Temporal Action Segmentation in Laparoscopic Surgery Videos: An Evaluation Study

CRINA GUREV, University of Twente, The Netherlands

Temporal Action Segmentation (TAS) is a crucial task in video understanding, aimed at segmenting videos into distinct temporal actions and assigning predefined labels to each segment. This study addresses the challenges of TAS in laparoscopic surgery videos, characterized by lack of data, dynamic transitions and non-static backgrounds, by proposing an unsupervised framework. We utilize ResNet-101 for feature extraction, focusing on both low-level features (e.g., edges, textures) and high-level features (e.g., object semantics, spatial relationships). These features are used to evaluate and compare the clustering performance of two widely used algorithms: Normalized Spectral Clustering (NSC) and Agglomerative Hierarchical Clustering (AHC). The framework is validated on a custom-annotated dataset of laparoscopic surgery videos, using both frame-level and boundary-detection evaluation metrics such as Precision, Recall and F1-Score. This research aims to provide insights into the effectiveness of NSC versus AHC and the impact of low-level versus high-level features in accurately segmenting complex surgical videos, offering a valuable contribution to medical video analysis and training.

Additional Key Words and Phrases: TAS, Frame, Feature extraction, CNN, Clustering, NSC, AHC

1 INTRODUCTION

The rise of minimally invasive surgical techniques, particularly laparoscopic surgery, has revolutionized modern healthcare by offering numerous benefits such as reduced recovery times, minimized scarring, and lower risks of infection compared to traditional open surgery. However, the analysis and interpretation of laparoscopic surgery videos remain a significant challenge due to their dynamic nature and lack of available datasets. To enhance the understanding of laparoscopic surgery videos, temporal action segmentation (TAS) was implemented. TAS involves dividing a video into segments based on distinct actions or events, each labeled with a specific action class [10]. TAS can significantly enhance video understanding by providing detailed distinctions of actions performed during specific segments and detect the frames where surgical events start and end. To tackle the lack of labeled data, we make use of unsupervised learning methods to implement TAS, specifically clustering. Clustering is one of the most widely used technique for TAS implementation, and it was proven to be effective in grouping semantically consistent frames [28], which is why we chose it for the research.

In this study, we implement, evaluate and compare two widely used clustering algorithms for TAS: Normalized Spectral Clustering [29] and Agglomerative Hierarchical Clustering [16], applied to a custom-made dataset of educational laparoscopic surgery videos. Normalized Spectral Clustering uses graph theory to identify non-convex clusters by analyzing the eigenstructure of a similarity matrix, effectively capturing complex frame relationships [29]. On the other hand, Agglomerative Hierarchical Clustering is a bottom-up clustering method that starts by treating each frame as an individual cluster and iteratively merges the most similar clusters based on their visual similarity [20]. In this study, we adapt AHC by using the same similarity matrix as the one used for Normalized Spectral Clustering.

The performance of these clustering algorithms depend on the quality of the features extracted from the video frames. We use the pre-trained Convolutional Neural Networks (CNNs) ResNet-101 [12] for feature extraction. This model has demonstrated exceptional success in various computer vision tasks due to its ability to learn hierarchical feature representations [11]. Furthermore, our goal is to compare the performance of clustering algorithms using low-level features (e.g., background color, texture, edges, lighting, and basic geometric structures) and high-level features (e.g., abstract object representation, object relationships, and global spatial information). Low-level features are extracted from the conv1_relu layer of ResNet-101, while high-level features are derived from the Global Average Pooling (GAP) layer.

Post clustering, we apply a majority vote smoothing technique to refine cluster labels by counting the labels of neighboring frames and choosing the majority one. Finally, we use the following evalution metrics to assess the clustering results: Precision, Recall and F1-Score.

The structure of this paper is organized as follows: **Section 2** provides an overview of the related work, highlighting gaps and methodologies relevant to our research. **Section 3** introduces the research questions that guide the study. **Section 4** details the methodologies used and contains the following subsections: Dataset Creation (4.1), Feature Extraction (4.2), Similarity Matrix Construction (4.3), Clustering Algorithms (4.4) and Majority Vote Smoothing (4.5). **Section 5** covers the implementation process, divided into two subsections: Validation Metrics (5.1) and Implementation Details (5.2). **Section 6** presents the Results (6.1 and 6.2) and discusses Possible Performance Reasons (6.3). Finally, **Section 7** concludes the paper and **Section 8** presents Future Work.

2 RELATED WORK

Temporal Action Segmentation (TAS) has been studied across various domains, including surgery. However, studies where TAS is performed on laparoscopic surgery videos, which handle dynamic transitions, varying backgrounds, and lack of labeled data are limited. This section reviews related work on TAS implementation in surgical videos and other industries, as well as studies the importance of feature selection for video segmentation.

In many TAS studies, domains such as cooking [13], [6], [15], [17], [7] and surgical procedures [23], [27], [24], [22], [30], [14] are

TScIT 42, January 31, 2025, Enschede, The Netherlands

investigated. These studies assume that a limited working area (e.g., cooking table) is shot by a stationary camera, where the background is fixed and has no dynamic transitions, introduced by video editing, such as fades, wipes etc. When reviewing related work on temporal action segmentation for laparoscopic surgery videos, we found out that the studies assume the same stable environments, static backgrounds, focusing mainly on surgical sub-tasks (e.g., cutting, stitching) and not transitions with dynamic backgrounds (e.g., doctor speaking, showcasing the surgical room) [27], [24], [22]. However, many educational laparoscopic videos online feature dynamic transitions and varying backgrounds.

