TECHNISCHE UNIVERSITÄT WIEN

ACIN
AUTOMATION & CONTROL INSTITUTE
INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

Unsupervised Spatio-Temporal Action Boundary Localization (UnSTABL) and Summary Generation in Surgical Environments

DIPLOMARBEIT

Conducted in partial fulfillment of the requirements for the degree of a

Diplom-Ingenieur (Dipl.-Ing.)

supervised by

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. M. Vincze
DI (FH) Dr.techn. Heindl Christoph, Profactor GmbH
DI (FH) Bauer Harald, Profactor GmbH

submitted at the

TU Wien
Faculty of Electrical Engineering and Information Technology
Automation and Control Institute

by
Stefan Kuen
[redacted]

Vienna, July 2024

Vision for Robotics Group
A-1040 Wien, Gusshausstr. 27, Internet: http://www.acin.tuwien.ac.at

Abstract

Understanding the actions and context within a video comes naturally to human observers. However, replicating this ability through artificial intelligence to automate the time-consuming manual video analysis in areas like security and healthcare remains a challenging task in computer vision. While most existing video understanding algorithms try to localize and classify all actions within a video, they often depend on heavily annotated datasets or can not deal with the complexities found in multi-person environments. Consequently, in many real-world environments like the operating room (OR), featuring multiple individuals performing concurrent actions and experiencing frequent occlusions - and where public datasets are scarce due to the sensitive nature of the video content - a significant gap remains in automatic action detection and video summary generation.

This thesis addresses these gaps with two key contributions. First, it presents the **Un**supervised **S**patio-**T**emporal **A**ction **B**oundary **L**ocalization (**UnSTABL**) framework, which leverages person-specific action information to localize significant action boundaries in an unsupervised manner. By focusing on each individual independently, the framework is able to effectively handle multi-person environments. Secondly, it improves the base framework specifically for the challenging OR environment. This "collision-robust" framework successfully handles brief person overlaps during boundary detection. It additionally improves the accuracy of a state-of-the-art person tracker by detecting and correcting ID swaps using the previously extracted action information.

To evaluate the performance of our proposed contributions, we conduct a two-step validation process. Using two datasets, the UnSTABL framework is initially benchmarked against existing unsupervised action boundary detection methods. This benchmark establishes a performance baseline in single-person environments, revealing that our framework is able to identify action boundaries with state-of-the-art accuracy. It successfully detects ground-truth action segments of various durations, ranging from several seconds up to almost a minute, showing a high flexibility in action length.

In the second part of the evaluation, we perform a qualitative assessment of the framework's performance in the complex OR setting to verify the benchmark results in multi-person environments. We assess the accuracy of the detected action boundaries, the improvements in person tracking, and identify the limitations of our proposed framework. While our person-specific approach proved effective in moderately crowded scenes, delivering similar results as in both benchmarks, densely crowded and collaborative tasks reveal certain limitations. Due to continuous, long person overlaps, neither framework reliably detects action boundaries in these scenarios.

Despite these limitations, the framework demonstrates strong results in moderately crowded scenes, making unsupervised action boundary detection feasible in multi-person environments without sacrificing accuracy. Additionally, the proposed ID Swap Correction Module is able to correct about 50% of the tracker's incorrect ID Swaps, successfully improving the tracking accuracy in this challenging setting.

Kurzzusammenfassung

Menschen erfassen den Kontext und die Handlungen in Videos intuitiv und schnell. Die Entwicklung intelligenter, automatisierter Systeme, die in der Lage sind, die zeitaufwändige manuelle Videoanalyse in Bereichen wie dem Gesundheitswesen oder der Überwachung zu ersetzen, bleibt jedoch eine große Herausforderung im Bereich der Computer Vision.

Aktuelle Algorithmen im Bereich der automatisierten Videoanalyse zielen darauf ab, alle Aktionen im Video zu lokalisieren und zu klassifizieren. In den meisten Fällen benötigen diese Algorithmen jedoch große Datensätze, um ihre Netzwerke zu trainieren, oder sie scheitern an der Komplexität von Mehrpersonenumgebungen. In hochkomplexen Umgebungen wie dem Operationssaal (OP), wo öffentlich zugängliche Datensätze aufgrund ihres sensiblen Inhalts praktisch nicht existieren, besteht daher eine erhebliche Forschungslücke im Bereich der automatisierten Aktionserkennung und Videozusammenfassung.

Diese Masterarbeit schließt diese Lücke durch zwei wesentliche Beiträge: Erstens zeigt das entwickelte UnSTABL-Framework, dass es möglich ist, signifikante Aktionsübergänge in komplexen Ein- und Mehrpersonenumgebungen auf der Basis von personenspezifischen Aktionsinformationen zu identifizieren, ohne dass Trainingsdaten zur Verfügung stehen. In einem zweiten Schritt wird das Basis-Framework speziell für die anspruchsvolle OP-Umgebung weiterentwickelt. Das "kollisionsrobuste" Framework ist in der Lage, kurze Überlappungen von Personen bei der Aktionserkennung zu berücksichtigen. Darüber hinaus verbessert es die Genauigkeit eines State-of-the-Art-Personentrackers, indem es ID-Verwechslungen anhand der extrahierten Aktionsinformationen erkennt und korrigiert.

Zur Bewertung des Frameworks wird ein zweistufiger Evaluierungsprozess durchgeführt. Zunächst wird das UnSTABL-Framework anhand von zwei Benchmark-Datensätzen mit bestehenden State-of-the-Art-Methoden verglichen. Dabei können wir zeigen, dass unser Framework in der Lage ist, Handlungsgrenzen in Ein-Personen-Umgebungen mit der gleichen State-of-the-Art-Genauigkeit zu erkennen. Die identifizierten "Ground-Truth"-Aktionen umfassen Längen von wenigen Sekunden bis fast zu einer Minute, was eine sehr hohe Flexibilität hinsichtlich der erkennbaren Aktionslängen zeigt.

Im zweiten Schritt erfolgt eine qualitative Evaluierung beider Systeme im komplexen OP-Umfeld, um die Benchmark-Ergebnisse in Mehrpersonen-Szenarien zu validieren, aber auch um die Grenzen unseres Ansatzes aufzuzeigen. Gleichzeitig evaluieren wir die Verbesserungen im Personentracking durch das "ID-Swap-Erkennungsmodul". In Szenen mit gelegentlichen Überlappungen erzielt unser Framework ähnlich gute Ergebnisse wie in beiden Benchmarks. Bei Videos mit längeren oder permanenten Personenüberlappungen - wie etwa bei der Zusammenarbeit zweier Personen oder bei Arbeiten an einem dicht befüllten OP-Tisch - ist unser Framework jedoch nicht in der Lage, die Aktionsgrenzen aufgrund dieser Überlappungen zuverlässig zu erkennen. Trotz dieser Einschränkungen ist das "ID-Swap Detection Module" in der Lage, ca. 50% der falschen ID-Swaps des State-of-the-Art "Person Tracker" zu korrigieren und damit die Genauigkeit in diesem anspruchsvollen Umfeld deutlich zu verbessern.

Contents

List of Figures

List of Tables

1 Introduction

In an effort to develop an intelligent operating room (OR), it is essential to obtain a full understanding of the actions that occur within it. Current state-of-the-art action detection algorithms address this challenge by attempting to localize and classify all actions within a video. These algorithms can serve as the foundation for an intra-operative decision-support system, that could assist the surgeon by identifying surgical actions, recognize deviations from the standard procedure, suggest future steps, and produce concise summaries for documentation purposes. However, the primary issue such algorithms face is the accurate temporal localization of actions, especially in complex and crowded environments. To address this, in this thesis we will introduce an unsupervised action boundary localization algorithm capable of handling the complexities found in an OR environment.

1.1 Challenge

The OR presents a complex environment with numerous individuals dressed in similar attire, performing concurrent activities, and experiencing frequent occlusions and overlaps, as shown in Figure 1.1. In addition to that, surgical data is highly sensitive, limiting the availability of publicly accessible datasets that contain videos of surgical procedures.

Figure 1.1: **OR Environment:** Illustration of the complexity within a OR, featuring multiple persons dressed in a similar attire and engaged in various actions.

Current state-of-the-art temporal action localization frameworks are all fully supervised, making them unsuitable for environments without labeled datasets. Although recently, researchers have started to develop unsupervised methods for action boundary detection and summary generation, most of these approaches rely on globally assessing frame

similarity to detect action changes. However, this method proves to be very ineffective in multi-human environments due to the high amount of concurrent actions occurring within a single frame. As a result, a significant gap remains in unsupervised action boundary detection, particularly in complex multi-person scenes.

1.2 Contribution

In this thesis, we will introduce the **Un**supervised **S**patio-**T**emporal **A**ction **B**oundary **L**ocalization (**UnSTABL**) framework, which identifies person-specific action boundaries in an unsupervised manner, making it adaptable to various environments. The framework extracts action information of each individual separately, successfully dealing with multi-person environments, and groups these person-specific feature vectors by similarity to determine significant action changes. These timestamps of key-action transitions can be used to produce concise video summaries, as shown in Figure 1.2, or to provide temporal action proposal segments for a subsequent action recognition stage.

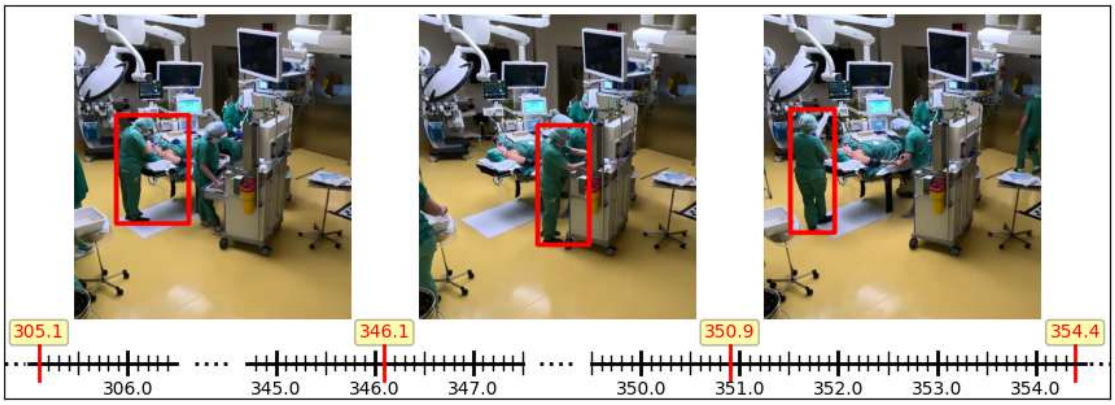

Figure 1.2: **Video Summary Generation in OR:** By identifying person-specific action boundaries, the UnSTABL framework is able to summarize 50 seconds of video content with three key images that capture: (1) patient preparation, (2) retrieving new utensils from the workstation, and (3) walk back to patient.

In a second step, we will introduce the "collision-robust" UnSTABL framework to address the complexities found in an OR. This extension aims to mitigate the problem of incorrectly detected collision-induced boundaries. Additionally, by leveraging action information, the framework is able to improve the performance of the state-of-the-art person tracker, successfully minimizing ID swaps in this challenging environment.

1.3 Thesis outline

Chapter 2 provides the necessary background for this thesis, introducing the underlying concepts of person tracking, action recognition, and action localization tasks.

In Chapter 3, we provide a thorough overview of the current research in action recognition, temporal action detection, and video summary generation while identifying gaps and limitations.

Chapter 4 introduces the UnSTABL framework, providing a detailed overview of the methodologies and key components for a person-specific unsupervised action boundary detection.

Chapter 5 presents further concepts, introducing the "collision-robust" UnSTABL framework, designed to better handle person overlaps and incorrect tracking ID switches in the complex OR setting.

In Chapter 6 we evaluate the framework on a series of datasets, first to establish a performance baseline and then to validate these results in the real-world OR environment, demonstrating promising results and limitations.

The final Chapter 7 completes the thesis with a summary of the evaluation results and two propositions for future research.

2 Background

Understanding and interpreting video data is a crucial task in Computer Vision. To better grasp the challenges and principles underlying action boundary detection and video summary generation, this chapter provides a comprehensive overview of the fundamental concepts and methodologies relevant to person detection and tracking, action recognition, and action localization tasks.

2.1 Person Detection

Person detection, or more generally, object detection, is a fundamental task in computer vision and image processing. Unlike image classification, which assigns a single label to an entire image, object detection involves two steps:

- **Object Localization:** This step involves determining the exact location of each object within an image, typically represented by bounding boxes (visible in Fig.2.1).
- **Object Classification:** Once the objects are localized, each detected object is classified into one of the predefined categories - in this case, the class "Person".

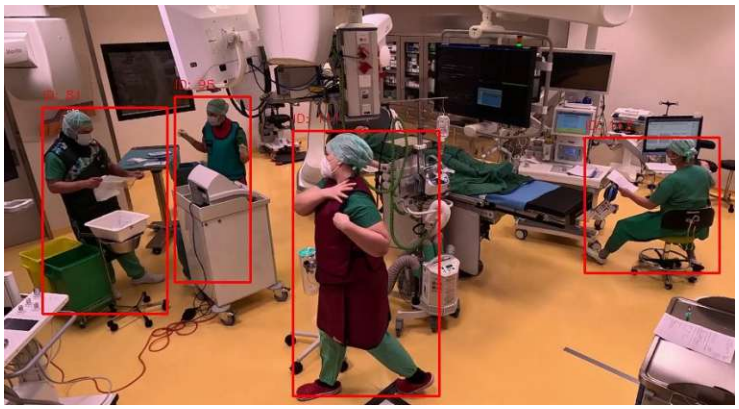

Figure 2.1: **Person Detection and Tracking:** The detection algorithm identifies and locates all persons in each video-frame (red bounding boxes), while the tracking algorithm assigns a unique ID to each person throughout the video sequence.

Historically, object detection relied on matching handcrafted features like SIFT [1] and HOG [2] to reference models. However, with recent advancements in deep learning, these methods have largely become obsolete. Deep learning has introduced two main approaches:

- **Two-Stage Approach:** Models such as R-CNN [3], Fast R-CNN [4], and Faster R-CNN [5] first generate region proposals and then classify them. Although accurate, this method can be computationally intensive.

- **One-Stage Approach:** YOLO (You Only Look Once) [6] simplifies the process by predicting the objects location and class in a single step, making it faster and more suitable for real-time applications.

As a result, deep learning has become the leading approach in object detection, surpassing traditional methods in both performance and efficiency.

2.2 Person Tracking

Following the outputs generated by the Person Detector, the task of the Person Tracker is to identify and track individuals across a series of video frames, even in the presence of occlusions and complex (non-linear) motions. Multi-Object Tracking (MOT) extends this concept by simultaneously tracking multiple individuals and objects within the same scene, as illustrated in Fig.2.1. Object tracking employs two fundamental concepts:

- **Appearance-Based Tracking:** Appearance-based tracking relies on extracting and utilizing visual features from video frames to identify and track individuals. These features might include handcrafted features like color histograms, textures, key points, or deep features extracted using Convolutional Neural Networks (CNNs).
  - **Advantages:** The visual cues enable the tracking of objects or persons in crowded scenes with similar motion patterns and allow re-identification (Re-ID) across different camera perspectives or after periods of disappearance.
  - **Limitations:** Appearance-based methods are sensitive to occlusions and objects with similar appearance, as the visual features become less reliable and can be computationally demanding.
- **Motion-Based Tracking:** Motion-based tracking primarily predicts an object's position based on its previous movements. This is typically achieved through the utilization of a recursive algorithm, such as the Kalman Filter, to estimate the person's states in an optimal manner. The Kalman Filter operates in a two-step process: First, it predicts the current state based on its previous states. Then, it updates the predicted state with the current observations.
  - **Advantages:** By predicting future positions, motion-based tracking can maintain a track even when the object is temporarily occluded.
  - **Limitations:** If the motion model is inaccurate or for complex, erratic motions, the tracker may drift, leading to incorrect predictions. Additionally, it will struggle to distinguish between multiple objects with similar motion patterns.

Modern MOT systems combine appearance-based and motion-based techniques to achieve robust and accurate tracking, even in challenging scenarios. These techniques are integrated through an optimization algorithm like the Hungarian Algorithm to find the optimal assignment of detected objects to existing tracks.

2.3 Action Recognition

Action recognition is the task of automatically identifying and classifying human actions within a video sequence. In specific contexts, such as the 2017 ActivityNet Challenge [7],

the task is defined as the classification of short, trimmed videos containing only one single action class (e.g., walking, jumping). In order to provide a better understanding of the topic, we will discuss the concepts, network models, and architectures currently employed in state-of-the-art action recognition systems.

