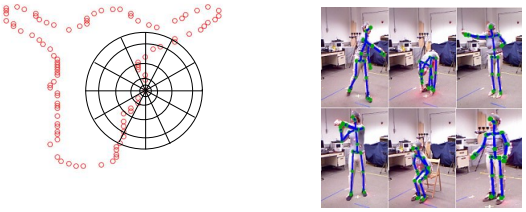
7.3 TEMPORAL CLUSTERING TECHNIQUES

For comparison of our approach we have selected two recently proposed unsupervised approaches :

Aligned Clustering Analysis(ACA) Zhou et al. *[2008]:* - Given an action sequence $S = [f_1, f_2, ...f_n]$ with n frames, ACA decomposes S into m segments, each of which corresponds to one of k classes. The length of segments is constrained to be $\leq n_{max}$, where n_{max} is the maximum length of the segment and controls the temporal granularity of the factorization. A DP-based approach is used to solve for action boundaries of each action segment with time complexity of each iteration being $\mathcal{O}(n^2 n_{max})$.

Hierarchical Aligned Cluster Analysis(HACA) Zhou et al. *[2013] :* This approach reduces the computational complexity of ACA, and provides a hierarchical decomposition at different scales. HACA, starts the search with smaller temporal scales and propagates the result to larger temporal scale. Similar to ACA, the length constraints are now defined for multiple levels. For e.g., for a two-level HACA, two constraints n^1_{max} and n^2_{max} are defined.

In the next section we illustrate performance comparison of our algorithm with these algorithms where we first discuss the performance of unsupervised sequence segmentation on stitched MHAD and KTH dataset, followed by anomaly detection performance. Finally we illustrate the improvement of our autoencoder technique towards action assessment.



(b)

Figure 7.3 : a) Shape context feature for Weizmann dataset human silhouette b) MHAD101-s human motion skeletal data

7.4 EXPERIMENTS

This section evaluates the performance of Community based action segmentation on stitched segments from human action videos from Weizmann dataset [Gorelick *et al.* [2007]] and motion capture MHAD101-s dataset [Ofli *et al.* [2013]]. We then discuss the performance of the algorithm on anomlaous sub-segment discovery. Finally we illustrate the performance of this preprocessing step in Autoencoder-based skill assessment.

7.4.1 Human Action Segmentation

Evaluation Metrics The performance of community detection(CD) algorithm is compared with the state-of-the-art ACA and HACA algorithms towards unsupervised human motion segmentation algorithms. Additionally, performance is also compared with Spectral Clustering(SC) technique that is one of the most popular clustering techniques.

To evaluate the clustering accuracy, a confusion matrix between the segmentation output and the ground truth segmentation output is calculated. Entries of the confusion matrix, $c(i,j)$ represent the total number of frames which the algorithm labels as segment i and the ground truth carries label j. ACA, HACA and SC algorithms are evaluated by providing the correct number of action segments in the stitched sequence. On the other hand the community detection algorithm is run without specifying the number of action segments in the videos. Communities extracted may be more than the types of action segments present in the stitched video. However, this does not affect the segmentation accuracy as long as the frames in the resulting communities represent the correct semantic action label.

Video Data - Stitched Weizmann dataset [Gorelick et al. [2007]]: In the first experiment we evaluate the performance of community based segmentation algorithm for human action video sequences available in the standard Weizmann dataset.

The dataset contains 9 different actions performed by 10 individuals. The silhouettes of humans are extracted using background subtraction and are aligned to be frame centered and scale normalized. Shape context features [Belongie *et al.* [2001]] with 100 boundary samples, 12 angular bins and 6 radial bins are computed for silhouette masks as shown in Figure 7.3a. Distance between shape context is used for network construction.