Due to the limited research in this area, labeled data for training models is not available. Efforts to address the lack of labeled data in other domains have incorporated unsupervised learning to group alike segments together. One study done by K. Nakamura et al. [17] discusses how semi-supervised learning, using a hierarchical tree of the category labels to cluster the weakly labeled fragments was used for TAS. Many of the videos contained a lot of diverse scenes, but no dynamic transitions, introduced by video editing. Other studies have addressed challenges such as low resolution and significant motion in videos. For instance, an unsupervised method called R-Clustering was used to segment low-resolution videos with substantial motion, done by E. Talavera et al. [26]. This approach demonstrated how low-level features, which focus on background elements rather than precise objects, can make event boundary detection more precise in such conditions. There is also research done using high-level features [5] to partition videos into clusters by focusing on global spatial relationships and object semantics, which tended to have better results.

To address the limited research on TAS in laparoscopic surgery videos we aim to create our own dataset and perform such study. Given the high resolution (30 fps) of these videos and their varying levels of motion, it remains unclear which type of feature high-level or low-level will yield better clustering performance and which of the popular clustering method is the most appropriate. Therefore, we seek to compare the effectiveness of clustering using both feature types and compare Spectral Clustering to Agglomerative Hierarchical Clustering to identify the optimal approach.

3 PROBLEM STATEMENT

The problem statement, derived from the gaps identified in the relevant work and the uncertainties surrounding existing methodologies, leads to the formulation of the following research questions:

- **RQ1:** How does Normalized Spectral Clustering compare to Agglomerative Hierarchical Clustering in segmenting laparoscopic surgery videos?
- **RQ2:** How do high-level features influence clustering performance compared to low-level features in the context of Temporal Action Segmentation in laparoscopic surgery videos?

4 METHODOLOGY

This section details the methodology used to perform and evaluate clustering algorithms for the temporal segmentation of laparoscopic surgery videos. The research approach is structured into five key stages: dataset creation, feature extraction, similarity matrix construction, clustering algorithms and the implementation of majority vote smoothing. The workflow pipeline can be seen in Figure 4.

4.1 DATASET CREATION

Although we start by explaining dataset creation, this was the third step (Step 3) of our workflow, which can be seen in Figure 4. For the dataset creation 20 educational laparoscopic surgery videos were downloaded from Youtube [2]. The videos can be seen in Table 4 from the Appendix. The selected videos have an average frame rate of 30 frames per second (fps). The total duration of all 20 videos is 100.39 minutes, with an average length of 5.02 minutes per video. The videos were downsampled to 5 fps and segmented into individual frames. Each frame was saved as an individual image in the JPEG format in directories organized by video. To preserve the chronological order of the frames, each frame was labeled using a zero-padded sequential format (e.g., `frame_0001.jpg`). The frame extraction process was automated using a custom video processing pipeline developed with the OpenCV library [1].

Following the fragmentation process, each frame was manually annotated with corresponding event labels based on visual context. From the formed dataset, we observed that the number of events vary across each video. Figure 1 illustrates the average duration of each event per video in minutes. The total set of unique events observed across all videos combined is as follows:

1. Showcasing Surgical Instruments (Appendix Figure 5)
2. Surgery External View (Appendix Figure 6)
3. Surgery Internal View (Appendix Figure 8)
4. Doctor Speaking (Appendix Figure 7)
5. Showcasing the Surgical Room (Appendix Figure 13)
6. Informative Slide (Appendix Figure 14)
7. Blank White Background (Appendix Figure 11)
8. Blank Black Background (Appendix Figure 11)
9. Animation (Appendix Figure 9)
10. Surgery Combined View (Appendix Figure 12)
11. Undetermined (Appendix Figure 10)

Since the videos were edited, they exhibit a significant number of dynamic transitions between frames. The transitions which we identified are of two types: abrupt (Figure 2) and gradual transitions (Figure 3). Abrupt transitions, referred to as hard cuts, involve an instantaneous change from one scene to another. In abrupt event transitions, frame X represents one event and frame $X + 1$, represents another event. Due to their clear-cut nature, abrupt transitions are easy to identify and manually annotate. In the case of gradual event transitions, it is much harder to identify the borders of the events. These transitions progress over several frames and include effects such as fade-ins, fade-outs, dissolves, and wipes. The lack of distinct visual boundaries, makes it difficult to determine the exact frame at which one event ends and another begins [4]. To address this challenge, two cosine similarity matrices were constructed, one based on high-level features and the other on low-level feature, and used to annotate the ambiguous frames. The feature similarity matrix, discussed in detail in Section 4.3, shows the pairwise similarity between consecutive frames based on extracted feature vectors.