Spatiotemporal Features

In the context of action recognition, as well as motion detection and video analysis, networks or algorithms typically process spatiotemporal features in order to capture both spatial and temporal information within data. As the name says, they are a combination of two components:

- **Spatial Features:** Spatial features capture the appearance information (e.g. edges, textures, colors, or shapes) within individual frames. They are typically extracted using 2D CNNs.
- **Temporal Features:** Temporal features capture the changes or dynamics over time between successive frames in a video. They are essential for understanding motion and tracking objects. One example of a temporal feature is the optical flow, which measures the motion (direction and magnitude) between two consecutive frames.

Neural Network Models

In recent years, with the significant advancements in deep learning, the focus has shifted towards learning features directly from raw video data rather than manually crafting them from video frames. Currently, three main types of neural network-based models are in use:

- **CNN based models:** Convolutional Neural Networks (CNN) [8] are primarily used in image processing due to their strong ability to capture spatial patterns while being computationally very efficient. A considerable number of pre-trained CNN models are available, which can be fine-tuned for action recognition tasks. However, they cannot capture temporal dynamics across frames, which is crucial for action recognition.
- **RNN based models:** Recurrent Neural Networks (RNNs) are designed to process sequential data. They are frequently paired with CNNs to create so-called hybrid models that are able to capture temporal dependencies between spatial features extracted from consecutive frames. LSTMs [9], a special version of RNNs, excel at retaining important information over long sequences; however, both demand significant memory and computational resources.
- **Transformer based models:** Transformer models, originally developed for natural language processing, have frequently been adapted for action recognition. Their self-attention mechanism enables them to capture long-range dependencies and complex relationships within video data. In contrast to RNNs, which process data sequentially, Transformers simultaneously attend to all frames, which provides a global understanding of the action. However, they also come with challenges like computational complexity and high data requirements.

Action Recognition Architectures

To process both spatial and temporal information, different network architectures have been developed to optimize the extraction and integration of spatiotemporal features. Each of these architectures offers unique advantages for human action recognition:

- **Two-Stream Networks** are designed with two separate, parallel streams: a spatial stream to process appearance-based features and a temporal stream that captures motion dynamics across frames (visible in Fig.2.2a). However, to capture this motion information, the temporal stream relies on optical flow images, which are computationally expensive to calculate. Furthermore, while the separation allows for the independent design and optimization of each stream, the late fusion can limit the ability to fully exploit the interaction between spatial and temporal features.
- **3D-CNN (C3D) Networks** extend traditional 2D convolutions by adding a temporal dimension, allowing them to process spatiotemporal data like frame sequences directly. While they offer a much simpler pipeline than Two-Stream and SlowFast, they may struggle with longer video sequences, as convolutional neural networks are limited in their ability to capture long-term dependencies. Furthermore, 3D convolutions are computationally expensive.
- **Slow-Fast Networks** process video data at two different frame rates: a "slow" pathway for capturing spatial semantics and a "fast" pathway for capturing motion at high temporal resolution (visible in Fig.2.2b). Lateral connections between the two pathways allow the network to learn the interaction of spatial and temporal information, boosting its accuracy even further. Additionally, compared to the Two-Stream Network, it is able to capture motion information without the computation of optical flow images. However, due to their complexity, they typically require large amounts of data to train effectively and avoid overfitting.

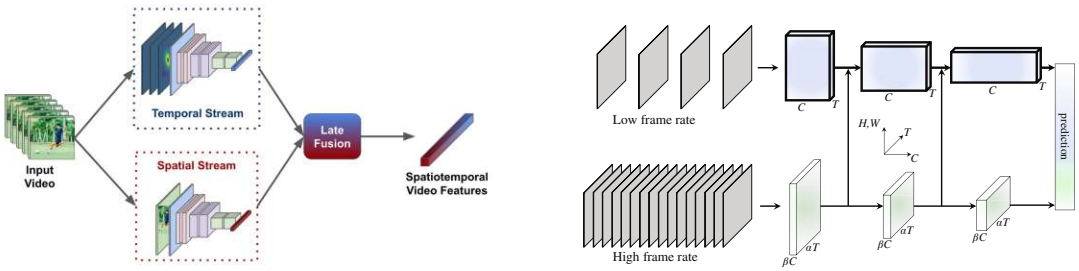

Figure 2.2: **Network Architectures:** (a) The Two-Stream Network, which processes spatial and temporal information through two separate streams [10], and (b) The SlowFast Network, featuring a slow and a fast pathway, fused at specific stages in the network [11].

2.4 Temporal Action Detection

Understanding human actions in long, untrimmed videos is a complex task that requires precise localization and classification of multiple, potentially overlapping actions with varying lengths while also dealing with background noise.

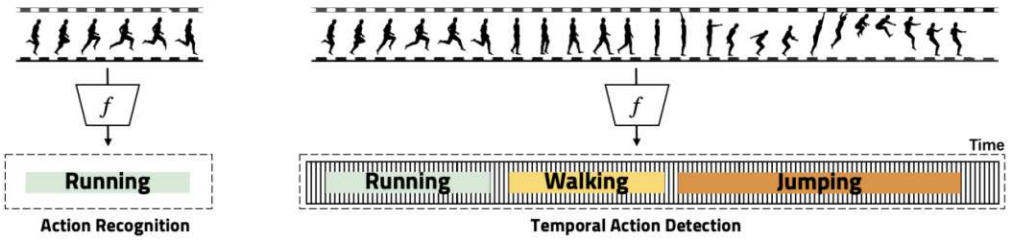

Figure 2.3: **Temporal Action Detection:** (a) Action Recognition deals with classifying trimmed videos while (b) Temporal Action Detection aims to localize action instances in time and classifying them. [12]

Unlike Action Recognition, which focuses on classifying a single action within pre-segmented video clips (Fig.2.3a), Temporal Action Detection (TAD) involves identifying the start and end times of all action instances within the video and then classifying them (Fig.2.3b), similar to the difference between Image Classification and Object Detection. Most of the current TAD algorithms can be split into four general frameworks, as can be seen in Fig. 2.4. In this thesis, we will only focus on the second framework that consists

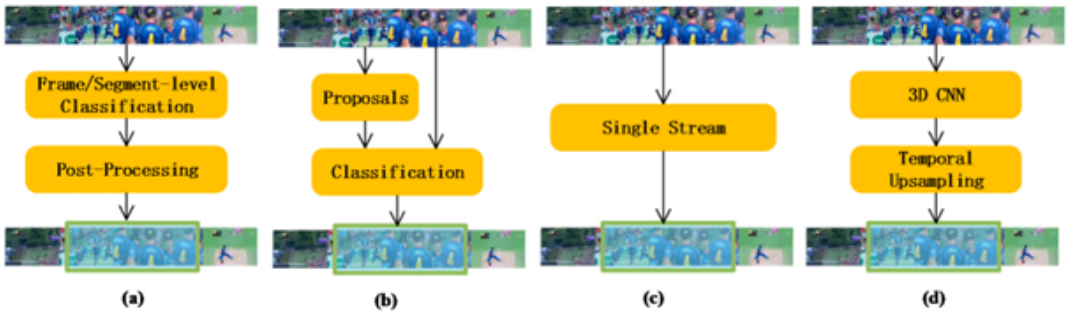

Figure 2.4: **Frameworks for Temporal Action Detection:** (a) classification then post-processing, (b) proposal then classification, (c) single stream, and (d) temporal up-sampling. [13]

of two stages: (1) A temporal action proposal stage, which produces a set of temporal segments that most likely contain a single action instance (which we will implement), and (2) an action classification stage, which determines the specific category of all proposed temporal segments.

2.4.1 Action Boundary Detection

Action Boundary Detection describes the task of identifying the start and end points of all actions within an untrimmed video. These candidate action boundaries can be used as temporal action proposals in a "proposal-then-classification" framework for Temporal

Action Detection (Fig.2.4b). Additionally, these boundaries enable the segmentation of videos into trimmed sequences, which can be used for tasks such as action recognition and video summary generation.

Unsupervised Action Boundary Detection

In unsupervised learning, the model has no prior knowledge of the underlying data, as it was not provided during the training process. In the context of action boundary detection, the model must autonomously discover patterns in video data that correspond to action boundaries without relying on labeled training data. These methods are essential in real-world scenarios where labeled data is limited, such as in operating rooms. Commonly used techniques in unsupervised learning include:

- **Clustering:** Clustering techniques, such as K-Means and Hierarchical Clustering, group data points into clusters based on their similarity. Temporal boundaries can then be identified at timestamps where the cluster changes, indicating a transition between actions.
- **Dimensionality Reduction:** Dimensionality reduction methods like Principal Component Analysis (PCA) [14] and t-SNE reduce the number of features in a dataset while retaining the essential information.
- **Feature Extraction:** Feature extraction involves transforming raw data into feature vectors that can be more easily analyzed. These features can either be handcrafted or extracted using a pre-trained Network by stripping the last classification layer. Networks identify high-level features that are not easily observable by manual methods. By combining feature extraction with a dimensionality reduction algorithm, it is possible to eliminate irrelevant or redundant features. This reduces the computational complexity while increasing the information value.
- **Gaussian Mixture Models:** Gaussian Mixture Models (GMMs) are probabilistic models that assume the data is generated from a mixture of Gaussian distributions. They are commonly used for clustering, density estimation, and modeling complex data distributions where simple clustering techniques may not be sufficient.

There are numerous other techniques and models, such as Autoencoders and Generative Adversarial Networks (GANs) [15], that can be utilized for unsupervised learning. However, this thesis will focus primarily on the techniques mentioned above.

2.4.2 Temporal Action Localization

As defined in the 2017 ActivityNet Challenge [7], temporal action localization methods are designed to identify specific action instances within untrimmed videos by achieving two key objectives:

1. When does the action occur (i.e., identifying the start and end times of the action).
2. What action class does each proposal belong to (e.g. Walking, Jumping).

Essentially, Temporal Action Localization (TAL) covers the same tasks as Temporal Action Detection. However, much of the current research has shifted its focus toward the temporal action proposal generation part since action classification is typically well-addressed in fully supervised settings. The challenge of the proposal generation is to accurately identify the exact start and end points of each action within the video, and to distinguish true actions from background content where no relevant activity is present, as shown in Figure 2.5. Additionally, some researchers are pursuing weakly supervised or even unsupervised

Figure 2.5: **Temporal Action Localization** localizes and classifies actions while filtering out background content with no relevant action. [16]

approaches to Temporal Action Localization, aiming to reduce the need for extensive labeled data.

2.4.3 Spatio-temporal Action Detection

In addition to localizing the start and end points of all actions inside a video and classifying them, Spatio-temporal Action Detection algorithms additionally determine where the action occurs in each frame, usually represented by a bounding box, visible in Figure 2.6. This is particularly useful when numerous actions occur simultaneously in one video frame.

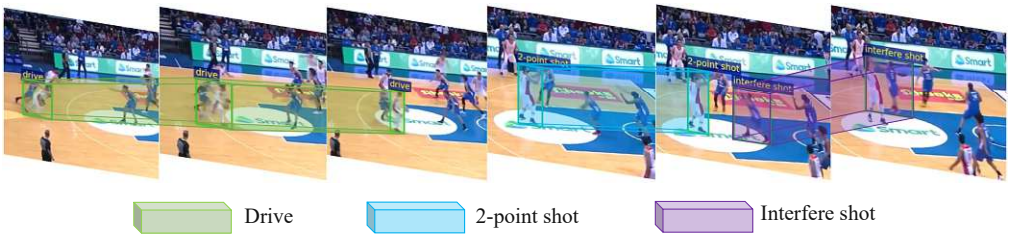

Figure 2.6: **Spatio-temporal Action Detection** classifies and localizes actions in space (bounding boxes) and time. [17]

2.5 Video Summary Generation

Video summarization algorithms aim to produce a concise representation of a video by selecting its most informative parts. What constitutes the "most informative parts" is often subjective, but it typically involves detecting and representing key action instances within the video. Summaries can be video-based, creating a shortened clip, or image-based, using key-frame extraction to create a timeline of images as shown in Figure 2.7. In each detected key-action segment, key-frames are either selected at specific points (such as the

beginning, middle, and end) or identified through frame comparison to determine the most distinct or representative images of the segment's content. Automatic summary generation

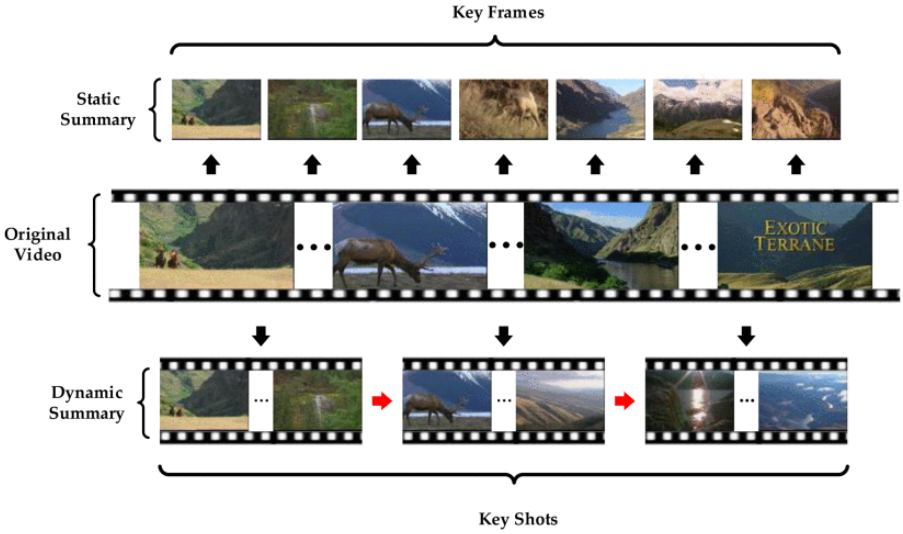

Figure 2.7: **Video Summary Generation approaches:** (a) Key frame extraction creates summaries by displaying selected key frames, while (b) Video skimming creates a compressed video with key shots from the most informative parts of the original content. [18]

techniques range from supervised methods, that learn from numerous human-generated summaries, to unsupervised methods, that use clustering or GANs [15] to identify and select the most representative segments of the video.

3 Related Work

As the quantity of online video data has grown significantly, the need for automatic video understanding has driven rapid advancements in the field. This chapter critically examines prior research in action recognition, temporal action detection, and video summary generation while identifying gaps and limitations in more specialized areas such as person tracking, surgical phase recognition, and action boundary detection.

3.1 Person Detection and Tracking

Many approaches have been developed for person detection and tracking, each offering trade-offs between real-time performance and accuracy. In complex, crowded environments like the operation room, reliable person detection and tracking, even under frequent occlusions and complex movements, is crucial. Consequently, developers opted for a modular design, enabling the independent optimization of the detector and tracker.

3.1.1 YOLO series

As a one-stage approach, the YOLO (You Only Look Once) framework [6] has stood out for its remarkable balance of speed and accuracy, making it a popular choice for real-time object detection tasks. Unlike multi-stage methods discussed in Section 2.1 YOLO divides

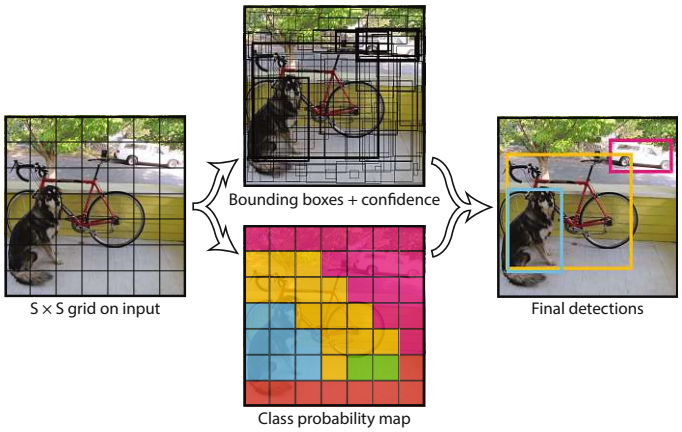

Figure 3.1: **YOLO Model:** YOLO divides the image into a $S \times S$ grid, with each cell predicting object locations and classes in one step. [6]

the input image into a grid of $S \times S$ cells. Each cell is responsible for predicting a fixed number of bounding boxes B, along with a confidence score and class probabilities for C possible classes. This approach allows YOLO to simultaneously predict both the location and the class of objects in a single step, as illustrated in Figure 3.1.

Since its introduction in 2016, various versions of YOLO have been developed to address the challenges of detecting objects in diverse and complex environments. YOLOv3 [19] marked an important milestone, incorporating multi-scale detection, which significantly improved the prediction of smaller objects. It remains one of the most widely used detectors in the industry due to its robustness and reliability.

Building on the foundation of YOLOv3, YOLOX [20] introduced significant enhancements, including an anchor-free detection and a decoupled head design, contributing to superior detection precision and computational efficiency. Consequently, YOLOX has become a preferred backbone in many state-of-the-art tracking algorithms, offering a refined balance between high accuracy and real-time performance. Additionally, newer versions like YOLOv7 [21] and YOLOv8 [22] have further enhanced the framework, making them strong contenders in the field of real-time object detection.