Motion Capture Data - MHAD101-s dataset [Ofli et al. [2013]]: In the second experiment we evaluate the performance of community based segmentation algorithm for human motion capture sequences available in the MHAD-101s dataset. The Berkeley Multimodal Human Action Database (MHAD) contains human motion capture data from optical motion capture system and conventional RGB video with depth data acquired from multiple views and depth sensors (Figure 7.3b). The dataset contains 11 actions performed by 12 subjects. The human skeletal data are represented

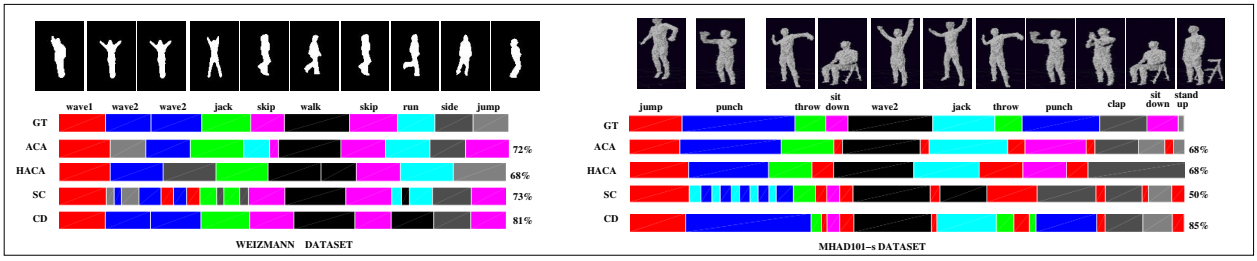


Figure 7.4 : Qualitative segmentation results where *GT* : Ground Truth; Different colors correspond to distinct actions

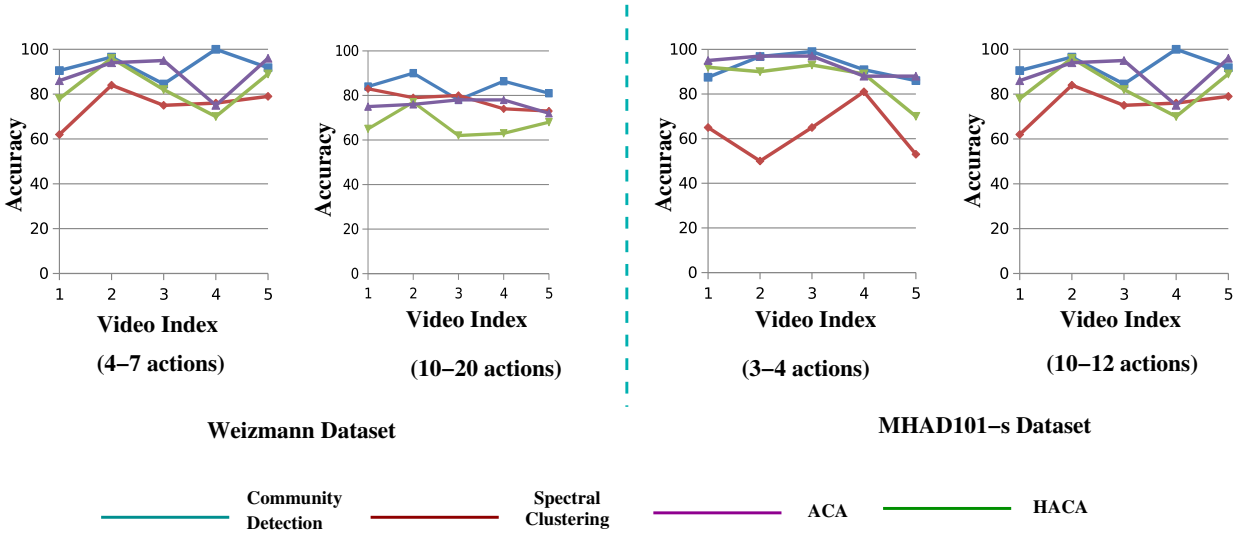


Figure 7.5 : Comparative results for all videos

as a 30-dimensional vector that encodes angles of the selected body parts with respect to the body-centered coordinate system Papoutsakis *et al.* [2017]. Euclidean distance between angular features are used as entries of distance matrix.