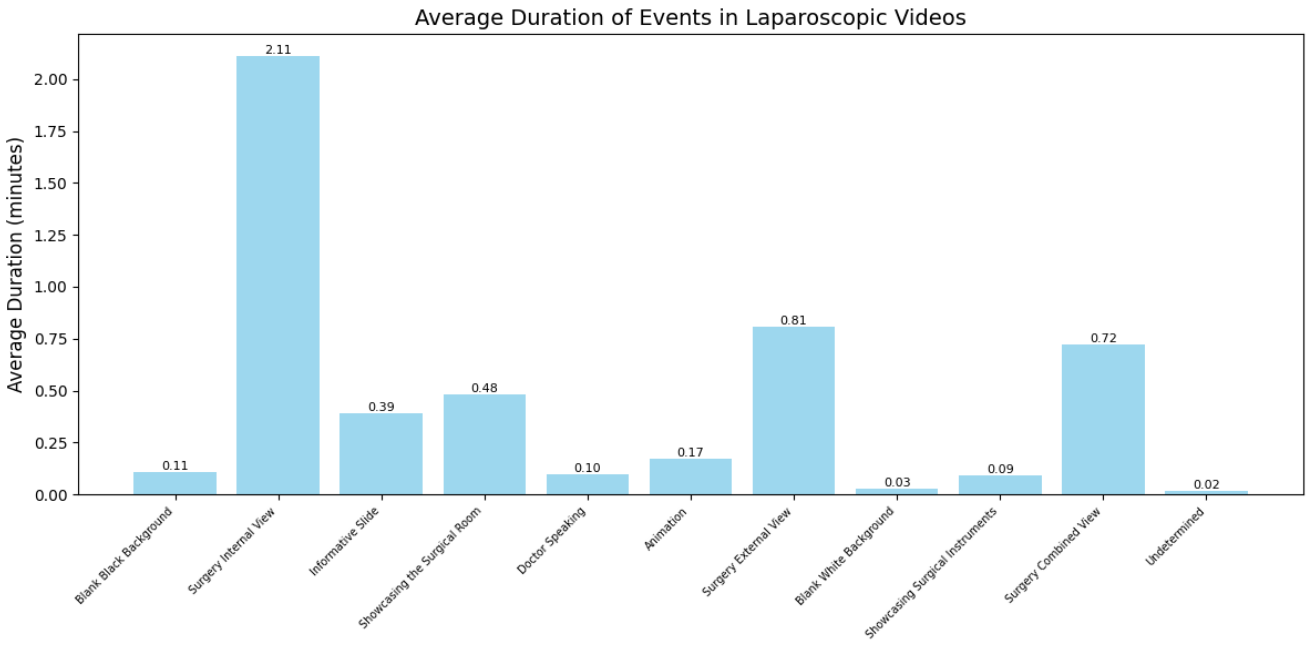


Fig. 1. Average duration of each event per video in minutes

To make annotation easier we created a function which iteratively compares similarity scores across frames undergoing gradual transitions.

Let frame X belong to event A and frame $X + n$, where $n \in \mathbb{N}^+$, belong to event B. The frames in the interval $(X, X + n)$ contain mixed visual features of both events A and B. We define the set of these intermediate frames as:

$$M = \{i \mid X < i < X + n,\ n \in \mathbb{N}^+\}$$

For each frame $i \in M$, the function calculates two similarity scores to determine the event classification:

- $\text{Sim}(i, X)$: Similarity between frame i and frame X.
- $\text{Sim}(i, X + n)$: Similarity between frame i and frame $X + n$.

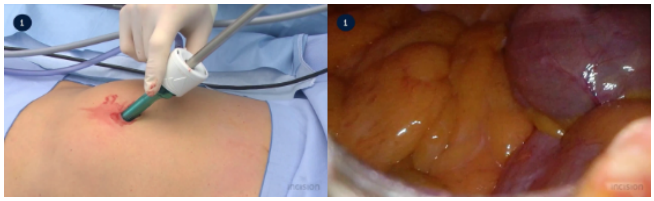
The classification decision is made as follows:

1. If $\text{Sim}(i, X) = \text{Sim}(i, X + n)$, then frame i is assigned to $\arg\min_{j \in \{X, X+n\}} |i - j|$, which corresponds to the closest event frame.
2. Else if $\text{Sim}(i, X) > \text{Sim}(i, X + n)$, then frame i is assigned to event A.
3. Otherwise, frame i is assigned to event B.

Once all frames were annotated, a table was created to make clustering evaluation easier, Table 1. The table has 4 columns: Video number (unique identifier), Start Frame, End Frame and Events. Two separate datasets were constructed: one annotated using the low-level similarity matrix and the other using the high-level similarity matrix. These datasets were not used for training any supervised models but served as benchmarks for evaluating the clustering techniques.

Video Number	Start Frame	End Frame	Event
1	0	100	Informative Slide
1	100	200	Performing Surgery – External Patient View
2	0	10	Blank Background

Table 1. Example rows from the dataset



Frame X (left) and Frame X+1 (right)

Fig. 2. Example of an abrupt transition

Frame X (left), Frame X+1 (middle) and Frame X+2 (right)

Fig. 3. Example of a gradual transition

4.2 Feature Extraction

To address RQ2 and implement clustering, as well as annotate dynamic event transitions, it is necessary to compute the distances between feature vectors. For this purpose we extracted features, which is the first step (Step 1) of our workflow (Figure 4), using pre-trained Convolutional Neural Network (CNN), specifically ResNet-101 [11]. The model was trained on the ImageNet dataset [9], which has been widely used for transfer learning in various visual recognition tasks, including scene classification and image retrieval [26].

For feature extraction we used two layers, conv1_relu for low-level features and Global Average Pooling (GAP) for high-level features. These two layers are shown in Appendix, Figure 15 with arrows pointing towards them. The conv1_relu layer is the first layer in the network after the input image is processed by the convolution, BatchNorm and activation [11]. The GAP layer is the last one, and focuses on complex scenes by integrating object semantics with global spatial information [21].

- **Extracted Low-Level Features from conv1_relu layer**: color and lightening across the image, texture (patterns and structures formed by the spatial arrangement of pixel intensities), edges and contours (sharp transitions between regions), basic geometric shapes (elements such as lines and curves present) [18]. The Low-Level Features are computed from the background rather than the foreground, indicating their focus on background elements [8].
- **High-Level Features from GAP layer**: object representation within the image, scene semantics (interactions of objects with the environment), global spatial information (summary of the scene, without focusing on specific object locations).

Each image was individually processed by resizing it to 224×224 pixels, followed by normalization using model-specific preprocessing functions (`preprocess_input`). This preprocessing scales the pixel values to match the distributions on which the model was originally trained with the ImageNet dataset. The extracted feature vectors were flattened and stored as `.npy` files. To maintain the chronological sequence of the frames consistent with the frame fragmentation process, each feature vector was systematically labeled (e.g., `frame_0001.jpg.npy`). Consequently, the total number of generated `.npy` files directly corresponds to the number of fragmented frames extracted from each video. For every video two directories are created, one containing the low-level features and the other containing the high-level ones. Each `.npy` file corresponding to a video was sorted in ascending order and aggregated into a unified NumPy array for processing.