To effectively implement this one-stage approach, large annotated datasets like Common Objects in Context (COCO) [23] are necessary to train the models for accurate object detection across diverse real-world scenarios.

3.1.2 Deep OC-SORT

Deep OC-SORT [24] and ByteTrack [25] are the leading state-of-the-art object trackers, each offering specific advantages for different scenarios. ByteTrack is mainly used for robust real-time tracking. At the same time, Deep OC-SORT is highly effective in handling occlusions and maintaining identity through complex interactions, but at a higher computational cost. Consequently, in the context of the complex surgical environment, this thesis will focus on Deep OC-SORT due to its robustness in challenging conditions, however, at the cost of real-time performance.

The Kalman Filter-based tracking algorithm OC-SORT [26] forms the foundation of the DeepOC-SORT framework. OC-SORT improves SORT's [27] tracking robustness in non-linear motion scenarios and mitigates the impact of object occlusion or disappearances. Building on this framework, DeepOC-SORT integrates appearance-based multi-object

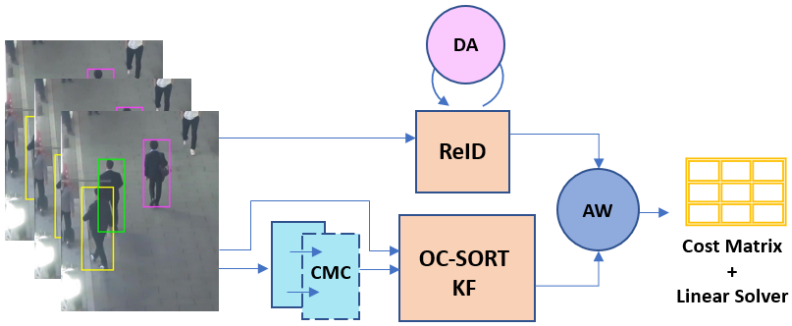

Figure 3.2: **Deep-OC-SORT Model:** Deep-OC-SORT integrates motion-based tracking (OC-SORT's Kalman Filter) with appearance information (Dynamic Appearance Module) and combines them using Adaptive Weighting (AW) for improved Multi-Object Tracking. [24]

tracking (MOT) into the motion-based approach of OC-SORT, similar to how DeepSORT [28] enhances the original SORT algorithm with appearance features. The dynamic appearance (DA) module of Deep-OC-SORT extracts and weights appearance features

based on the object detectors' confidence. The adaptive weighting (AW) module ensures that only high-quality information is incorporated into the tracking predictions by adjusting its weights according to the discriminative power of the features. An overview of the model is given in Figure 3.3.

However, despite these advancements, occlusion handling remains one of the most critical challenges in MOT, as it is a primary cause of ID switches and trajectory fragmentation.

Datasets for Person Tracking

MOT17 [29] and MOT20 [30] are among the most widely used datasets for multi-object tracking. MOT17 focuses on moderately crowded scenes in diverse environments, while MOT20 presents more challenging scenarios with extremely dense crowds, testing algorithms' ability to handle high occlusion rates.

The DanceTrack [31] dataset is a large-scale multi-human tracking dataset, primarily consisting of group dancing videos that feature scenarios with individuals who have similar appearances, complex motion patterns, and significant occlusions. These characteristics

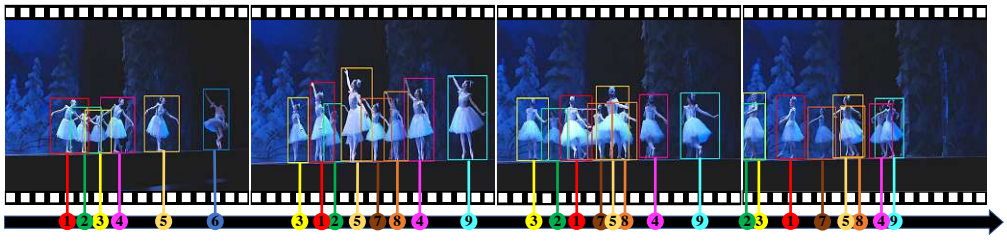

Figure 3.3: **DanceTrack Dataset:** The multi-human tracking dataset consists primarily of group dancing videos, featuring individuals with: (1) uniform appearance and (2) diverse motion, including position switches and occlusions. [31]

make it particularly relevant for surgical environments. Similarly to the dance videos, the OR contains individuals dressed in similar attire, engaged in complex movements, and experiencing occlusions. Notably, DeepOC-SORT outperforms all state-of-the-art algorithms on the DanceTrack dataset, demonstrating its effectiveness in handling such challenging conditions.

3.2 Action Recognition

In 2016, Feichtenhofer et al. made a significant contribution to the field of video action recognition with the introduction of the Convolutional Two-Stream Network Fusion [32], which builds upon the Two-Stream ConvNets framework [33] introduced in 2014. This method remains a leading 2D CNN-based approach for action recognition. It employs a two-stream CNN architecture, as explained in Section 2.3 - one stream captures spatial information from RGB frames, while the other captures temporal dynamics via optical flow images (illustrated in Figure 2.2a). By combining these streams, the model effectively captures both appearance and motion cues, resulting in robust action recognition.

Following advancements in 3D CNNs, such as the C3D [34] and I3D networks [35], as and the introduction of 3D ResNet architectures [36], Feichtenhofer et al. introduced the SlowFast Network [11] in 2019. In this work, he addressed the limitations of optical flow

images used in his previous works, namely their high computational costs and lack of end-to-end processing. The SlowFast Network employs two temporally strided 3D ResNet streams—one operating at a slow frame rate to capture detailed spatial information and the other at a fast frame rate to capture rapid temporal dynamics. The information from the two pathways is repeatedly fused by lateral connections, enabling the network to learn the interactions between slow and fast features. An overview of the model is provided in Figure 2.2b. This slow-fast approach excels at processing both quick actions, such as gestures or sudden movements, and slower, more deliberate actions, such as walking or stretching. Consequently, it is currently one of the best non-transformer-based action recognition models, as demonstrated on the Kinetics-400 benchmark dataset [37].

3.2.1 SlowFast Action Detection with the AVA Dataset

In addition to SlowFast's strengths in action recognition, Feichtenhofer highlights that the network can also be used as a Spatio-Temporal Action Detection algorithm, achieving state-of-the-art results. The model first uses a person detector to identify and generate

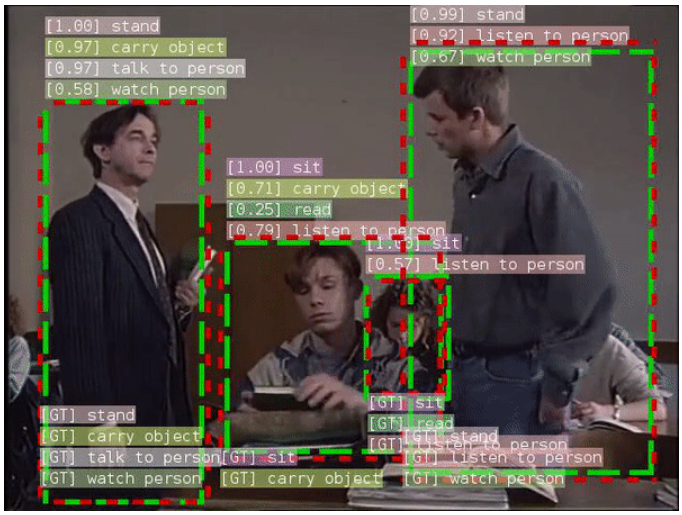

Figure 3.4: **SlowFast Spatio-Temporal Action Detection:** Utilizing a Person Detector and the AVA Dataset, the Slowfast Network is able to determine the spatio-temporal actions of each person in the video. [11]

region-of-interest (RoI) proposals for each person detected in the video frames. These 2D RoI proposals are then extended to 3D RoI's across the temporal axis and fed into the modified SlowFast network for multi-label action prediction. This allows the network to accurately detect and classify actions within the identified spatio-temporal video regions as shown in Figure 3.4. This approach positions the SlowFast network as an ideal feature extractor for our unsupervised spatio-temporal action boundary detection algorithm.

The SlowFast backbone is initially pre-trained on the Kinetics-400 dataset [38] and then fine-tuned on the AVA dataset [39], which provides spatio-temporal annotations of human actions in video segments.

3.3 Temporal Action Detection

Over the recent years, significant progress has been made in various categories of temporal action detection; in the following subsections, we provide a brief overview of state-of-the-art techniques across different areas.

3.3.1 Operating Room Activity Recognition

The objective of Operation Room (OR) Activity Recognition is to identify and classify activities within the surgical environment to improve workflow efficiency and documentation. In recent years, the majority of approaches have been developed for robot-assisted surgeries and invasive OR videos (Fig.3.5a). At the same time, only a limited number of methods have been applied to surveillance-like OR videos that capture the full workflow and interactions occurring throughout the entire room (Fig.3.5b).

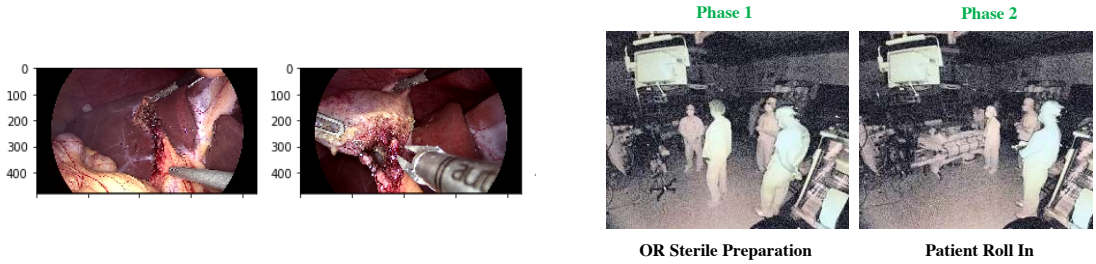

Figure 3.5: **Surgical Activity Recognition Datasets:** (a) The invasive Cholec80 dataset [40] and (b) the surveillance-like OR-AR dataset [41]

In 2017, Twinanda proposed two vision-based approaches for surgical activity recognition [42] utilizing (1) laparoscopic (invasive) and (2) RGBD (surveillance-like) videos from the m2cai16-workflow dataset. He demonstrated state-of-the-art results by evaluating the laparoscopic approach on the Cholec80 dataset, which features 80 invasive videos of cholecystectomy surgeries.

In 2020, Sharghi et al. introduced a method for automatic recognition of surgical activity in robot-assisted surgery [41], employing an inflated 3D ConvNet combined with temporal Gaussian mixture layers and a long short-term memory (LSTM) unit. The model was trained on a large-scale dataset of 400 annotated OR videos, achieving state-of-the-art results. However, this OR-AR dataset is not publicly available.

The lack of publicly available datasets featuring surveillance-like OR videos remains a significant challenge, limiting the development of fully supervised action detection models that can effectively capture the complexity of the entire surgical environment.

3.3.2 Unsupervised Action Boundary Detection

Over recent years, new unsupervised algorithms for action boundary detection (ABD) have emerged, driven by the need to analyze high amounts of untrimmed video data without relying on labor-intensive annotations.

In 2022, Du et al. proposed a novel unsupervised ABD method [43] that detects boundaries by analyzing frame-to-frame similarities, visible in Figure 3.6a. They identify

precise boundary points by applying non-maximum suppression to the similarity curve. . By refining these points through a clustering process, they achieve good results on datasets like Breakfast [44] without the need of training.

Two years later, Li et al. introduced the unsupervised, Object-centric Temporal Action Segmentation (OTAS) framework [45], which enhances boundary detection by integrating global and local features. OTAS consists of a global perception and object attention module that learns global visual features, local interaction features, and object relational features in a self-supervised manner, as illustrated in Figure 3.6b. The boundary selection module fuses these features to detect action boundaries. OTAS offers a more refined and context-aware approach to boundary detection, delivering state-of-the-art results on the Breakfast dataset [44].

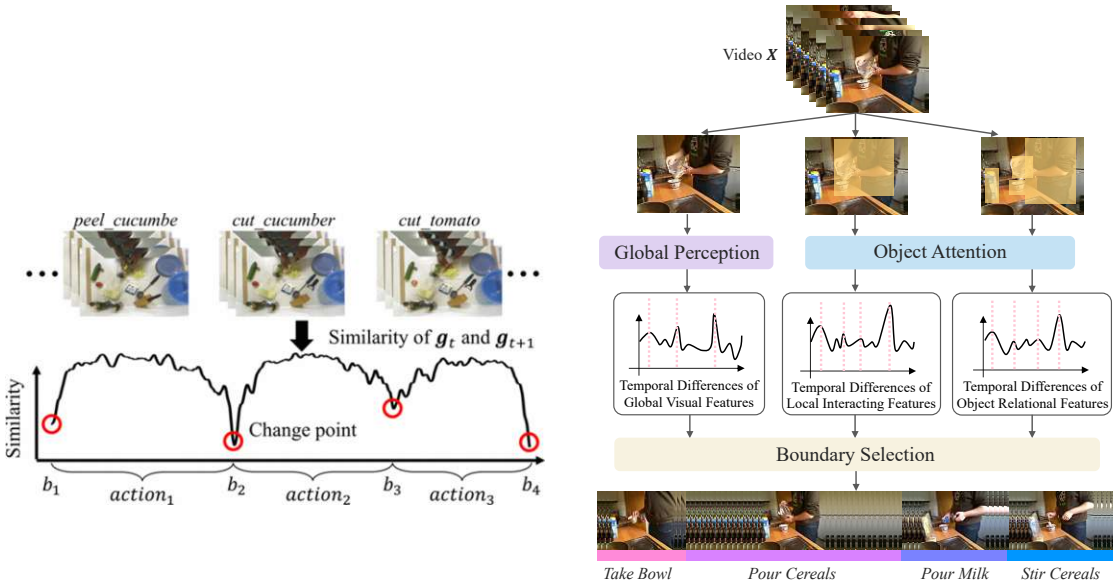

Figure 3.6: The (a) **Unsupervised Action Boundary Detection** algorithm proposed by Du et al. [43] leverages frame-to-frame similarities, while the (b) **OTAS** framework [45] combines global visual-, local interaction-, and object relational features for action boundary detection.

Despite these advances, the focus has primarily been on single-person environments such as in Breakfast [44], which do not fully capture the complexity found in real-world settings. Therefore, addressing challenges such as high levels of occlusion and interactions among multiple individuals with similar appearances will require further research.

Breakfast

As previously mentioned, the Breakfast dataset [44] is one of the most widely used benchmark datasets for action boundary detection and action segmentation. It features videos of breakfast preparation tasks captured in various kitchen environments and recorded from multiple camera viewpoints. The videos consist of sequential, non-overlapping actions with varying durations, such as preparing coffee, frying eggs, and pouring juice, performed by one person at a time. The structured and ordered nature of activities makes the dataset

particularly useful for evaluating systems that focus on detecting changes between distinct actions (e.g. for action boundary detection). However, it lacks the complexities found in more natural, multi-person environments, where frequent occlusions and overlaps pose greater challenges.

3.3.3 Temporal Action Proposal Generation

In her 2017 work, Lin identified the quality of action proposals as the primary bottleneck in temporal action localization. To address this issue, she proposed the Temporal Convolution Based Action Proposal network [46]. Building on this work, she introduced the Boundary Sensitive Network (BSN) [47] in 2018, establishing the foundation for future Temporal Action Proposal Generation algorithms.

BSN uses a two-stream network as its backbone for feature extraction, leveraging spatial and temporal information to learn the start, end, and actionness probabilities of video snippets, as illustrated in Figure 3.7. Subsequently, a three-step process is applied in a local-to-global fashion: first, the Temporal Evaluation Module functions as an action boundary detection algorithm, identifying potential action boundaries. These boundaries are subsequently refined and combined in the Proposal Generation Module using the actionness score to create temporal action proposals. Finally, the Proposal Evaluation Module globally assesses the confidence that each proposal contains an action. This approach significantly advanced the state of the art by improving the precision and recall of action proposals.