We synthesize 10 testing videos for both the datasets. For Weizmann dataset, 5 videos are made by concatenating $4-7$ random clips while the other 5 are made from $10-20$ clips. Similarly for MHAD101-s dataset, 5 videos are made by concatenating $3-4$ clips and the other 5 are made from $10-12$ clips. Louvain-based community detection algorithm is run over the network generated from each sequence. The extracted communities are used to assign action labels to the frames. The ACA, HACA and SC are ran for 10 iterations for each concatenated sequence with a length constraint $n_{max} = 16$ for ACA, and $n^1_{max} = 10$, $n^2_{max} = 8$ for the two levels of HACA in case of Weizmann dataset. For MHAD101-s dataset, length constraints are set to $n_{max} = 24$ for ACA, and $n^1_{max} = 10$, $n^2_{max} = 6$.

Figure 7.4 shows qualitative segmentation results for a sample video sequence of each dataset. The colored segments are derived from the segmentation achieved using the specified algorithm. Table 7.1 shows the average accuracy for each method over 2 sets for each dataset and Figure 7.5 presents the individual accuracy of all videos. This exhibits the improved performance of the proposed

Table 7.1 : Segmentation Accuracy : Community based approach outperforms even without knowledge of no. of actions

	Weizmann (4-7 actions)	**Weizmann** (10-20 actions)	**MHAD101-s** (3-4 actions)	**MHAD101-s** (10-12 actions)
ACA [Zhou *et al.* [2008]]	89.2	75.8	**93**	67.4
HACA [Zhou *et al.* [2013]]	83	67	86.8	78.4
SC [Ng *et al.* [2002]]	75.2	77.8	62.8	52
CD	**92.7**	**83.8**	92	**80.2**

method and its generality over different datasets, variable no. of sub-actions and for different feature representations.

7.4.2 Computational Complexity

The space complexity of both ACA/HACA and Community detection algorithm is $\mathcal{O}(n^2)$ due to computation of distance matrix while the computational complexity of ACA and HACA algorithms is $\mathcal{O}(n^2 n_{max})$ and this limits their applicability over long sequences. In contrast, the Louvain-based community detection algorithm speeds up the segmentation, such that the complexity is $\mathcal{O}(n \log n)$. Thus we see an increased segmentation performance at reduced time complexity, with an advantage of not specifying the number of action labels in the sequence.

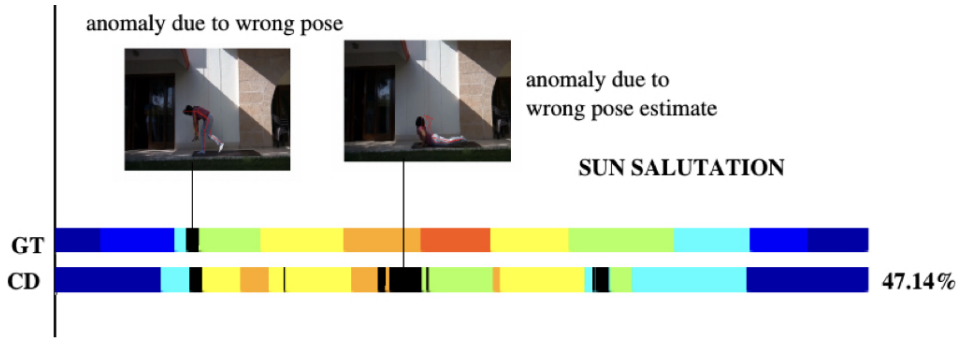


Figure 7.6 : Anomalous pose detection result

7.4.3 Anomaly detection

The algorithm 3 holds importance in recognizing an anomalous pose and a wrongly estimated pose too. Figure 7.6 shows the performance of our algorithm towards recognizing wrong poses. It can be seen that the algorithm correctly identifies the wrong poses and also gives an anomalous label when many consecutive frames are labeled with incorrect poses(due to wrong esimation by a pose estimation library). All anomalies are marked as black in Figure 7.6. Community detection based clustering of poses during preprocessing step leads to an improved Sun Salutation scoring accuracy using the autoencoder-based approach in the last Chapter. This can be seen in Table 7.2 where our autoencoder model with new clustering technique leads to better rank correlation and reduced mean square error.