4.3 Similarity matrix construction

Two similarity matrices were constructed per video, so the frame-level similarity can be measured based on the two types of features: low-level and high-level. Similarity matrix construction is the second (Step 2) of the workflow, and examples of such matrices can be seen in (Figure 4). The cosine similarity metric was selected due to its effectiveness in high-dimensional spaces. Cosine similarity measures the cosine of the angle between two non-zero vectors, capturing their orientation regardless of magnitude:

$$\text{Similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

where A and B represent feature vectors, $A \cdot B$ is the dot product, and $\|A\|$ and $\|B\|$ denote their magnitudes. The computed similarity values were normalized to the $[0, 1]$ range to ensure compatibility with clustering algorithms.

4.4 Clustering Algorithms

In this section, we describe the implementation of the two clustering methods: Normalized Spectral Clustering and Agglomerative Hierarchical Clustering (AHC). Clustering is the fourth step (Step 4) of the workflow (Figure 4)

4.4.1 Normalized Spectral Clustering. For this study we chose Jordan-Weiss normalization for the Laplacian matrix during Spectral Clustering. This choice was motivated by its proven effectiveness in producing well-separated clusters and its ability to enhance the robustness of clustering when data distributions are non-uniform or contain noise [19].

The Jordan-Weiss normalization modifies the Laplacian matrix as follows:

$$L_{\text{JW}} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2}$$

- W is the similarity matrix.
- D is the degree matrix, a diagonal matrix where $D_{ii} = \sum_{j=1}^{n} W_{ij}$.
- I is the identity matrix.

4.4.2 Agglomerative Hierarchical Clustering. We selected Agglomerative Hierarchical Clustering (AHC) with average linkage due to its ability to provide flexible cluster merging criteria by considering the average distance between all pairs of data points in two clusters [25]. The average linkage method determines the distance between two clusters C_r and C_k as the average of all pairwise distances between points in C_r and C_k. The formula is expressed as:

$$d(C_r, C_k) = \frac{1}{|C_r| \cdot |C_k|} \sum_{x \in C_r} \sum_{y \in C_k} d(x, y)$$

- $|C_r|$ and $|C_k|$ represent the number of elements in clusters C_r and C_k,
- $d(x, y)$ is the distance between data points $x \in C_r$ and $y \in C_k$, typically measured using Euclidean distance.

4.5 Majority Vote Smoothing

To improve the clustering results, we implemented a post-processing technique called majority vote smoothing. This method refines the cluster labels by considering the labels of neighboring frames within a defined window. The goal is to address label inconsistencies caused by noise in the features.

A sliding window of size win_len is centered on each frame in the video. The labels of the neighboring frames within the window are collected. The most frequently occurring label (majority vote) is assigned to the current frame. In cases of ties, the label of the closest neighbor is chosen. The algorithm is formally expressed as:

$$L'(i) = \arg\max \left(\text{Count} \left(L(j) \;\middle|\; j \in \left[i - \frac{\text{win\_len}}{2}, i + \frac{\text{win\_len}}{2} \right] \right) \right)$$

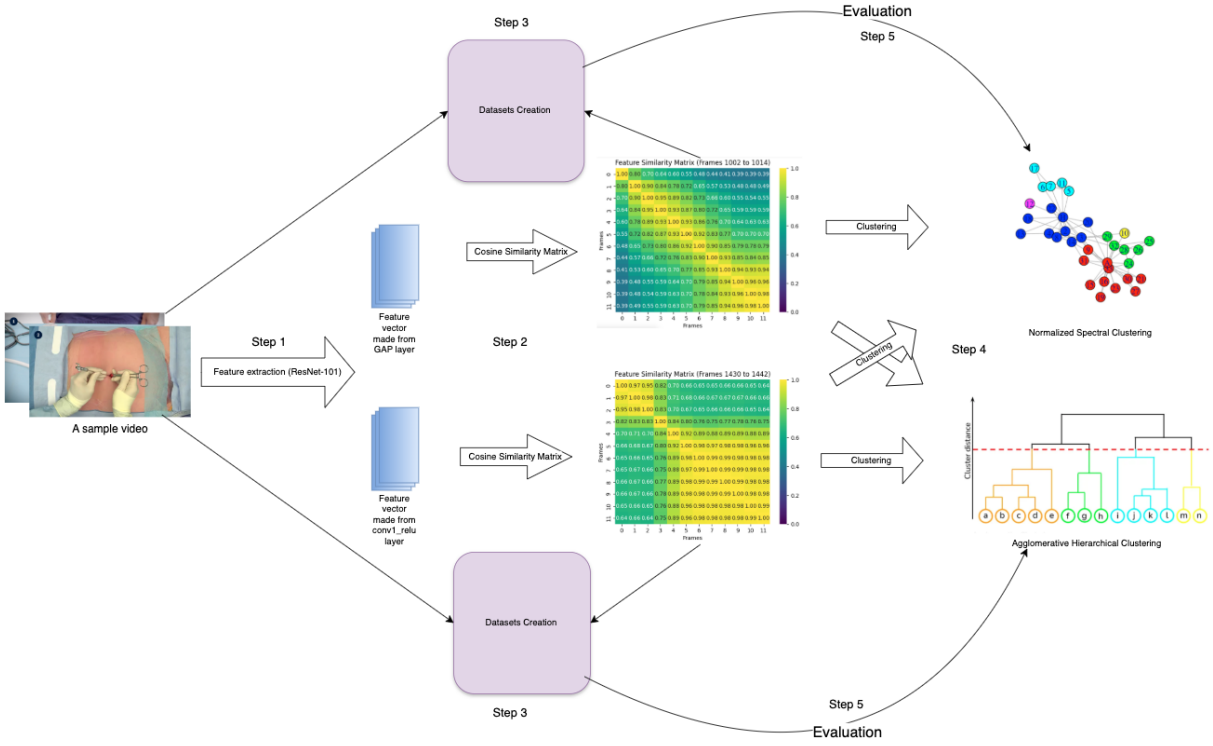


Fig. 4. Workflow Pipeline for Temporal Action Segmentation

where $L'(i)$ is the refined label of frame i, $L(j)$ represents the label of a neighboring frame, and win_len is the window size.

This technique was applied to the clustering results for both low-level and high-level features, ensuring greater local consistency within consecutive frames.