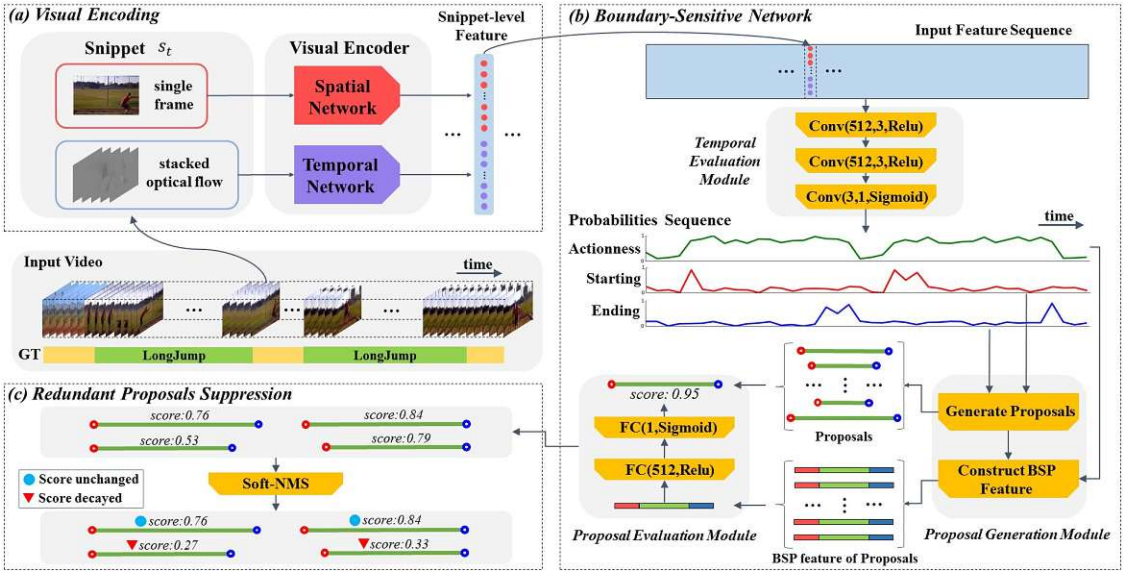

Figure 3.7: **Boundary Sensitive Network** BSN leverages a two-stream network for feature extraction, followed by a three-step process: (1) detecting action boundaries through the probability sequence, (2) refining and combining the boundaries into proposals, and (3) globally assessing the confidence of each proposal. [47]

The Boundary-Matching Network (BMN) [48], introduced in 2019, builds upon BSN

builds upon BSN while refining the proposal generation process. Through a Boundary-Matching mechanism that evaluates pairs of temporal boundaries more effectively, leading to higher-quality proposals with improved recall and precision. In 2021, BSN++ [49] further advanced these foundations by introducing a complementary boundary regressor, which enhances boundary precision through a U-shaped architecture and bi-directional boundary matching mechanism, and a proposal relation block that better models relationships between proposals.

Nevertheless, despite their success in delivering precise temporal action proposals and boundaries, all these approaches are fully supervised. This makes them unsuitable for environments like the operating room, where annotated data is limited to non-existent.

3.3.4 Toyota Smarthome Untrimmed Dataset

Current state-of-the-art datasets often fail to capture the complexity and spontaneous behaviours required to develope robust action detection systems in real-world scenarios. The Toyota Smarthome Untrimmed (TSU) dataset [50] addresses these shortcomings by providing long and untrimmed "surveillance-like" videos that capture unscripted daily activities within a smart home environment. Since the TSU dataset was developed for temporal action detection, it includes annotations for the temporal locations and classes of each action while also distinguishing between background and foreground activities.

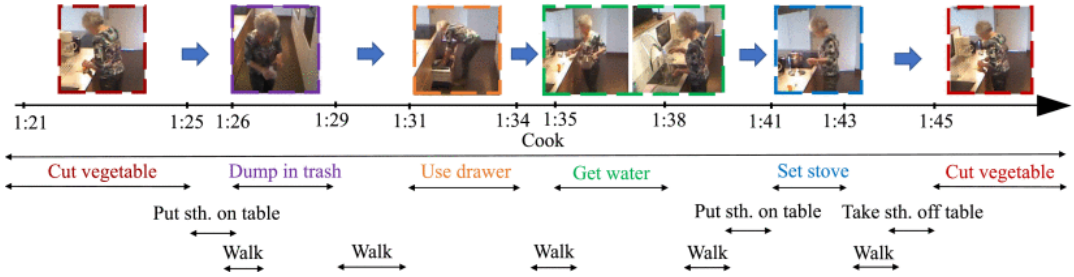

Figure 3.8: **Toyota Smarthome Dataset:** An example of the actions and annotations in the TSU dataset. [50]

The dataset contains 536 videos recorded from different rooms, including the kitchen and living room, using multiple different camera angles. It includes a a total of 51 different actions, ranging from coarse, composite actions such as "Cooking" to fine-grained actions such as "Drinking from a cup" or "Use Drawer". Figure 3.8 provides an illustrative example. The actions vary significantly in duration, from just a few seconds to several minutes, providing an ideal test for the temporal flexibility of our algorithm. The activities are entirely unscripted, with subjects often behaving unexpectedly, which, along with the challenges of high temporal variance, concurrent activities, and complex composite actions, makes the TSU dataset a challenging benchmark for testing action detection algorithms in real-world scenarios.

3.4 Video Summary Generation

Unsupervised video summarization techniques represent a powerful means of automatically generating concise video summaries without the need for labeled data. These methods

typically utilize some form of similarity measure to detect regions of interest where significant (scene) changes occur. By selecting keyframes from these regions, they create concise summaries that capture the key content of the video.

The GVSUM approach [51], proposed by Basavarajaiah in 2020, implements this process. It extracts deep visual features using a pre-trained Convolutional Neural Network (CNN) from video frames and groups them based on similarity using k-means clustering. Keyframes are selected whenever there is a change in cluster labels, capturing significant scene changes and efficiently summarizing videos with minimal computational cost.

In their study on multiview video summarization, Parihar et al. [52] leverage frame similarity across multiple videos captured from individual cameras. This method involves an early redundancy elimination using the BIRCH clustering algorithm, followed by shot boundary detection through similarity measures like Jaccard and Dice. After partitioning the video based on these boundaries, a multi-level K-means clustering algorithm is applied to identify the most representative frames across different camera angles, which are then merged to form the final summary.

However, both methods primarily rely on frame similarity without explicitly considering human actions within the scenes. In contrast, our approach goes a step further by detecting action changes at the individual level, allowing for more precise summaries, particularly in videos featuring multiple people.

4 UnSTABL

Action detection algorithms are able to provide a detailed analysis of longer, untrimmed videos by accurately localizing and classifying all action instances within it. However, the bottleneck of these algorithms is the accurate temporal localization of actions, leading to a significant research focus on identifying action boundaries and generating temporal action proposals. Most existing temporal action proposal generation algorithms rely on fully supervised learning methods, making them unsuitable for environments like the OR, where annotated datasets are sparse to nonexistent. Although recently, Li et al. [45] introduced a self-supervised method for action boundary detection by leveraging local and global features, this approach has been designed for simpler environments containing only one single individual. Consequently, a significant gap remains in effectively localizing actions in complex, real-world settings like the OR, which feature multiple individuals and concurrent actions, especially when annotated datasets are unavailable.

To address these challenges, we propose the **Un**supervised **S**patio-**T**emporal **A**ction **B**oundary **L**ocalization (**UnSTABL**) algorithm. This chapter provides a detailed overview of our proposed UnSTABL framework, while the next chapter presents further concepts to adapt the base module for more complex and crowded environments.

4.1 Model overview

Our UnSTABL framework employs a person-based methodology for action boundary detection, which we will show not only delivers state-of-the-art results but is also capable of accurately detecting individual action boundaries in multi-person, real-world environments.

The underlying concept of this approach is built upon the complexity of crowded scenes, where understanding the entire image at once becomes infeasible due to the high amount of concurrent activities. Similar to R-CNN [3] in object detection, our method focuses on specific Regions of Interest (RoI) rather than attempting to interpret the entire scene at once. By treating each individual person as a RoI, the framework is able to detect and pinpoint where and when person-specific actions change over time. This spatio-temporal approach provides a robust solution for complex multi-person environments by independently detecting each person's action boundaries. Furthermore, its unsupervised nature makes it fast and easy to implement, eliminating the need for extensive training and ensuring high adaptability to various environments. The UnSTABL framework can be split into three different stages as illustrated in Figure 4.1:

- **Person Detection and Tracking Stage:** This Stage is employed to identify the Regions of Interest (RoIs), represented by bounding boxes. The Person Detector locates individuals within each video frame. At the same time, the Person Tracker assigns consistent IDs to the same person across different frames, linking these RoIs over time.

- **Feature Extraction Stage:** This stage utilizes the bounding boxes and video-frames to extract spatio-temporal feature vectors of each Region of Interest. These feature vectors are collected throughout the entire video duration and serve as the input for the subsequent stage, where they will be further processed for boundary detection.
- **Boundary Detection Stage:** In this final stage, the previously collected person-specific feature vectors are analyzed and grouped based on their similarity. The model can then accurately detect person-specific action boundaries by identifying transitions in these action groups over time.

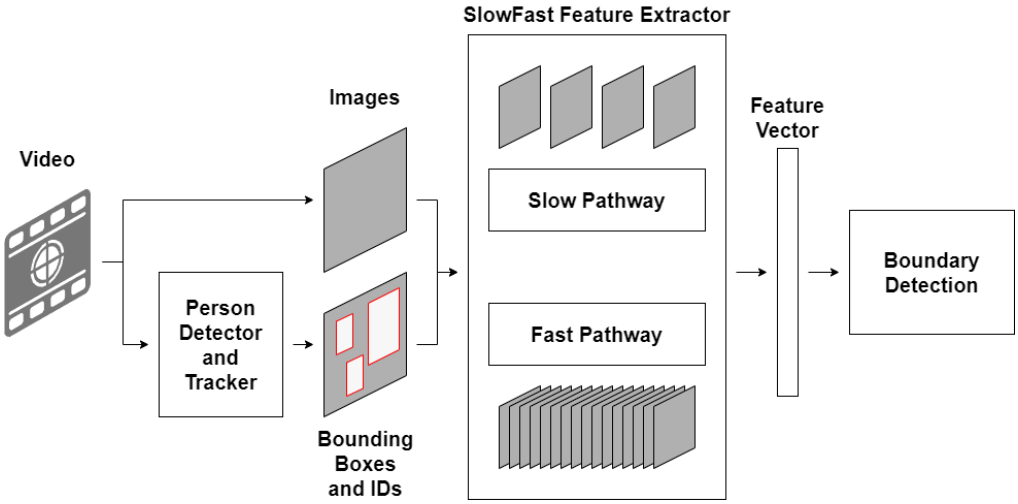

Figure 4.1: **UnSTABL Model:** The framework consists of 3 stages: (1) a Person Detector and Tracker to localize all individuals throughout the video, (2) a Feature Extractor to extract spatio-temporal action information; and (3) a Boundary Detection Module that utilizes this information to identify action boundaries.

This framework enables a precise detection of each person's action boundaries in space and time, while also identifying when individuals enter or leave the video. The above mentioned stages will be discussed in greater detail in the following subsections.

4.2 Person Detector and Tracker Module

To ensure a fast and easy implementation, we utilize a pre-built and pre-trained network for person detection and tracking. As discussed in Section 3.1.2, even though ByteTrack offers real-time performance, Deep OC-SORT excels at tracking individuals in challenging conditions. As shown in Figure 4.2, Deep OC-SORT is able to maintain the identity even through complex interactions and occlusions. When paired with YOLOX as the backbone person detector, which offers an ideal balance between high accuracy and real-time performance, Deep OC-SORT delivers state-of-the-art results, making them the optimal choice for our Person Detector and Tracker Module.

For our specific use case, which involves processing static, "surveillance-like" videos, we can disable the Camera-Motion-Correction (CMC) Module from the original Deep

Figure 4.2: **Deep OC-SORT Occlusion Handling:** The tracker is able to detect persons trough heavy occlusions (Image 3) and maintain the same ID's despite complex interactions and overlapping individuals.

OC-SORT tracker [24]. This modification helps to save computation time, as Deep OC-SORT is already computationally intensive.

Maintaining a consistent person ID throughout the entire video is crucial for our person-based action boundary detection approach. To enhance the tracker's accuracy in our complex environment, we utilize a network model pre-trained on the DanceTrack dataset. While the MOT17 and MOT20 datasets are designed for tracking pedestrians in crowded scenes, the DanceTrack dataset presents an even more challenging scenario. It primarily features dancers dressed in very similar attire (Fig.2.1), with complex movement patterns, heavy occlusions, and frequent crossovers. This dataset helps us to push the detection and tracking even further, since individuals in the operating room (OR) are also dressed similarly and encounter similar challenges of occlusion and crossover, as demonstrated in Figure 4.2. By leveraging this dataset for our tracker model, we can more effectively match feature vectors and action boundaries to the correct person throughout the entire video. Despite all these measures and improvements, the tracker still introduces occasional ID swaps after person overlaps, which we will examine in more detail in the next chapter.

4.3 SlowFast Feature Extractor

Similar to the Boundary Sensitive Network (BSN) [47], we employ a modified Action Recognition Network to extract spatio-temporal information, so-called feature vectors, from video data. However, whereas BSN employs a subsequent network to predict action scores, along with action start and end probabilities in a fully supervised manner, we will implement an unsupervised Boundary Detection Module in the following section.

BSN employs a Two-Stream Network for feature extraction; however, we argue that the SlowFast Network is the superior choice. As previously outlined in Sections 2.3 and 3.2, the SlowFast Network offers several advantages. First, it eliminates the need for computationally expensive optical flow images required by the Two-Stream Network to capture temporal dependencies, decreasing the computation time by almost *half*. Secondly, the SlowFast Network excels at capturing both fast and slow movements by processing the video at two different frame rates. This allows the feature vectors to capture a wider range of temporal dynamics, which is particularly beneficial in an unsupervised

setting, where clustering similar actions of varying durations and temporal characteristics is essential. Thirdly, through lateral connections between the two pathways, the network learns relations between spatial and temporal information, further enhancing its accuracy.

We utilize a modified version of Feichtenhofer's SlowFast Network [11] for Feature Extraction. As discussed in Section 3.2.1, when paired with a Person Detector, the model can be used as a Spatio-Temporal Action Detection algorithm. Inspired by Faster R-CNN [5], the model processes temporally concatenated Regions of Interest, in our case, the detected and tracked persons over time, and determines the spatio-temporal actions of each person over time. By slightly modifying the provided SlowFast Network [53] and omitting the last classification layer, we are able to retrieve a feature vector of size 2304 for each Region of Interest containing action-specific information.

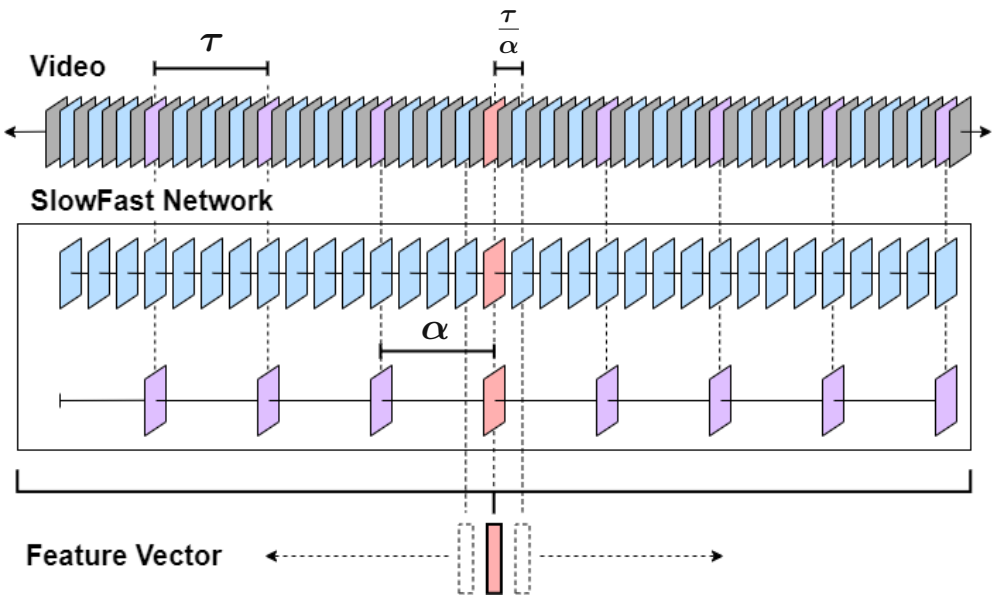

Figure 4.3: **SlowFast Feature Extraction:** The Fast Pathway (blue) processes αT frames at a smaller temporal stride to capture fast movements, while the Slow pathway (violet) processes only T frames at a larger temporal stride to capture slower actions. α denotes the frame rate ratio between both pathways. The feature vector (red) contains information from $T \times \tau$ video frames.

The Slow pathway in the SlowFast network is a temporally strided 3D ResNet, processing frames at a large temporal stride, meaning it analyzes only one out of every τ frames (Fig.4.3 violet pathway). The Fast pathway, on the other hand, works with a smaller temporal stride of $\frac{\tau}{\alpha}$ (Fig.4.3 blue pathway), where α represents the frame rate ratio between the Fast and Slow pathways. Both pathways operate on the same raw video clip of lenght $T \times \tau$, with the Slow pathway sampling T frames, while the Fast pathway is α times denser, processing αT frames.