7.4.4 Leveraging community detection based approach towards improved segmentation

All the existing training datasets [Soran *et al.* [2015]; Wu *et al.* [2015a]] towards action sequence learning contain complex actions that are available in videos with perfectly correct ordered sequences. This requires carefully monitoring all the videos while dataset building and rejecting the incorrect ones

Table 7.2 : Comparison of Rank Correlation and Mean Square Error of various techniques for Sun Salutation action scoring with Community Detection based encoding

Model	MSE	Rank Correlation
SVR-DCT [Pirsiavash et al. [2014]]	0.35	-0.46
SVR-DFT [Pirsiavash et al. [2014]]	0.33	-0.39
Pose Words + LSTM	0.18	0.19
Pose Words +LSTM+SVR	0.22	0.23
C3D + SVR [Parmar and Tran Morris [2017]]	0.23	-0.026
C3D + LSTM + SVR [Parmar and Tran Morris [2017]]	0.17	0.37
Template Matching	0.33 ± 0.026	0.13
Ours	0.12	0.48
Ours + CD	**0.103**	**0.57**

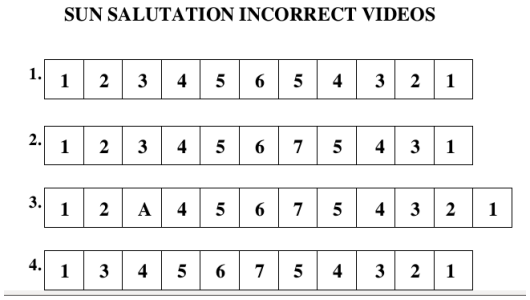


Figure 7.7 : Incorrect sequences of the dataset; A denotes the anomalous actions

during training, which is a labor intensive task. In addition to anomaly detection, we work towards developing a technique that has no constraints over the type of videos whether they are correct or incorrect and still help in good segmentation performance. To validate the same with Sun Salutation dataset that we proposed earlier

Yoga experts are rare to find, thus leading to fewer perfect videos. From the Sun Salutation dataset proposed in the previous chapter, we chose three perfect performers and 4 imperfect performers who missed out on some steps and took wrong poses. Figure 7.7 shows the incorrect action sequences for our dataset. The frames during the transition period between key poses are no more considered as belonging to one of the action classes but are treated as an unknown class.

We evaluate the segmentation performance for the three correct sequences of both the datasets and compare the results with the closest state-of-the-art approaches ACA [Zhou *et al.* [2008]], HACA [Zhou *et al.* [2013]] and Spectral Clustering. The following experiments were performed :

1. Segmentation of the three correct videos using HACA, ACA and SC.
2. The three correct videos were combined to form a single sequence. HACA, ACA, SC and community detection algorithms were performed on the combined sequence and the segmentation accuracy of individual videos was evaluated.
3. Finally we include four random incorrect executions to the second experiment and the

Table 7.3 : Segmentation results : Sun Salutation video segmentation in the presence of incorrect performances

		Video 1	Video 2	Video 3
Single Video Segmentation	ACA [Zhou *et al.* [2008]]	47.02	44.24	37.91
	HACA [Zhou *et al.* [2013]]	55.34	41.96	33.12
	SC [Zhou *et al.* [2008]]	65.67	44.55	39.79
Segmentation in presence of correct action instances	ACA [Zhou *et al.* [2008]]	51.13	50.12	39.63
	HACA [Zhou *et al.* [2013]]	58.09	48.54	35.16
	SC [Zhou *et al.* [2008]]	57.33	42.82	34.26
	CD	69.62	76.23	56.87
Segmentation in presence of perfect and imperfect videos	CD	**72.04**	**76.56**	**62.12**

segmentation performance over individual correct videos was evaluated.