5 IMPLEMENTATION

In this section we discuss implementation details and describe the metrics we use to evaluate the framework.

5.1 Validation Metrics

The metrics are categorized into two groups: Frame-Level Clustering Metrics for assessing overall segmentation quality and Boundary Detection Metrics for evaluating the accuracy of detecting event transitions.

5.1.1 Frame-Level Metrics. Frame-level metrics evaluate how well the clustering algorithm segments the video frames in alignment with the ground truth annotations.

- **Precision (P)** measures the proportion of true positive identifications among all instances that were retrieved. In the context of clustering, it quantifies how many of the data points assigned to a cluster actually belong to the corresponding ground truth class [3].
- **Recall (R)** assesses the proportion of true positive identifications among all instances that should have been retrieved. For clustering, it indicates how well the algorithm captures all the data points of a particular ground truth class within a single cluster [3].
- **F1-Score (F1)** also known as the F-measure is the harmonic mean of precision and recall. It is useful when the distribution of classes is imbalanced [3].

5.1.2 Boundary Detection with ±5 Frame Tolerance. To account for minor misalignments in event transition detection, we introduce a ±5 frame boundary tolerance in our evaluation. We use the following metrics for the evaluations: Boundary Precision (measures the accuracy of detected boundaries), Boundary Recall (measures how effectively the model detects true event boundaries) and Boundary F1-Score (provides a balanced evaluation of boundary detection).

5.2 Implementation Details

The research was conducted using Google Colab Pro using the Tesla T4 GPU (15 GB GPU RAM) and High RAM mode (51.0 GB System RAM). These resources were needed, because the session would crash on lower resources. All the experiments were conducted using Python 3. The following libraries were used: PyTorch, Scikit-learn, NumPy, SciPy and Tensorflow.

The following pipeline was used (Figure 4):

- Feature extraction
- Similarity matrix construction
- Dataset creation

- Clustering: for each of the two selected clustering algorithms, two sets of experiments were conducted: one using the similarity matrix derived from the features of the GAP layer and the other using the matrix generated from the features of the conv1_relu layer.
- Majority vote smoothing: for this algorithm we chose a window length (win_len) of 5.

6 RESULTS

In this section we present and discuss the results. Table 2 presents clustering frame-level results and Table 3 displays the results for boundary detection with with ±5 frame tolerance.

Average values		Precision	Recall	F1
High-level features	NSC	0.3355	0.2505	0.2901
	AHC	0.2574	0.2203	0.2371
Low-level features	NSC	0.3034	0.1877	0.2323
	AHC	0.1971	0.1841	0.1901

Table 2. Clustering Frame-Level Results

Average values		Precision	Recall	F1
High-level features	NSC	0.5399	0.7024	0.6105
	AHC	0.8074	0.7388	0.7716
Low-level features	NSC	0.3489	0.5921	0.4391
	AHC	0.6304	0.5396	0.5815

Table 3. Boundary Detection Results with ±5 Frame Tolerance

6.1 Answer to RQ1: Performance Comparison of Clustering Algorithms

The performance of the two clustering algorithms, Normalized Spectral Clustering (NSC) and Agglomerative Hierarchical Clustering (AHC), was evaluated using high-level and low-level features, as shown in Tables 2 and 3.

For frame-level clustering, the results from Table 2 indicate that Normalized Spectral Clustering outperformed Agglomerative Hierarchical Clustering across all metrics for both high-level and low-level features. For high-level features NSC has an F1 score of 0.0530 or 22.35% higher than AHC. Similarly, for low-level features, NSC's F1 score exceeded AHC's by 0.0422, or 22.19%.

When considering boundary detection with ±5 frame tolerance, the results are presented in Table 3, AHC outperformed SCN across both feature types and all metrics. AHC demonstrated a 26.39% (0.1611) higher Boundary F1-Score than NSC for high-level features and a 32.43% (0.1424) improvement for low-level features.

Boundary detection with a ±5 frame tolerance has a much higher F1 score than frame-level clustering. This suggests that the models are better at detecting event transitions than accurately classifying all frames into clusters.

Finally, the results indicate that Normalized Spectral Clustering is better at classifying frames within the clusters, while Agglomerative Hierarchical Clustering is better at detecting event transitions.

6.2 Answer to RQ2: Impact of Feature Types on Clustering Performance

In terms of frame-level clustering, high-level features resulted in better average scores compared to low-level features for both NSC and AHC, as shown in Table 2, 3. For NSC, clustering with high-level features achieved an F1 score that was 0.2901 higher, representing a 24.88% improvement over low-level features. Similarly, for AHC, clustering with high-level features resulted in an F1 score that was 0.047 higher, or 24.72%, than low-level features. For boundary event detection, high-level features yielded better scores as well. For NSC, detecting events using high-level features achieved an F1 score that was 0.1714 higher, showing a 39.04% increase compared to low-level features. For AHC, high-level features achieved an F1 score improvement of 0.1901, or 32.68%, over low-level features.

This results indicate that high-level features, which capture object semantics and global spatial information, were more effective at clustering frames and identifying events transitions in laparoscopic surgery videos. This finding might suggest that laparoscopic surgery videos have on average less motion and clear distinguishable objects, that do not change as much from frame to frame. The high-level features also capture the relationship between objects (e.g, hand holding instrument, instrument touching organ), which might be one of the reason high-level features performed better.

6.3 POSSIBLE PERFORMANCE REASONS

The performance observed in this study can be attributed to several factors related to the dataset characteristics, feature extraction, and clustering methodologies.

Inaccurate Dataset Annotation. One potential reason for the lower performance in frame-level clustering could be inaccuracies in dataset annotation. The number of events chosen might not reflect the true number present in the videos. Additionally, frames belonging to the same action but with drastically different visual features (e.g., due to changes in camera angles, lighting, or present objects) might need to be labeled as different events.