Following Feichtenhofer's paper, we must choose a specific sampling rate ($T \times \tau$) for our Slow pathway, while balancing accuracy and computational cost. Doubling the number of frames in the Slow pathway (increasing T or decreasing τ) increases the performance at double the computational cost. As shown in the paper [11], the 8×8 model offers an optimal tradeoff, providing nearly the same accuracy as the best model (16×8) at half

the cost. With a maximum frame length of 32 in the Fast pathway, we chose α to be 4, ensuring that $\alpha T = 32$. This configuration means that each feature vector contains information from roughly 2 seconds of video content ($T \times \tau = 64$ frames at 30 fps), therefore is able to capture both faster and slower movements.

Again, we utilize a pre-trained model for a faster and easier implementation. The SlowFast model, with our chosen parameters τ, T, and α, is initially pre-trained on the Kinetics600 dataset and then fine-tuned on the AVA action detection dataset, ensuring robust feature extraction tailored for unsupervised action localization tasks.

4.4 Boundary Detection Module

The Boundary Detection module is the core component of our UnSTABL framework, and it is responsible for identifying person-specific action boundaries based on the information provided by the previous modules. Lin et al. trained a Temporal Evaluation Module in their Boundary Sensitive Network [47] to determine the starting, ending, and actionness probabilities from the feature vectors in a fully supervised setting. However, due to the

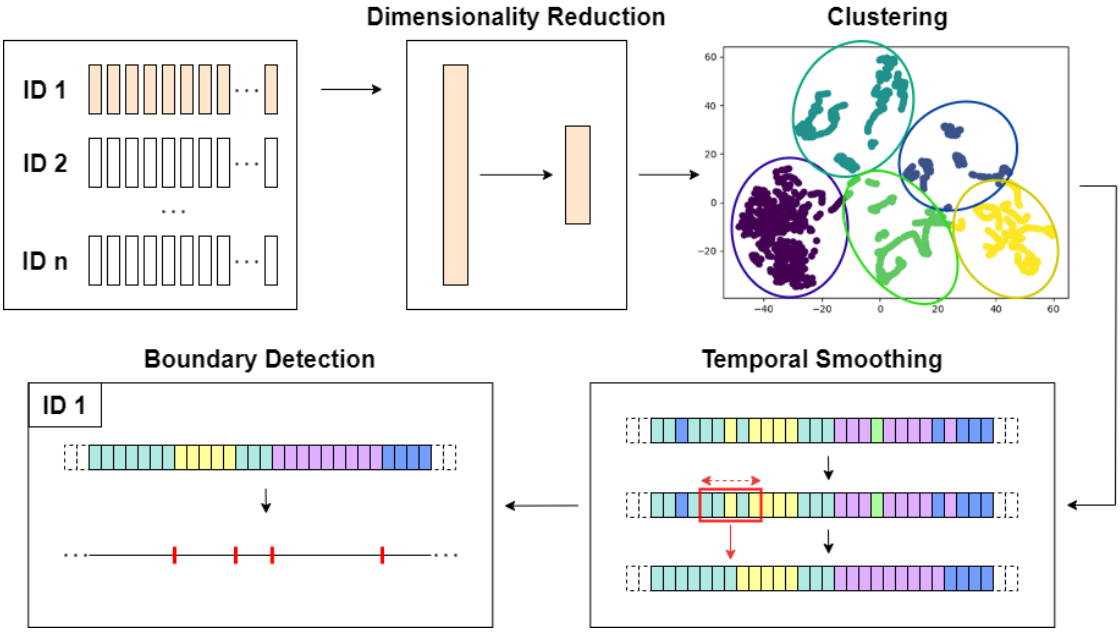

Figure 4.4: **Boundary Detection Module:** Action boundaries of the selected person are identified through four steps: (1) Dimensionality Reduction and (2) Clustering of all extracted person-specific feature vectors; and (3) Temporal Smoothing and (4) Boundary Detection of the chronologically ordered cluster labels.

lack of labeled training data, we developed an unsupervised approach for action boundary detection. Inspired by recent unsupervised learning techniques, our approach is divided into four stages, as illustrated in Figure 4.4:

- **Dimensionality Reduction Stage:** We start by applying Principal Component Analysis (PCA) to reduce the dimensionality of all feature vectors associated

with the selected person. By removing redundant information, PCA enhances the informational value of the feature vectors.

- **Clustering Stage:** Secondly, the reduced feature vectors of each person are grouped based on similarity using Gaussian Mixture Models (GMMs).
- **Temporal Smoothing Stage:** The previously determined cluster labels are arranged in chronological order, and a sliding window majority filtering approach is applied to remove outliers and noisy cluster assignments.
- **Boundary Detection Stage:** In the final stage, action boundaries are identified at timestamps where transitions in the chronologically ordered cluster labels occur.

The following subsections will provide a more detailed explanation of these stages and how they relate to each other.

4.4.1 Dimensionality Reduction and Clustering

The curse of dimensionality is a common problem when working with high-dimensional data, particularly in machine learning and clustering. In high-dimensional spaces, data points become sparse, meaning that they are spread far apart, and the relative distances between all points tend to converge and become similar. This phenomenon is called distance concentration [54]. As a result, it becomes challenging for clustering algorithms to distinguish between different clusters based on distances or densities alone. Furthermore, the higher the dimensionality of the data, the greater the computational resources required to form clusters and find optimal parameters and solutions.

Principle Component Analysis (PCA)

To address the curse of dimensionality, we employ Principal Component Analysis (PCA) [14] to reduce the number of dimensions in the data. PCA identifies the eigenvectors of the covariance matrix, called principal components, along which the variance in the data is maximized, as illustrated in Figure 4.5. The original data is then transformed by projecting it onto the top n components with the largest eigenvalues, which capture the most variance in the data, reducing the dimension of each data point to n. In the following Experiments, we will set $n = 500$, as this has been shown to deliver optimal results. Additionally, it is important to normalize all feature vectors before employing PCA. This ensures that each feature vector contributes equally to the analysis, preventing features with larger scales from dominating the results.

PCA improves the results of clustering algorithms like GMM by reducing the dimension of all data points while maximizing the data variance and transforming them into uncorrelated components. GMM assumes that features are uncorrelated, consequently high correlations can distort the shape of the Gaussian components, leading to slower convergence and less accurate clustering. By projecting the data onto the orthogonal principal components, PCA retains the most important patterns (uncorrelated features with maximized variance), which allows GMM to identify clusters more effectively and accurately.

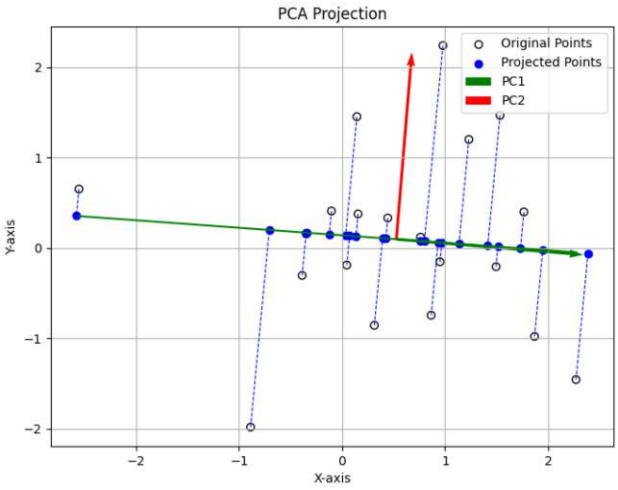

Figure 4.5: **Example of PCA:** The original data points are projected onto the principal component (green) that retains the most variance in the data, reducing the dimensionality from two to one.

Gaussian Mixture Models (GMM)

Many state-of-the-art unsupervised summary generation approaches, such as GVSUM [51], use K-Means clustering to group images or feature vectors based on similarity. We argue however, that when dealing with more complex data, such as feature vectors of human actions in videos, K-Means assumption of spherical clusters becomes limiting. Actions often exhibit complex dynamics and may not be well-separated in feature space. In contrast to K-Means, Gaussian Mixture Models (GMM) assume that data is generated from a mixture of Gaussian distributions, allowing them to capture clusters of varying shapes, sizes, and orientations, as illustrated in Figure 4.6.

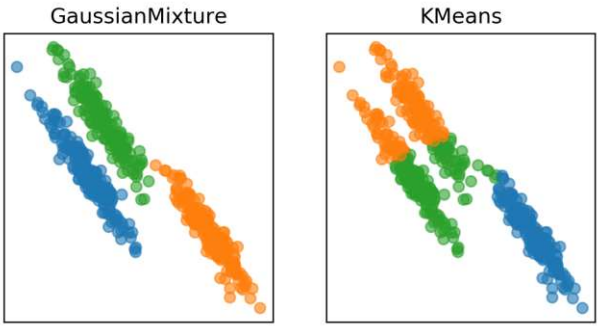

Figure 4.6: **GMM vs K-Means Clustering:** On complex, non spherical data distributions, the K-Means algorithm fails to cluster the data correctly, while the GMM effectively models the covariance structure of the data. [55]

The GMM uses the Expectation-Maximization (EM) algorithm to iteratively estimate the parameters (means, variances, and mixing coefficients) of these distributions, assigning data points to clusters based on their likelihood of belonging to each Gaussian. Two criteria can be applied to determine the optimal number of Gaussian distributions (clusters)

automatically, balancing model fit and complexity. The Akaike Information Criterion (AIC) focuses on minimizing information loss, thereby favoring more complex models, while the Bayesian Information Criterion (BIC) penalizes complex models more heavily. By choosing the AIC criteria, the framework is able to detect finer-grained action transitions, as it allows more complex models that can better capture subtle variations in the data. On the other hand, for coarser summaries, where the focus is on identifying broad, distinct actions, the BIC criteria would likely be a better fit. To determine the optimal number of clusters, multiple models with varying numbers of Gaussian distributions are tested, and the one with the lowest score is selected.

Additionally, GMM clustering can converge to sub-optimal solutions due to its sensitivity to the initial parameters, which can result in poor cluster assignments. To address this, we run the algorithm multiple times with different initializations and select the model with the highest log-likelihood value as the final clustering solution.

4.4.2 Temporal Smoothing and Boundary Detection

Temporal smoothing is a crucial step in reducing noise and abrupt label changes. Fluctuations in raw cluster-label sequences can result from minor variations or inconsistencies in the data, even when the underlying pattern remains stable. By employing smoothing techniques such as majority filtering, we can more accurately identify significant patterns, leading to a more precise detection of action boundaries.

Majority filtering operates by using a fixed-size window that slides over the temporal label sequence, selecting the most frequent value within each window as the representative label for the central position, as illustrated in Figure 4.4. Majority filtering is particularly effective in tasks such as denoising and smoothing labels in sequence labeling applications. In our Experiments, this window has a size of 25, corresponding to nearly 2 seconds of video content ($25 \times$ sampling rate), effectively removing noise and insignificant small actions while still capturing fine-grained action transitions.

Following the temporal smoothing stage, the boundaries detection stage identifies boundaries whenever the label in the temporal sequence of smoothed cluster labels changes. Such a label change indicates a transition between two different actions, as illustrated in Figure 4.4. In addition to label-based action boundary detection, the framework uses tracker information to identify the disappearance and reappearance of certain individuals in the video sequence. If the absence of a specific ID exceeds a predefined threshold, Enter and Leave Boundaries are established. This threshold is large enough to ensure that temporary occlusions do not result in false boundaries.

During our experiments in Section 6, we will compare three different approaches to automatically determine the optimal number of GMM clusters. This comparison aims to identify the best method for detecting all significant action transitions while minimizing the over-detection of small-grained action transitions.

- **The AIC Approach** uses the AIC score to determine the optimal number of GMM clusters, favoring more complex models. While this leads to a finer action granularity, it may also result in an over-detection of minor action transitions.
- **The Reduced Feature Approach** sub-samples each fourth (α) feature vector from the SlowFast Network. Reducing the number of feature vectors helps the

AIC-based clustering to generalize better, avoiding over-detections of minor action changes.

- **The BIC Approach** utilizes the BIC score to find the optimal number of GMM clusters, favoring simpler, more generalized models. While this simplification reduces the risk of over-detection, it may result in missing crucial, finer action transitions.

4.5 Summary Generation

Automatic video summary generation approaches can either be image or video-based, both aiming to produce a concise summary of the video by selecting its most informal parts. In this work, we opted for an image-based timeline approach, selecting one or three keyframes for each detected action instance within the video and placing them on a timeline that marks the detected action boundaries. The single keyframe method selects a frame positioned at the midpoint between two detected action boundaries, providing a simple visual representation of the action. In contrast, the three keyframe method captures additional context by selecting images at each action boundary, as well as one in the middle. While this method enhances the clarity of action context and transitions, it also doubles the number of frames in the summary.

We argue that by first detecting person-specific action boundaries, our approach ensures a comprehensive and high-quality summary. Unlike other unsupervised approaches that analyze the entire scene at once, thereby capturing irrelevant background information involving other individuals or objects, our method focuses specifically on a targeted individual. This results in a more focused and informative summary that clearly reflects the course of action of the selected individual, as shown in Figure 6.15.

In this chapter we presented the Unsupervised Spatio-Temporal Action Boundary Localization (UnSTABL) framework. In contrast to other state-of-the-art approaches, our method focuses on regions of interest, specifically the individuals detected by the person tracking module, rather than processing the entire scene at once. By analyzing person-specific action boundaries, UnSTABL significantly enhances the accuracy in crowded, real-world environments with multiple concurrent actions.

Given the limited availability of annotated data, we opted for an unsupervised approach. We utilized the SlowFast Action Detection Network for feature extraction, capturing detailed, person-specific action information across the entire video. These feature vectors were optimized through Principal Component Analysis (PCA) to reduce dimensionality and increase their information value. They were clustered using Gaussian Mixture Models (GMMs) to group similar action segments together. The resulting cluster labels were then temporally smoothed to eliminate smaller inconsistencies and actions. As a last step, action boundaries were identified at timestamps where changes occurred in the smoothed label sequence, marking transitions between distinct actions. We proposed three distinct boundary detection models, which will be thoroughly evaluated during our experiments.

The detected action boundaries serve two key purposes: generating concise video summaries and providing temporal action proposals for tasks such as temporal action localization and action segmentation. The UnSTABL framework demonstrates how unsupervised methods can effectively handle complex real-world scenarios, producing robust and accurate action boundaries without the need for extensive labeled data.

5 Collision Robust UnSTABL

Crowded environments pose significant challenges for person tracking and action detection algorithms. The main issue in such environments arises from person occlusions and crossovers, leading to wrong detected action boundaries and ID swaps. This challenge is particularly prominent in operating rooms (ORs), where medical staff wear identical clothing, making visual features unreliable for maintaining consistent identities.

During overlaps, the system can only detect the action of the person in the foreground and mistakenly assigns this action to both individuals. For example, in the second image of Figure 5.1, both individuals' actions are labeled as "walking". Consequently, the framework identifies an action change, assigning an incorrect action boundary to the person. Additionally, when people cross paths, the tracking system often loses track of the person in the background (Fig.5.1c). Current state-of-the-art trackers rely on visual features to reassign the correct IDs after such collisions. However, in our OR setting, where individuals have highly similar appearances, the tracker struggles to distinguish between them, often resulting in ID swaps or the assignment of new IDs, as shown in Figure 5.1d.

Figure 5.1: **Collision Problem:** In the first image, the action detection algorithm correctly identifies the individual actions "sitting" and "walking", while in the second image both individuals are mistakenly assigned the same action "walking" due to the overlap. In the last two images, the tracker loses track of the background person and assigns her a new ID after the crossover.

To tackle these challenges in our OR setting, we propose the collision-robust UnSTABL framework. Building upon the base framework from the previous chapter, we are able to improve the action boundary detection, as well as the tracker performance in crowded and complex multi-person environments.

5.1 Model overview

The Collision-Robust UnSTABL framework, built upon the base module, is specifically designed to handle collisions more effectively in the complex OR environment. It incor-

porates all three stages from the previous chapter (Fig.4.1) while introducing two major extensions, as illustrated in Figure 5.2. These extensions address specific challenges in crowded environments, each serving a distinct purpose:

- **Collision Avoidance Module:** The Collision Avoidance Module detects collisions and overlaps between individuals and excludes the feature vectors associated with these overlap sequences. By removing these features, the module prevents the incorrect assignment of foreground action information to overlapped individuals, ensuring a collision-robust action boundary detection.
- **ID Correction Module:** The ID Correction Module is designed to improve the Re-Identification (ReID) of individuals after strong overlaps, particularly in cases where visual features alone are insufficient. The module can detect and correct potential ID swaps or incorrect new ID assignments by comparing action information before and after the overlap. As a result, the module improves tracking accuracy in challenging conditions like the OR.