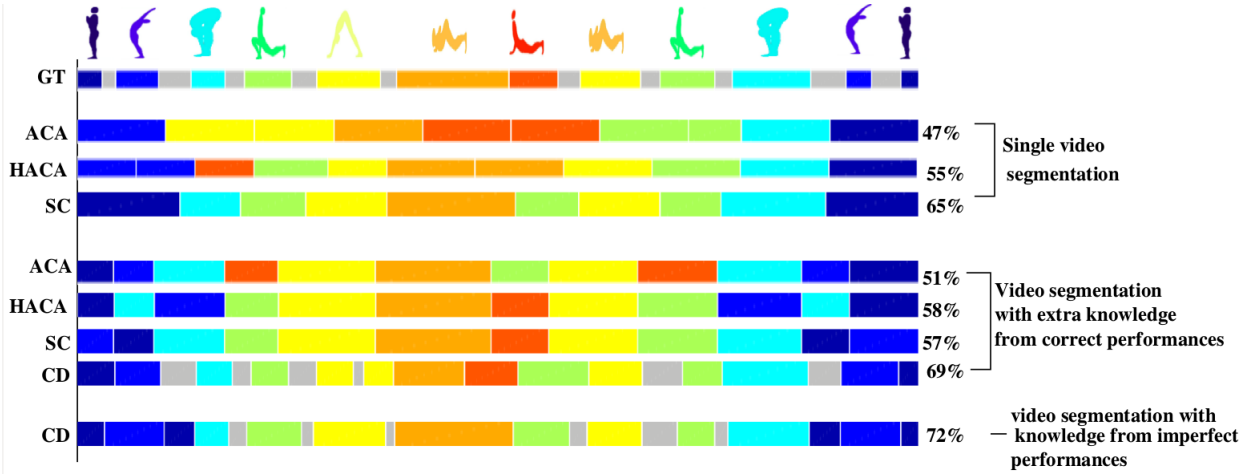


Figure 7.8 : Segmentation results : GT refers to the ground truth; different colors represent different poses; Gray color indicates transition frames between the key poses

Figure 7.8 shows the qualitative segmentation result for one video from each dataset. It is seen that using our algorithm and some imperfect videos as an extra instance for learning, the segmentation performance of the algorithms has considerably improved.

The increased performance is achieved due to two reasons : 1) Since there is no constraint on the number of poses to be given to the community detection algorithm, the transitions between the key poses are not forced to be classified as a key pose but instead form smaller communities different from the key pose communities, and thus results in a better matching with the ground truth. 2) The imperfect videos yield communities that are more coherent and thus leads to increased segmentation accuracy.

Table 7.3 lists the segmentation accuracy of our framework and compared it with state-of-the-art approaches. It is seen that the community detection algorithm has a reasonably better segmentation accuracy compared to the state-of-the-art techniques. This is significant because our method did not have any prior knowledge about the number of poses and it took advantage including

knowledge from imperfect videos. It is important here to note that the state-of-the-art technique were run in best settings i.e. the number of key poses are set for these techniques.

7.5 CONCLUSION

In this chapter, we propose a novel community detection-based unsupervised human action segmentation technique that do not require the number of action labels present in the video to be specified unlike other clustering techniques like ACA, HACA and Spectral clustering. The method is evaluated for Weizmann video dataset and MHAD101-s motion dataset. The technique outperforms previous unsupervised techniques - ACA and HACA at reduced complexity of $\mathcal{O}(n \log n)$. This framework helps in identifying anomalous and correct poses as separate communities. This results in a better representation of the test videos eventually leading to improved capability of our autoencoder-based assessment framework developed in last chapter to provide feedback for test videos.

Unlike Sun Salutation, the expert performances (which carry a high rating) are too few in case of Olympics events like diving and vaults. Thus the previously proposed autoencoder-based assessment technique is unsuitable for such a usecase. To overcome this limitation, we adopted a deep metric learning method that learns to score the similarity in performances of two input videos where the performance in the pair is not constrained to contain an expert performance. The learned metric is utilized by the score prediction model that compares a video with a reference high-rated expert video and maps the likeliness to the final score. We discuss this model in the next chapter.

...