Suboptimal Feature Extractor. The choice of ResNet-101 as the feature extractor, might not be the most suitable for laparoscopic surgery videos. Since the most common event in the dataset is "Surgery Internal View" (Figure 8), a CNN pre-trained more on medical datasets that contain images of internal organs, might give better results.

Majority Vote Smoothing. The majority vote smoothing algorithm used in this study could be further improved. Currently, it considers only the majority labels of neighboring frames. A more advanced approach could combine the neighbor labels with the similarity scores from the similarity matrix. This hybrid method could improve frame-level clustering performance.

7 CONCLUSION

In conclusion, this study evaluated the effectiveness of unsupervised clustering methods for Temporal Action Segmentation (TAS)

in laparoscopic surgery videos, specifically Normalized Spectral Clustering (NSC) and Agglomerative Hierarchical Clustering (AHC) on a hand-made dataset. Using ResNet-101 for feature extraction, we analyzed the impact of both high-level and low-level features on clustering performance and boundary detection.

The results showed that high-level features consistently outperformed low-level features in clustering frames and detecting boundary events, which might indicate that laparoscopic surgery videos likely exhibit minimal motion and feature consistent objects that persist across frames.

When it comes to frame-level clustering, NSC demonstrated better performance, achieving higher Precision, Recall, and F1 scores, while AHC excelled in boundary detection with a ±5 frame tolerance, indicating its strength in identifying event transitions.

These findings show the importance of feature representation and algorithm selection in TAS for medical video analysis and can help with medical training and provide insights for future work in this domain.

Several ideas for future research can be explored to improve the accuracy of our TAS framework. In the next section we discuss these ideas.

8 FUTURE WORK

Even if this study offers valuable insights into TAS for laparoscopic surgery videos, there are several ideas for future exploration.

Dataset Expansion: Future research could include a larger and more diverse dataset, covering a larger range of surgical procedures and events to improve the generalizability of the findings.

Different Feature Extractor: Exploring alternative feature extractors, such as EfficientNet, Vision Transformers, etc., or fine-tuning the selected model, ResNet-101 on surgical video data, could improve the quality of extracted features. Fine-tuning the model would likely lead to improved clustering performance.

Different Clustering Technique: Investigating alternative clustering techniques, such as DBSCAN, k-means with advanced distance metrics, or Gaussian Mixture Models, would offer better insights on which unsupervised technique performs the best.

Different Dataset Creation Technique: Refining the dataset creation process by exploring more accurate and automated annotation techniques. Possibly combining similarity matrix with metrics like color histogram, texture etc.

Implementing these future research ideas can improve the performance of the framework and offer more valuable insights.

9 ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Estefanía Talavera Martínez, whose help and guidance during this project, made the completion of this research possible.

10 ADDITIONAL TOOLS

During the preparation of this work, the author used ChatGPT to make LateX tables and figures, as well as correct grammatical errors in it. After using this tool, the author reviewed and edited the content and takes full responsibility for the work.

REFERENCES

[1] 2000. OpenCV. https://opencv.org/. Open source computer vision and machine learning software library..
[2] 2005. YouTube. https://www.youtube.com.
[3] 2018. Springer. https://link.springer.com/content/pdf/10.1007/978-3-319-78503-5_6.pdf
[4] Sadiq H. Abdulhussain, Abd Rahman Ramli, M. Iqbal Saripan, Basheera M. Mahmmod, Syed Abdul Rahman Al-Haddad, and Wissam A. Jassim. 2018. Methods and Challenges in Shot Boundary Detection: A Review. *Entropy* 20, 4 (2018), Article 214. https://www.researchgate.net/publication/323956580_Methods_and_Challenges_in_Shot_Boundary_Detection_A_Review
[5] Wasfi G. Al-Khatib. 1999. Semantic Modeling and Knowledge Representation in Multimedia Databases. *IEEE Transactions on Knowledge and Data Engineering* 11, 1 (1999), 64–80. https://www.researchgate.net/figure/Semantic-modeling-of-video-data_fig5_229022767
[6] S. Bianco, G. Ciocca, and P. Napoletano. 2014. On the Use of MKL for Cooking Action Recognition. In *Image Processing: Machine Vision Applications VII*, K. S. Niel and P. R. Bingham (Eds.), Vol. 9024. International Society for Optics and Photonics, 90240G. https://doi.org/10.1117/12.2041939
[7] S. Bianco, G. Ciocca, P. Napoletano, R. Schettini, R. Margherita, G. Marini, and G. Pantaleo. 2013. Cooking Action Recognition with IVAT: An Interactive Video Annotation Tool. In *Image Analysis and Processing – ICIAP 2013 (Lecture Notes in Computer Science, Vol. 8157)*, A. Petrosino (Ed.). Springer, Berlin, Heidelberg, 631–641. https://doi.org/10.1007/978-3-642-41184-7_64
[8] Sébastien M. Crouzet and Thomas Serre. 2011. What Are the Visual Features Underlying Rapid Object Recognition? *Frontiers in Psychology* 2 (2011), 326. https://www.frontiersin.org/articles/10.3389/fpsyg.2011.00326/full
[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. http://www.image-net.org
[10] Guodong Ding, Fadime Sener, and Angela Yao. 2022. Temporal Action Segmentation: An Analysis of Modern Techniques. *arXiv preprint* 2210 (2022). https://arxiv.org/pdf/2210.10352
[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 770–778. https://arxiv.org/abs/1512.03385
[12] Qingsheng Jiang, Dapeng Tan, Yanbiao Li, and Qiming Zheng. 2019. Object Detection and Classification of Metal Polishing Shaft Surface Defects Based on Convolutional Neural Network Deep Learning. *Applied Sciences* 9, 24 (2019), 5470. https://www.researchgate.net/figure/ResNet-101-feature-extraction_fig1_338087022
[13] J. Lei, X. Ren, and D. Fox. 2012. Fine-grained Kitchen Activity Recognition Using RGB-D. In *Proceedings of the ACM Conference on Ubiquitous Computing*. ACM. https://www.researchgate.net/publication/262237442_Fine-grained_kitchen_activity_recognition_using_RGB-D
[14] Y. Li, Z. Zhao, R. Li, and F. Li. 2024. Deep Learning for Surgical Workflow Analysis: A Survey of Progresses, Limitations, and Trends. *Artificial Intelligence Review* 57 (2024), Article 291. https://link.springer.com/article/10.1007/s10462-024-10929-6
[15] S. Michibata, K. Inoue, M. Yoshioka, and A. Hashimoto. 2020. Cooking Activity Recognition in Egocentric Videos with a Hand Mask Image Branch in the Multi-Stream CNN. In *Proceedings of the 12th Workshop on Multimedia for Cooking and Eating Activities (CEA '20)*. 1–6. https://doi.org/10.1145/3379175.3391712
[16] Fionn Murtagh and Pedro Contreras. 2011. Methods of Hierarchical Clustering. In *Studies in Classification, Data Analysis, and Knowledge Organization*. Springer, 288–297. https://link.springer.com/referenceworkentry/10.1007/978-3-642-04898-2_288
[17] K. Nakamura, N. Nitta, N. Babaguchi, K. Fujii, S. Matsumura, and E. Nabata. 2021. Semi-Supervised Temporal Segmentation of Manufacturing Work Video by Automatically Building a Hierarchical Tree of Category Labels. *IEEE Access* 9 (2021), 68017–68027. https://ieeexplore.ieee.org/document/9420050
[18] Georgios Nanos. 2024. Low-Level vs. High-Level Features in Computer Vision. Baeldung. https://www.baeldung.com/cs/cv-low-vs-high-level-features Reviewed by Michal Aibin. Last updated: March 18, 2024..
[19] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2002. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems*, Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani (Eds.), Vol. 14. MIT Press, 849–856. https://papers.neurips.cc/paper/2092-on-spectral-clustering-analysis-and-an-algorithm.pdf
[20] Eric Uchenna Oti and Michael O. Olusola. 2024. Overview of Agglomerative Hierarchical Clustering Methods. *British Journal of Computer Networking and Information Technology* 7, 2 (June 2024), 14–23. https://doi.org/10.52589/BJCNIT-CV9POOGW
[21] Mohammad Javad Parseh, Mohammad Rahmanimanesh, and Zohreh Azimifar. 2023. Scene Representation Using a New Two-Branch Neural Network Model. *The Visual Computer* 40 (December 2023), 6219–6244. https://link.springer.com/