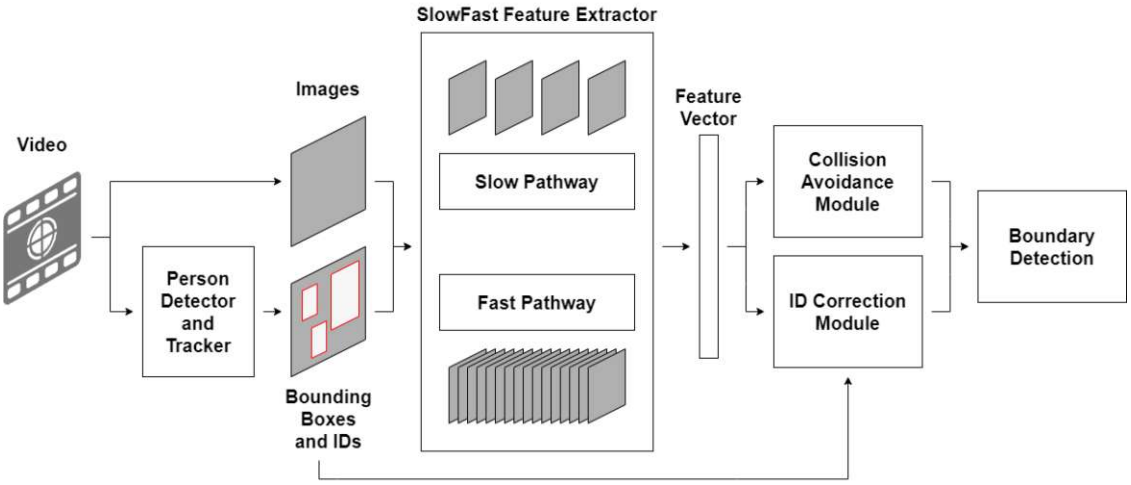

Figure 5.2: **Collision Robust UnSTABL Model Overview:** The framework employs all three stages of the base model: The (1) Person Detector and Tracker Stage, the (2) SlowFast Feature Extraction Stage, and the (3) Boundary Detection Stage. Additionally, it introduces two new modules: A (4) Collision Avoidance Module, which ensures a collision robust boundary detection and a (5) ID Correction Module to improve the tracker's Re-Identification after overlaps.

Both of the above modules enhance the spatio-temporal action boundary detection in complex and crowded environments and will be discussed in greater detail in the following subsections.

5.2 Collision Avoidance Module

The Collision Avoidance Module prevents incorrect action boundary detection resulting from overlapping individuals. In these cases, the bounding primarily captures the person

in the foreground (Fig.5.1b), causing the Feature Extractor to retrieve inaccurate action information. When the two individuals perform different actions, the framework detects an action change for the person in the background, resulting in an incorrect boundary detection during the overlap sequence.

To address this problem, the module operates in two phases: (1) **collision detection**, which identifies when and where individuals overlap, and (2) **collision avoidance**, which prevents the system from using incorrect action information during overlaps, ensuring a collision robust action boundary detection.

5.2.1 Collision Detection

The first step towards collision-robust action boundary detection is accurately and reliably identifying collision intervals. This is accomplished by computing the Intersection over Union (IoU) of all persons bounding boxes within each video frame. The IoU is defined as:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} = \frac{|A \cap B|}{|A \cup B|} \tag{5.1}$$

where A and B represent the bounding boxes of two individuals, $A \cap B$ denotes the area of overlap, and $A \cup B$ represents the total area covered by both bounding boxes. If the IoU exceeds a certain threshold, a potential collision is identified. During overlaps, the Person

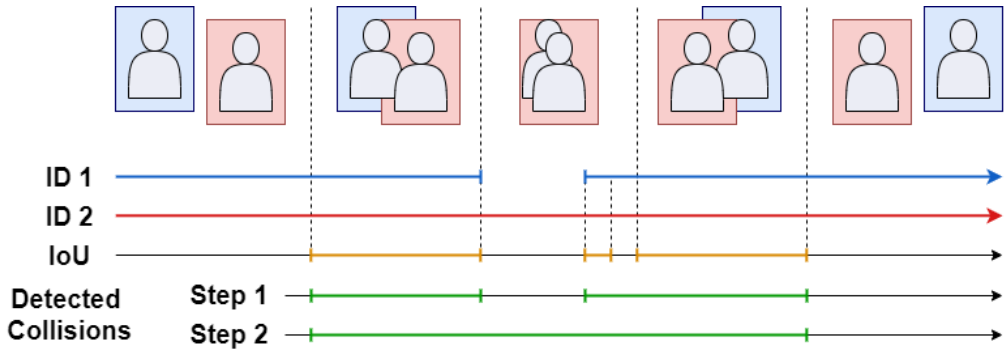

Figure 5.3: **Collision Detection:** A potential collision (yellow) is detected when the Intersection over Union (IoU) between two or more bounding boxes exceeds a predefined threshold. Through a two-step process (green), adjacent IoU collision that are part of the same overlap sequence are accurately merged.

Tracker sometimes loses track of the person in the background, causing their bounding box to disappear, which prevents the detection of an IoU overlap, as illustrated in Figure 5.3. As a result, the complete overlap sequence would be split into multiple parts that fail to capture the entire collision. To address this, we propose a two-step approach that identifies and merges adjacent IoU collisions involving the same individuals, correctly grouping them into a continuous overlap sequence:

- **Step 1:** During overlaps, the tracker can slightly shift the bounding box when a person is not fully visible, causing the IoU to drop below the overlap threshold. By merging consecutive IoU overlaps with the same IDs and a temporal gap smaller than a predefined threshold, we can compensate for these shifts.

- **Step 2:** In the second step, even longer overlap sequences are merged, provided that at least one of the involved IDs disappears for the entire duration between the sequences. This absence indicates that one individual was fully occluded by another, causing the Person Detector to lose track of them, splitting the overlap sequence in two.

This Collision Detection algorithm efficiently detects and combines overlap sequences, ensuring that fragmented IoU collisions are accurately merged. It forms the basis for the Collision Avoidance and the ID Correction modules.

5.2.2 Collision Avoidance

To effectively filter out collisions that could lead to incorrect action boundaries, we classify collisions into two types: strong and weak. This way, we can ensure that only feature vectors containing misinterpreted action information are removed while preserving those that could contain relevant action changes.

Strong collisions are characterized by a high IoU overlap or the disappearance of one ID, indicating a significant degree of occlusion. During such overlaps, the framework likely extracts incorrect action information. Weak overlaps, which have a lower IoU, are subdivided into weak short and weak long overlaps. Weak, long overlaps typically suggest individuals work or stand side by side with no significant person overlap. Consequently, the framework is still able to extract the correct action information, making the feature vectors relevant for boundary detection and are therefore retained. On the other hand, strong and weak short overlaps often occur when individuals briefly pass or overlap each other and are thus filtered out to avoid incorrect boundary detection.

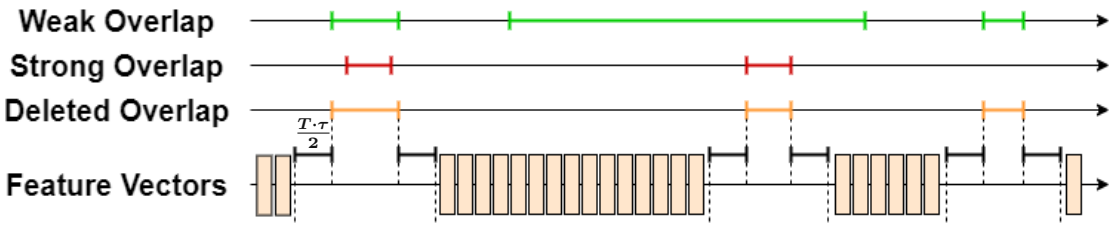

Figure 5.4: **Collision Avoidance:** Weak short and strong overlaps are combined and all associated feature vectors are removed to prevent misinterpreted action boundaries. Feature vectors in the range of $\frac{1}{2}(T \times \tau)$ frames still contain residual overlap information (black). Weak long overlaps are retained since they often contain relevant action information (e.g. collaborations, interactions).

As illustrated in Figure 5.4, we group weak short and strong overlaps together and remove all feature vectors associated with them to prevent the assignment of incorrect action information. Additionally, as discussed in the Section 4.3, we also remove all feature vectors within a range of $\frac{1}{2}(T \times \tau) = 32$ frames around the overlap (indicated in black in Fig.5.4), since these feature vectors still contain residual information from the overlap.

The Collision Avoidance Module ensures a collision-robust action boundary detection by filtering out potentially misinterpreted action information retrieved during overlaps.

Despite removing this information, the framework is still able to detect a single action change during short collisions, as it identifies the new action cluster label after the overlap with a minor temporal delay. However, during longer overlap sequences with multiple action changes, the framework can no longer detect all these transitions. Since it removed all associated feature vectors, it only notices that an action change occurred during the overlap, but not the number of changes and their exact timing. Consequently, longer or frequent overlaps pose a limitation to this approach.

5.3 ID Correction Module

The ID Correction Module is designed to improve the Re-Identification of individuals after complete overlaps in complex surgical environments. The Person Tracker introduced in Section 4.2 relies on FastReID [56] to reassign consistent IDs to individuals after overlaps or disappearances. However, FastReID primarily relies on visual features, which is insufficient in surgical environments where all individuals wear similar attire and face masks. Due to the lack of distinctive visual features, the tracker frequently confuses individuals' IDs or assigns them new IDs after strong overlaps.

To overcome these limitations, we propose a ID Correction Module that leverages action information extracted from the SlowFast Feature Extractor to detect such ID Swaps. By reassigning different IDs to the same person, this module enhances the Trackers performance in challenging conditions.

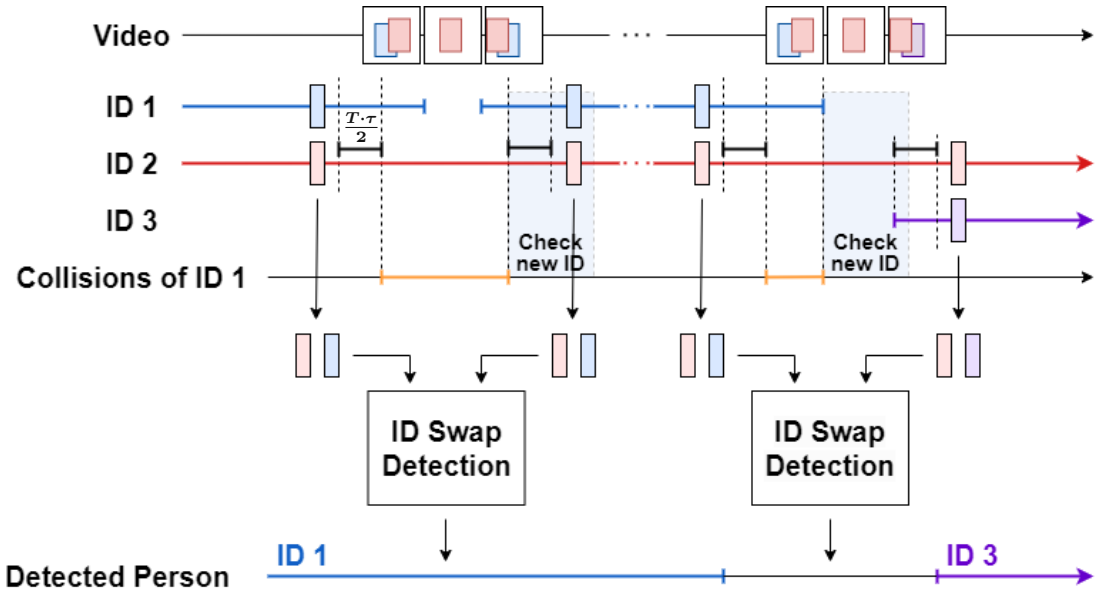

Figure 5.5: **ID Correction Module:** In the ID Correction process, strong overlaps of the target ID are identified. After each collision, the module checks for new IDs appearing at the same location. It then selects collision-free feature vectors before and after the overlap of all involved and new IDs. The ID Swap Detection module then determines whether an ID swap occurred, potentially merging different IDs to the same individual.

The ID Correction Module has two primary stages: (1) collision detection and feature selection and (2) ID Swap Detection. For collision detection, we employ the method outlined in Section 5.2.1. The process begins by selecting a target ID for which the ID correction will be performed, as shown in Figure 5.5 with ID 1. Each strong overlap with the target ID, or its permanent disappearance, is considered a potential cause for an ID swap. After such an overlap or disappearance, the algorithm assesses whether a potential new ID appears that (a) overlaps with the target ID's bounding box by exceeding a predefined IoU threshold and (b) emerges within a maximum number of frames after the overlap ends. As a next step, collision-free features are extracted $\frac{1}{2}(T \times \tau) = 32$ frames before and after each overlap for all involved IDs, including any newly detected IDs. These feature vectors are then evaluated in the ID Swap Detection Stage to determine whether an ID swap or new ID assignment occurred during the overlap, potentially linking different IDs to the same Person. However, if the overlap is too long, the actions of the involved individuals most likely changed during the overlap duration. Consequently, to prevent incorrect ID assignments, the ID Swap Detection module only checks short, strong overlaps for potential ID Swaps.

5.3.1 ID Swap Detection

The ID Swap Detection module is responsible for identifying whether an ID swap of our target ID has occurred during a strong overlap. This process is inspired by traditional Re-Identification (ReID) methods, but instead of relying on visual features, it utilizes collision-free action features extracted from the SlowFast network.

Figure 5.6 provides an overview of the ID Swap Detection process. All possible permutations of the previously selected feature vectors before and after collision are compared to find the most likely match. The likelihood Decision Step ensures that only very likely ID Swaps are selected, effectively minimizing incorrect ID assignments. The following subsections will briefly explain all components of the ID swap detection module.

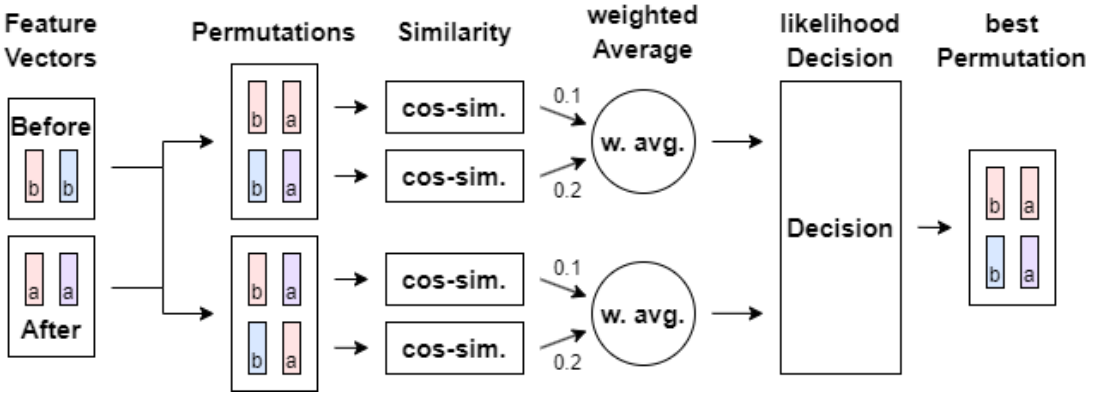

Figure 5.6: **ID Swap Detection:** All possible permutations of the feature vectors before and after the collision are compared using cosine similarity and a weighted average to determine the best permutation. The Likelihood Decision Stage then selects the best most likely match for the target ID (blue), ensuring accurate ID reassignments while minimizing errors.

Cosine Similarity

Each feature vector pair of a possible permutation is compared using cosine similarity, which evaluates the angular distance between the vectors to assess their similarity. Cosine similarity is particularly effective for high-dimensional feature vectors since it assesses whether two vectors are pointing in roughly the same direction, indicating similar information content while ignoring scale differences. The cosine similarity is defined as:

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\|\|B\|}$$

where A and B represent the feature vectors, and the result ranges from -1 (completely dissimilar) to 1 (identical).

Weighted Average

After computing the cosine similarity for each feature vector pair, a weighted average similarity score is calculated for each group of permutations. The weighted approach places a higher importance on the target ID by doubling the weight for feature pairs starting with this ID, as we are primarily interested in its state after the collision. All other feature pairs just provide supplementary information to enhance the overall accuracy of the ID swap detection and are therefore weighted lower.

Likelihood Decision

As a last step, we propose a likelihood Decision Stage to select the most likely permutation while minimizing incorrect ID assignments. We consider only permutations with an average similarity score above a certain threshold, set at 0.6. This ensures that ID swaps are only detected when the action of the target ID remains consistent throughout the overlap. If no permutation meets this threshold, indicating a likely action change, the system can no longer identify ID swaps based on action information alone, therefore it relies on the trackers Re-Identification. However, if multiple permutations have a similar likelihood scores (within ± 0.05 of each other), indicating similar actions across multiple individuals, a three-step process is employed to determine the optimal match for the target ID:

1. **Majority Check:** First, we check if the majority of the best-scoring permutations recommend the same new ID for the target ID. If a majority is found, we select that ID as the most likely match.