article/10.1007/s00371-023-03162-9
[22] M. J. Primus, K. Schoeffmann, and L. Böszörmenyi. 2016. Temporal Segmentation of Laparoscopic Videos into Surgical Phases. In *2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI)*. IEEE, 1–6. https://ieeexplore.ieee.org/document/7500249
[23] Y. Qin, S. A. Pedram, S. Feyzabadi, M. Allan, A. J. McLeod, J. W. Burdick, and M. Azizian. 2020. Temporal Segmentation of Surgical Sub-Tasks through Deep Learning with Multiple Data Sources. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. https://arxiv.org/abs/2002.02921
[24] G. Quellec, M. Lamard, B. Cochener, and G. Cazuguel. 2014. Real-Time Segmentation and Recognition of Surgical Tasks in Cataract Surgery Videos. *IEEE Transactions on Medical Imaging* 33, 12 (2014), 2352–2360. https://pubmed.ncbi.nlm.nih.gov/25055383/
[25] Robert R. Sokal and Charles D. Michener. 1958. A Statistical Method for Evaluating Systematic Relationships. In *University of Kansas Scientific Bulletin*, Vol. 38. University of Kansas, 1409–1438. https://www.jstor.org/stable/2427238
[26] Estefania Talavera, Mariella Dimiccoli, Marc Bolaños, Maedeh Aghaei, and Petia Radeva. 2017. R-Clustering for Egocentric Video Segmentation. *arXiv preprint* (2017). https://arxiv.org/pdf/1704.02809
[27] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy. 2017. EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos. *IEEE Transactions on Medical Imaging* 36, 1 (2017), 86–97. https://ieeexplore.ieee.org/document/7500249
[28] Arash Vahdat, Kevin Cannons, and Greg Mori. 2013. Compositional Models for Video Event Detection: A Multiple Kernel Learning Latent Variable Approach. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE. https://www.cv-foundation.org/openaccess/content_iccv_2013/papers/Vahdat_Compositional_Models_for_2013_ICCV_paper.pdf
[29] Ulrike von Luxburg. 2007. A Tutorial on Spectral Clustering. *Statistics and Computing* 17, 4 (2007). https://arxiv.org/pdf/0711.0189
[30] B. Zhang, B. Goel, M. H. Sarhan, V. K. Goel, R. Abukhalil, B. Kalesan, N. Stottler, and S. Petculescu. 2023. Surgical Workflow Recognition with Temporal Convolution and Transformer for Action Segmentation. *International Journal of Computer Assisted Radiology and Surgery* 18 (2023), 785–794. https://link.springer.com/article/10.1007/s11548-022-02811-z