2. **Similarity to Tracker:** If a majority consensus is not reached, we prioritize the permutations that align most closely with the original tracker's ID prediction. This way, we integrate the tracker's visual information into our action-based decision-making process to find the optimal ID match.

3. **Highest Permutation Score:** If multiple permutations still show equal similarity to the tracker's detection, we select the permutation with the highest overall similarity score from those tied options.

This multi-step process ensures that only the most likely match for the target ID is selected, enhancing the accuracy of the trackers ID reassignment while minimizing errors.

In this chapter, we introduced a collision robust extension to our UnSTABL framework to improve unsupervised action boundary detection in crowded, real-world environments. By introducing the ID Correction Module, we significantly improved the performance of the person tracker in challenging conditions. It links different IDs and their associated action boundaries to the same person by analyzing and comparing action data before and after significant overlaps.

Furthermore, the integration of the Collision Avoidance Module allowed us to minimize incorrectly detected action boundaries caused by overlapping individuals, thereby enhancing the accuracy of action boundary detection in complex and crowded environments, such as the operating room.

6 Experiments and Results

This chapter provides a detailed overview of the experiments conducted to evaluate the performance of the proposed UnSTABL framework. The experiments are split into two parts: In the first part, we compare our action boundary detection approach against state-of-the-art methods to establish a performance baseline. In the second part, we evaluate the framework in real-world OR environments, introducing challenges like frequent collisions, overlaps, and occlusions. Additionally, we determine how well our collision-robust UnSTABL framework improves person tracking in this challenging environment.

6.1 Setup

This section presents the datasets and evaluation metrics used in our experiments. The datasets have been selected to cover a range of conditions, from controlled single-person environments to real-world complexity, while the evaluation metrics provide a thorough assessment of the system's performance in action boundary detection.

6.1.1 Datasets

To conduct a comprehensive evaluation of the proposed action boundary detection framework, we utilize three distinct datasets. The first two are used to establish a performance baseline in single-person environments. Due to the absence of labeled datasets capturing complex, multi-person environments such as operating rooms, we opted for these simpler, controlled settings to enable direct comparison with state-of-the-art approaches. The third dataset then introduces the complexities of real-world OR scenarios to provide a qualitative performance analysis in multi-person environments where challenges such as occlusion, overlaps, and collisions are prevalent. Additionally, a fourth test video is included to demonstrate the framework's adaptability to various real-world environments beyond the OR.

Breakfast Dataset

Unlike more complex scenarios, such as those found in operating rooms, the Breakfast dataset features only a single person performing semi-scripted tasks for short durations, as highlighted in Section 3.3.2. However, as we will later demonstrate, our frameworks person-based approach, which focuses on detecting the actions of individual subjects while disregarding background information, allows it to maintain similar performance in more complex and crowded settings. This makes the Breakfast dataset a valuable benchmark for assessing the accuracy and quality of detected action boundaries, comparing our framework with other state-of-the-art approaches. For our evaluation, we utilized a selected subset of the Breakfast dataset. Since our algorithm relies on accurate person detection to identify action boundaries, we excluded videos where humans were only partially visible (e.g.,

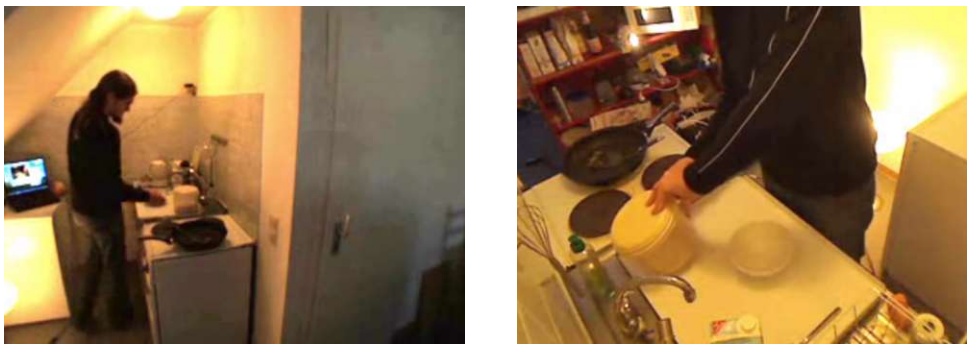

Figure 6.1: **Example Breakfast Dataset:** Examples of selected and excluded videos from the Breakfast dataset: (a) The first image shows videos used in our evaluation, where the entire person is visible, while (b) the second image shows excluded videos, where only partial visibility made person detection unreliable.

only hands or parts of the body) as illustrated in Figure 6.1, since the Person Detector struggled to identify individuals in these Videos. The specific list of videos used is detailed in the Appendix A.

Toyota Smarthome Untrimmed (TSU) Dataset

We selected the Toyota Smarthome Untrimmed (TSU) dataset as a second benchmark dataset due to its ability to simulate real-world challenges, as described in Section 3.3.4. Unlike the Breakfast dataset, the TSU dataset captures unscripted daily activities with a high degree of complexity. These long, untrimmed videos are recorded in various domestic environments, such as kitchens and living rooms, and include temporally overlapping and concurrent activities of different lengths.

These challenges reflect the real-world operating room environment, where spontaneous behaviors and overlapping actions sequences are common, making the TSU dataset comparable to the conditions in the OR. Additionally, the videos' long and untrimmed "surveillance-like" structure, similar to those in the OR, allows us to evaluate how the framework handles action boundary detection in these extended video sequences.

For our evaluation, we slightly modified the ground truth labels by merging "Enter" and "Leave" events into a single boundary (midpoint of start and end) to align the labels with our detection approach. It is important to highlight that the dataset was originally designed for temporal action localization, which distinguishes between foreground and background actions. Background actions are sequences where no significant labled action occurs (e.g. person standing in the room). However, since our unsupervised action localization approach merges consecutive boundaries into action segments without identifying background activities, we expect sub-optimal performance regarding the action-based results.

OR Dataset

To qualitatively evaluate the performance of our action boundary detection framework in complex operating room environments, similar to [57] and [58], we also created a small custom dataset consisting of 10 untrimmed videos, as no publicly available datasets exist

for our specific scenario. Each video, ranging from 5 to 10 minutes, captures different OR-related activities in a challenging multi-human environment, featuring various camera viewpoints, operating room layouts, and light conditions. These videos include multiple individuals, frequent overlaps and collisions, as well as occasional interactions between participants, all dressed in very similar attire.

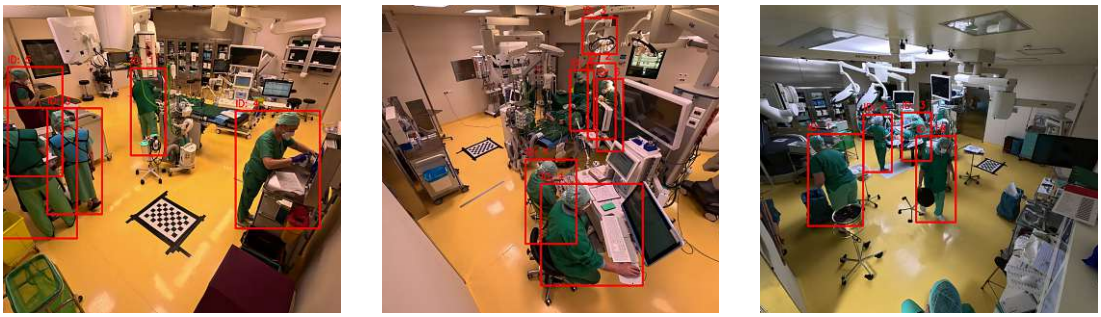

Figure 6.2: **The OR Dataset video classes:** (1) The first class features a single Person (right) performing tasks in a multi-human environment, (2) the second class showcases two-person collaborations (on the monitor), and (3) the third class involves a single person (ID 3, center) working in a densely crowded multi-human setting.

In a second step, we divided the OR dataset into three video classes of increasing complexity, as shown in Figure 6.2, to thoroughly evaluate the robustness and limitations of our "collision-robust" UnSTABL framework:

- **Single-Person in Multi-Human Environment:** This class includes action sequences performed by a single individual within a multi-human environment. The videos are kept relatively simple, with persons performing individual tasks and only occasional interactions or crossovers.
- **Two-Person Collaboration:** This class includes scenarios where two individuals work side by side, often collaborating on tasks. It evaluates the framework's ability to accurately detect boundaries in cooperative settings with constant partial occlusions.
- **Crowded Operating Table Setting:** The most complex class involves a person working in a highly crowded environment, such as an operating table, with frequent occlusions and crossovers. This setting presents significant challenges due to the high density of people and overlapping actions, pushing the limits of our framework.

Although we perform only qualitative evaluation due to the lack of ground truth labels, this dataset allows us to validate the benchmark results from the previous datasets in this challenging environment and to identify the limits of our action boundary detection approach. Additionally, we are able to assess the improvements of our collision-robust framework and how well it enhances boundary detection and tracker performance in demanding OR environments.

Industrial Dataset

As an additional evaluation, we include a video from an industrial human-robot collaboration scenario. This allows us to demonstrate the adaptability of our UnSTABL framework

to different environments, thanks to its unsupervised nature. The video features a clearly structured sequence of actions and introduces challenges such as the presence of multiple humans, along with minor overlaps and collisions, as illustrated in Figure 6.3.

Figure 6.3: **The Industrial Video** showcases a human-robot collaboration scenario with a clearly structured course of action and the presence of multiple individuals.

6.1.2 Evaluation Metrics

To objectively assess the performance of our action boundary detection framework on the proposed benchmark datasets, it is necessary to have appropriate evaluation metrics that quantify how effectively the framework detects actions. We categorize these metrics into two groups: (1) **boundary-level metrics**, which evaluate the accuracy of the detected boundaries between actions, and (2) **action-level metrics**, which assess how well the detected boundaries capture entire actions, ensuring a thorough evaluation of both boundary precision and action representation quality.

Boundary Level Metrics

Detected action boundaries should meet two key criteria: they must capture all the ground-truth boundaries to ensure high **recall**, and they should not exceed the number of actual boundaries to maintain high **precision**. These two metrics help evaluate how well the framework balances detecting all action transitions while avoiding false positives.

Precision is defined as the ratio of correctly detected boundaries (TP) to the total number of *detected* boundaries (TP + FP), evaluating the framework's ability to avoid false positives by ensuring that the detected boundaries correspond to real ground-truth transitions:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{6.1}$$

Recall on the other hand is the ratio of correctly detected boundaries (TP) to the total number of actual *ground-truth* boundaries (TP + FN), measuring the framework's ability to detect all ground-truth boundaries, ensuring that no true transitions are missed:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{6.2}$$

To offer a balanced assessment of the framework's performance, the **F1-Score** combines precision and recall into a single metric, measuring the framework's ability to accurately detect true action boundaries while minimizing false detections:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6.3}$$

In the context of action boundary detection:

- **True Positives (TP)** refer to detected boundaries that correctly correspond to actual ground truth boundaries.
- **False Positives (FP)** are detected boundaries that do not correspond to any ground-truth boundary, representing over-detections.
- **False Negatives (FN)** occur when a ground-truth boundary is missed by the detector, indicating a failure to detect a true action transition.

What remains to determine is the threshold used to decide whether a predicted boundary is considered a true positive. The standard approach is to use 5% of the video duration as an acceptable margin for error. However, similar to the argument made by Li in her OTAS paper [45], we find this threshold too coarse, particularly for longer, untrimmed videos such as those in the TSU dataset. Instead, we opted for Li's proposed smaller threshold of 2 seconds (± 1 second to the ground-truth boundary) for a more precise evaluation.

Action Level Metrics

Action-level metrics evaluate how well the detected action boundaries capture entire ground-truth actions. As highlighted in previous work such as BSN [47], action proposals should be retrieved with a high recall and a strong temporal overlap, typically measured using the temporal Intersection over Union.

In unsupervised settings, the algorithm typically produces unlabeled action segments, in our case represented by two consecutive action boundaries. To align these segments with ground-truth annotations, we use the Hungarian algorithm to establish a one-to-one correspondence, as demonstrated in previous works [43], [45]. The Hungarian algorithm performs optimal matching by associating each ground-truth action with the most likely detected action segment, based on maximizing the IoU.

The **Intersection over Union (IoU)** is a key metric for measuring the temporal overlap between a predicted action segment and its corresponding ground-truth action. Similar to the 2D IoU used to measure the bounding box overlaps (Eq.5.1), the temporal IoU calculates the ratio of the predicted and ground-truth time intervals to their union in one dimension.

Similar to boundary-level metrics, we evaluate the **Precision** and **Recall** of the detected action segments. The Equations 6.1 and 6.2 remain unchanged, but the context differs slightly:

- **True Positives (TP)** refer to detected action segments with an IoU greater than a predefined threshold when matched to a ground-truth action.
- **False Positives (FP)** detected action segments that either do not correspond to any ground-truth action or have an IoU below the threshold.

- **False Negatives (FN)** occur when a ground-truth action has no corresponding detected action segment.

As with boundary detection, it is necessary to define an acceptable true positive IoU threshold. Temporal action localization algorithms are typically evaluated using the average precision and recall across a range of IoU thresholds, starting at 0.5 and increasing to 0.95. For our evaluation, we argue that reliable temporal action proposals should achieve at least a 0.5 IoU overlap. Therefore, we report precision and recall at the 0.5 threshold (**Precision@0.5** and **Recall@0.5**) as a key metric in our action-level evaluation.

6.2 Action Boundary Detection

In this section, we will evaluate our detected boundaries quantitatively and qualitatively across four datasets of increasing complexity. The quantitative analysis is carried out on the first two Datasets, Breakfast, and TSU, using boundary-level and action-level metrics. The remaining two datasets are used for qualitative validation of these results in more complex, real-world scenarios, such as operating rooms and industrial environments, which feature multiple individuals and concurrent actions.

Boundary-level metrics assess how well our framework detects action transitions, while action-level metrics evaluate the accuracy of boundaries capturing entire actions. By using two subsequent boundaries as a single action proposal, we assess whether the framework correctly identifies actions or tends to over-detect them, splitting ground-truth actions into multiple sub-actions. To address this, as explained in Section 4.4, we compare three Boundary Detection models (AIC, Red. Features, and BIC) to determine which model best captures fine-grained transitions without excessive over-detection.

For our experiments, we used two detector models suited to the datasets characteristics. In the Breakfast and Smarthome datasets, individuals were tracked using the DeepOCSort network pre-trained on MOT17. For the OR dataset, we employed a model pre-trained on DanceTrack, which is optimized for environments where individuals wear similar attire but would result in numerous misdetections in more controlled environments.

6.2.1 Breakfast Dataset

We begin the quantitative evaluation of our boundary detection framework with the benchmark dataset Breakfast, highlighting the most significant results. The dataset features sequentially annotated breakfast preparation tasks characterized by relatively simple and coarsely defined actions. A comprehensive list of all results, along with details of the tested videos, can be found in Appendix A.

In terms of overall performance, all three models demonstrate a similar average boundary-level F1-score across all tested videos in the Breakfast dataset, as shown in Table 6.1. However, the average Precision and Recall differs notably across all models, indicating a different performance regarding the accuracy and reliability of the boundary detection. The AIC score favours more complex GMM models, resulting in the detection of more fine-grained action transitions as expected. It achieves the highest average recall, indicating the highest detection rate of ground-truth boundaries. However, this comes at the cost of a slight over-detection of boundaries, as reflected by the lower average precision score. The Reduced Feature vector approach helps the AIC model to generalize better, slightly

Table 6.1: **Boundary Level Results:** Average Precision, Recall and F1-Score over all tested videos of the Breakfast Dataset.

	AIC	AIC (Red. Features)	BIC
Precision	0.356	0.37	0.446
Recall	0.658	0.606	0.512
F1-Score	0.438	0.444	0.456

enhancing average precision, though at the cost of reduced recall. Given the coarser annotations of the dataset, the BIC score on the other hand provides the best overall performance by recommending simpler GMM clusters. This suggests that BIC's proposed boundaries closely match the ground-truth without excessive over-detection; however, it misses more ground-truth boundaries compared to the AIC approaches, indicated by the low recall.

We argue however, achieving a higher recall is more important for creating unsupervised video summaries, as it reduces the likelihood of missing important action transitions. While the lower precision may result in additional boundaries and extra frames in the summary, this is an acceptable trade-off to ensure comprehensive action coverage.

Table 6.2: **Comparison to SOTA:** Our UnSTABL framework delivers state-of-the-art results in terms of F1-Score for unsupervised Action Boundary Detection.

	F1-Score
ABD [43]	0.279
OTAS [45]	0.445
UnSTABL (ours)	**0.456**

A comparison of the UnSTABL framework with other unsupervised Action Boundary Detection methods, such as OTAS [45] and ABD [43], demonstrates state-of-the-art performance in terms of F1-score across all three models, as shown in Table 6.2. However, it is important to note that this comparison was conducted on a selected subset of the Breakfast dataset comprising 198 videos, as detailed in Section 6.1.1.