A APPENDIX

Table 4. Total Number of Events in Each Video

Video URL	Total Number of Events
https://www.youtube.com/watch?v=sPyZRkkxqNs&t=210s	6
https://www.youtube.com/watch?v=JV4oKUW8kRM&t=2s	6
https://www.youtube.com/watch?v=_aF42rYVyng&t=4s	5
https://www.youtube.com/watch?v=xxVmD6zx2Gg&t=10s	5
https://www.youtube.com/watch?v=I60p0etn6p4&t=72s	2
https://www.youtube.com/watch?v=PqFtJnSPDdc&t=1s	5
https://www.youtube.com/watch?v=Yhqm9aUz2Sg	3
https://www.youtube.com/watch?v=eqmi2ns7CKo	5
https://www.youtube.com/watch?v=VfT--J-MZlM&t=7s	2
https://www.youtube.com/watch?v=NwTEWycmB_o&t=1s	5
https://www.youtube.com/watch?v=lLMGXhaOKjo&t=141s	2
https://www.youtube.com/watch?v=68a3nvcXTA0&t=1s	3
https://www.youtube.com/watch?v=cLl275lch3o	3
https://www.youtube.com/watch?v=Q0ESQHxoHlk	8
https://www.youtube.com/watch?v=8oGgBvVUFeg	2
https://www.youtube.com/watch?v=vuNxdwsneWI&t=54s	5
https://www.youtube.com/watch?v=FE23x3jHMHg&t=1s	3
https://www.youtube.com/watch?v=IAqfB78umAM	5
https://www.youtube.com/watch?v=boy4Gp51JMI	3
https://www.youtube.com/watch?v=JH-Yx1QWUBk	2

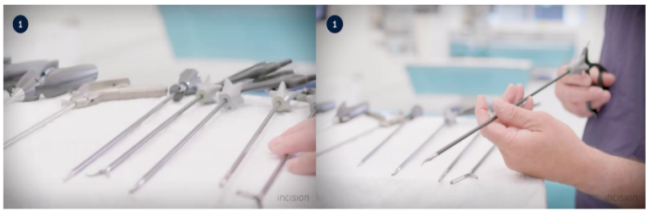


Fig. 5. Example of the event Showcasing Surgical Instruments

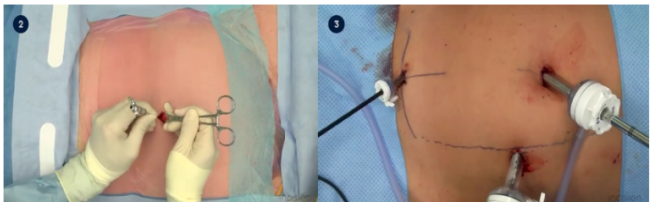


Fig. 6. Example of the event Surgery External View

Fig. 7. Example of the event Doctor Speaking

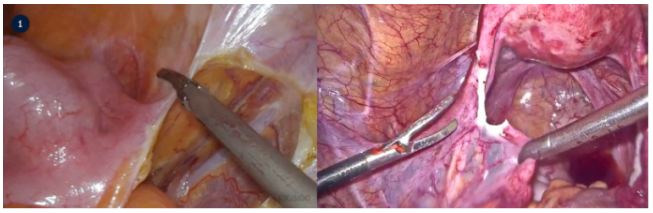


Fig. 8. Example of the event Surgery Internal View

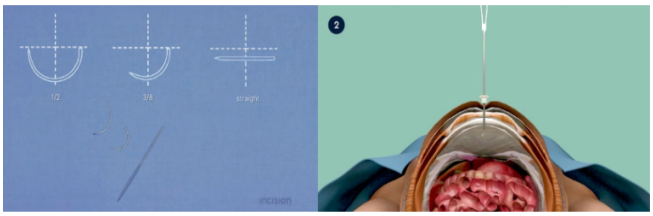


Fig. 9. Example of the event Animation

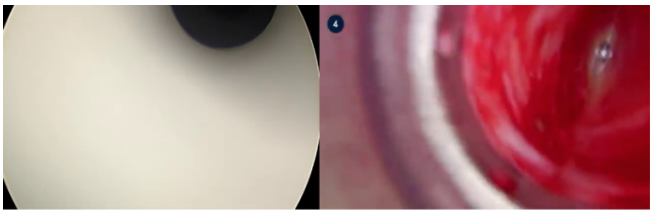


Fig. 10. Example of the event Undetermined

Fig. 11. Example of the event Blank White/Black Background

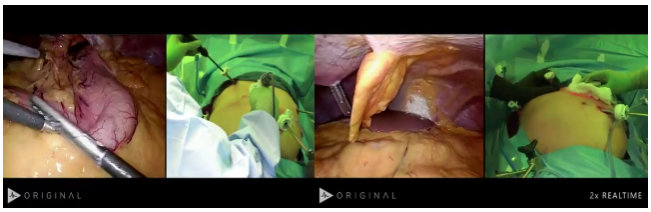


Fig. 12. Example of the event Surgery Combined View

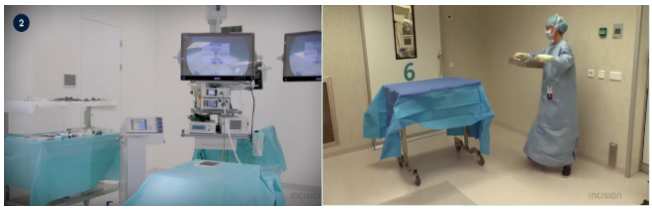


Fig. 13. Example of the event Showcasing the Surgical Room

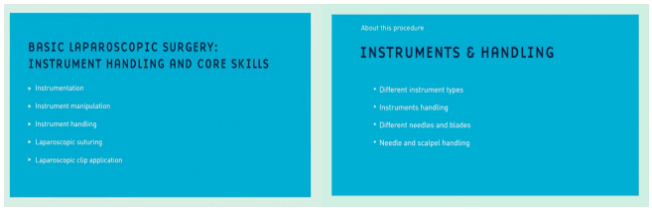


Fig. 14. Example of the event Informative Slide

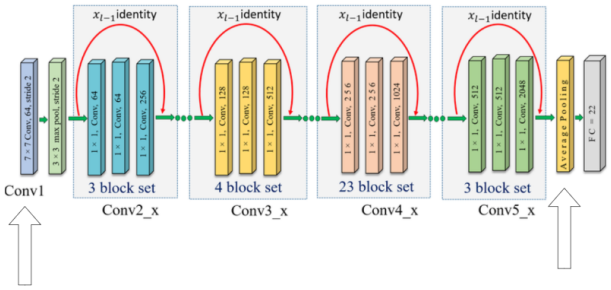


Fig. 15. ResNet-101 architecture