Boundary Results on different video classes

As the second step of our evaluation, we examined the boundary-level performance across different video classes within the Breakfast dataset, which include a diverse set of activities such as (1) tea, (6) sandwich, (7) scrambled egg, (9) salad, and (10) pancake. The results varied significantly between classes, especially in terms of boundary-level precision.

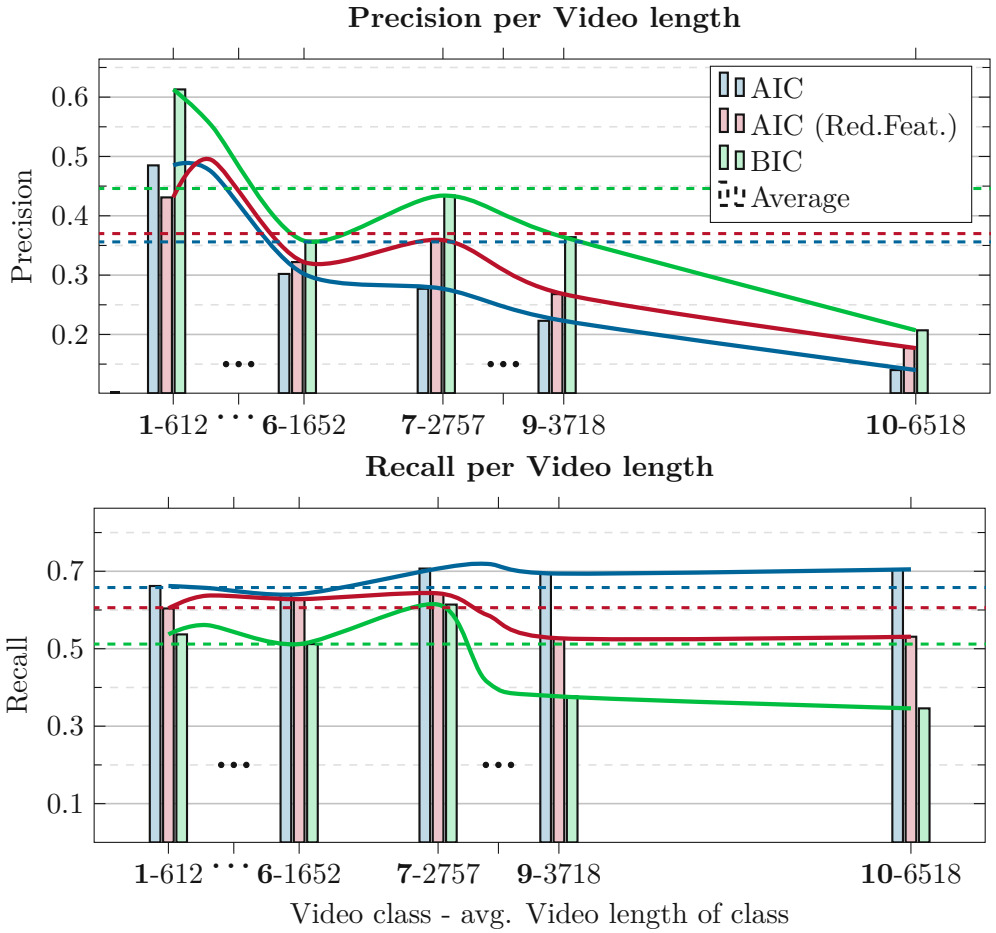

Figure 6.4: **Comparison of boundary-level Precision and Recall over the Breakfast Video classes:** The video classes (1) tea, (6) sandwich, (7) scrambled egg, (9) salad, and (10) pancake are ordered based on average video length. The precision decreases significantly with increasing video length, while the recall remains almost stable across all video classes.

When the video classes are ordered by average video length, a significant decline in precision is observed as the video length increases, as shown in Figure 6.4. However, this decline is not inherently caused by the longer duration of the videos, as will be further discussed in the next section. The primary issue lies in the coarse annotations of the dataset, particularly in longer videos like (10) pancake. These videos mainly feature lengthy ground-truth actions, such as "fry pancake," with an average duration of 2803 frames (approximately 2 minutes and 20 seconds). Our framework tends not to detect these as single actions but rather splits them into smaller, more fine-grained actions, as shown in Figure 6.5, while still detecting the broader boundaries defined in the ground-truth labels. This statement is supported by the relatively stable and high recall across all video classes, independent of the video length, especially for the AIC-based models (AIC and Reduced Features), as seen in Figure 6.4.

This suggests that while our framework consistently captures the main action boundaries

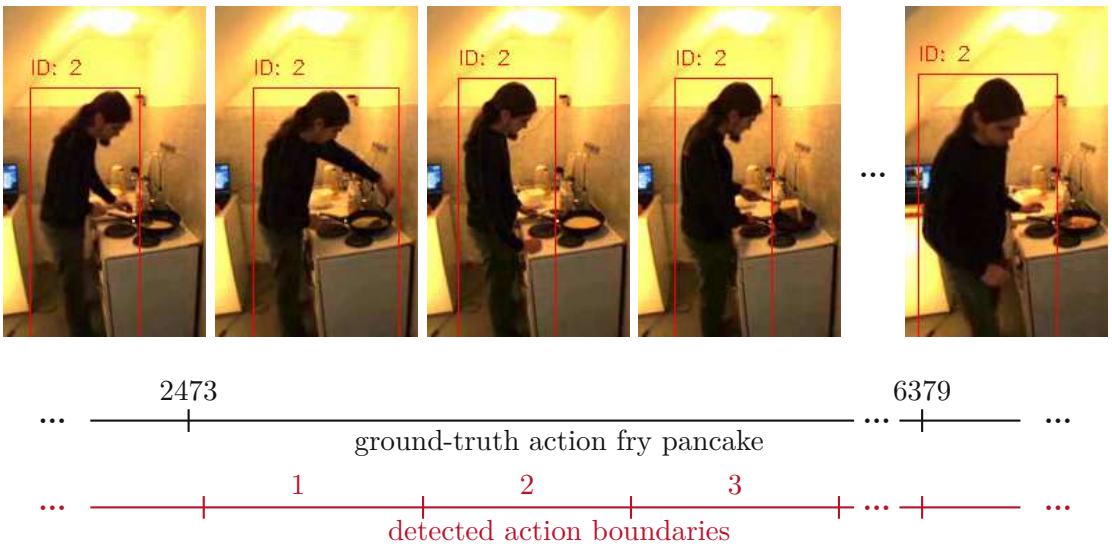

Figure 6.5: **Precision Problem of coarsely annotated actions:** The UnSTABL framework identifies more fine-grained action transitions, such as (1) working with a spatula, (2) placing the spatula aside, and (3) flipping the pancake, while still capturing the broader ground-truth boundaries of the "frying pancake" action.

across all classes and lengths, it additionally tends to identify more detailed, fine-grained action transitions, lowering the precision score in such coarsely labeled datasets. The full results for each video class are provided in the Appendix A in the Tables A.3 and A.4.

Action Level Results

As the third step of our evaluation, we analyzed the action-level metrics. A comparison of the action-level Precision@0.5 reveals significant differences among the three models, with the BIC-based model achieving by far the best result. This indicates that the boundaries proposed by the BIC model best capture entire ground-truth actions without excessive over-detection.

Table 6.3: **Action Level Results:** Average action-based Precision@0.5 and Recall@0.5 over all tested videos of the Breakfast Dataset.

	AIC	**AIC (Red. Features)**	**BIC**
Precision@0.5	0.282	0.299	0.433
Recall@0.5	0.497	0.486	0.466

These findings further support our observation that the AIC based methods struggle with the coarsely annotated dataset Breakfast, as they tent to detect more fine-grained action transitions. Consequently, larger ground-truth actions are frequently split into sub-actions, leading to an increase in false positives and a subsequent drop in precision. In contrast, the action-level recall remains nearly constant across all models, suggesting a comparable detection rate of ground-truth actions among the three approaches.

We illustrated the relationship between the average IoU and the ground-truth action length for all three models in Figure 6.6, highlighting their ability to detect shorter as well as longer ground-truth action intervals.

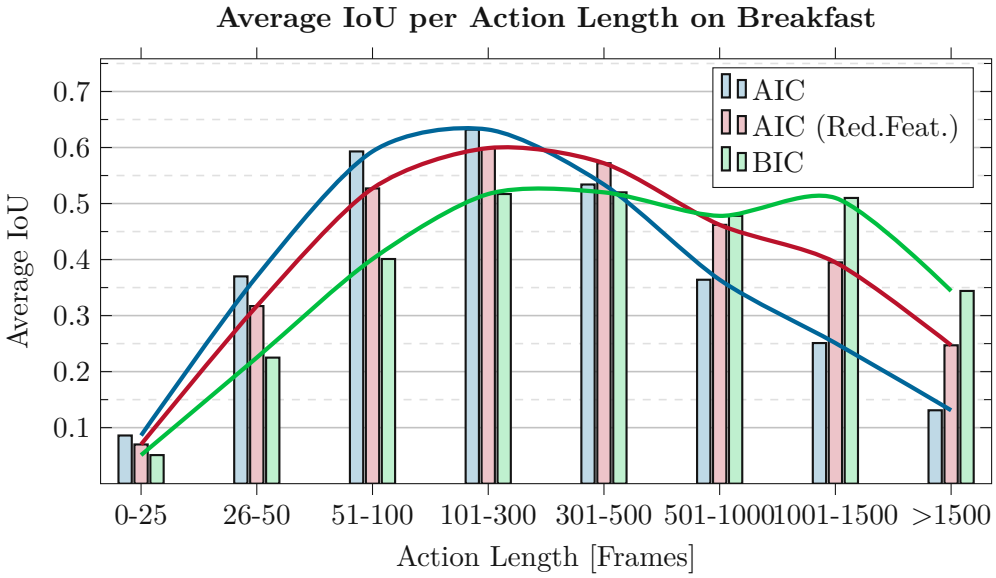

Figure 6.6: **Average IoU per Action length:** Both AIC based methods detect more fine-grained action transitions, enabling a more accurate detection of shorter ground-truth actions up to 30 seconds, while the BIC approach provides a better detection of longer action segments.

Ideally, we would observe a constant IoU across all ground-truth action lengths, indicating that our framework accurately detects action segments regardless of their duration. As expected however, this is challenging in an unsupervised setting, where there is no underlying information about the granularity of the ground-truth actions.

The AIC-based models deliver much better results than the BIC approach for shorter actions lasting up to approximately 30 seconds. This aligns with our previous observations that the AIC-based models tend to detect finer-grained action transitions, allowing for better detection of smaller and shorter actions while splitting longer actions into multiple sub-actions. Consequently, the BIC approach provides better accuracy for longer and more coarsely annotated actions.

Comparing the two AIC-based methods, the Reduced Feature approach enables the UnSTABL framwork to generalize more effectively, maintaining a high accuracy across a broader range of action lengths. Figure 6.6 illustrates this with an average IoU above 0.5 for actions ranging from around 2 seconds to nearly 50 seconds. This results in highly accurate detection of both short and moderately long actions, making it well-suited for our unsupervised summary generation approach.

6.2.2 Toyota Smarthome Dataset

The second part of our quantitative evaluation is conducted on the TSU dataset. As detailed in Section 6.1.1, the TSU dataset includes surveillance-like, untrimmed videos

that resemble those obtained in an operating room environment. It includes sequences with background actions (where no action occurs) and frequently overlapping foreground actions of varying durations, making it more challenging compared to the sequentially structured Breakfast dataset. This makes it an ideal benchmark for assessing whether the previously observed results can be replicated in longer untrimmed videos with more complexly annotated actions.

Comparing the overall boundary-level performance in Table 6.4, we observe clear differences between the three models. The AIC-based approach performs by far the best, achieving similar results to those on the Breakfast dataset, with even higher precision in detecting boundaries within these more complex settings. This increase in precision is

Table 6.4: **Boundary Level Results:** Average Precision, Recall and F1-Score over all tested videos of the TSU Dataset.

	AIC	**AIC (Red. Features)**	**BIC**
Precision	0.417	0.426	0.392
Recall	0.652	0.42	0.274
F1-Score	0.487	0.402	0.298

expected, as the TSU dataset is more densely annotated with shorter, more fine-grained actions where the AIC approach excels. In contrast, the other two approaches show a noticeable drop in performance. While the Reduced Feature approach maintains a high precision and reasonable recall, the BIC model recommends overly simplistic GMM clusters that are insufficient for accurate and robust boundary detection in the densely annotated setting. This performance supports our earlier claim that our framework, particularly the AIC-based approach, can effectively handle longer, untrimmed videos, achieving state-of-the-art results comparable to those on the Breakfast dataset.

Table 6.5: **Boundary Level Results over Video-classes:** Average Precision, Recall, and F1-Score for each video class in the TSU Dataset using the AIC-based Boundary Detection approach.

		AIC		
class	avg. length	**Precision**	**Recall**	**F1-Score**
Dining Room	23777	0.270	0.596	0.347
Kitchen	15191	0.499	0.708	0.569
Living Room	24319	0.406	0.621	0.472

By analyzing the boundary-level results per video class, as shown in Table 6.5, we again observe significant differences among them. Similar to the observations made on the Breakfast dataset, video classes with longer average lengths, such as those in the Dining and Living Room, experience a notable drop in precision. As before, these videos primarily contain longer action segments such as "watch TV, read, write, and use telephone/laptop" with average durations ranging from 30 seconds to several minutes.

In many of these videos, the person remains relatively stationary, for example, by sitting on the couch while shifting from reading to watching TV. The clustering algorithm tends

to slightly over-detect action changes, identifying non-labeled sub-actions like "grabbing the remote control" or "adjusting glasses" due to the overall low action content. An example of this over-detection during "watch TV" is shown in Figure 6.9.

These additional detections result in numerous false positives, reducing the overall precision score. However, as presented in the Tables B.4 and B.5 in Appendix B and reflected by the high recall score, our framework is still able to successfully detect most start and end boundaries of these longer action segments. This demonstrates our framework's ability to capture the significant action changes (with high recall) while occasionally identifying additional unlabeled and fine-grained transitions in low-action scenarios.

When comparing the average IoU per action length for all three models, as illustrated in Figure 6.7, we observe a similar pattern to that observed during our Breakfast evaluation.

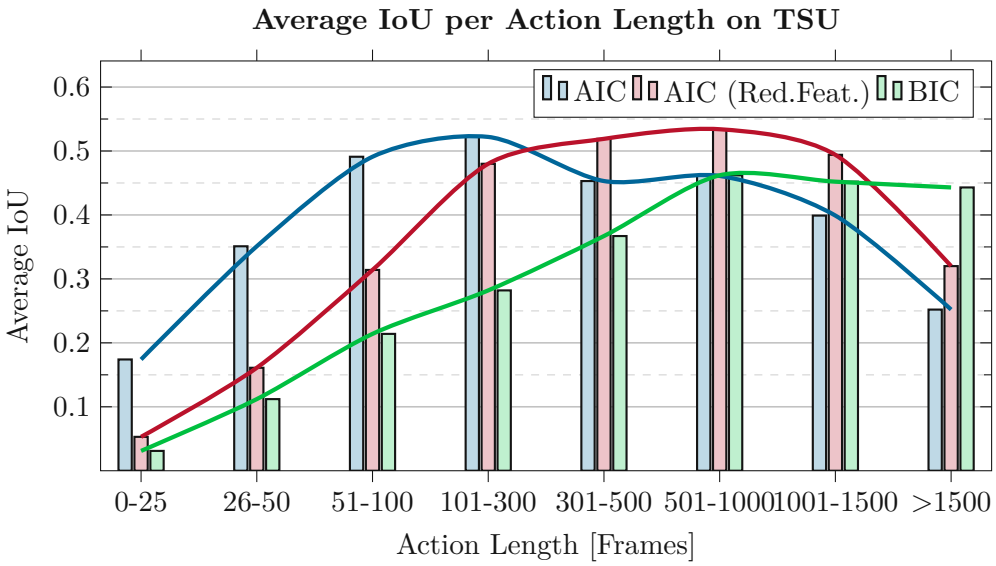

Figure 6.7: **Average IoU per Action length:** Both AIC based model provide good results on the TSU dataset. The standart AIC model excels at detecting shorter actions up to 10 seconds, while the Reduced Feature approach generalizes better, improving the average IoU for longer action up to a minute in duration. The BIC approach does not perform well, showing reasonable results only for very long actions (>1500 frames).

The AIC-based approach identifies fine-grained action transitions, achieving high detection accuracy for shorter actions ranging from one second to almost 40 seconds. The reduced feature vector approach helps the GMM generalize better, enabling the capture of broader action segments from several seconds up to nearly a minute. In contrast, the BIC approach recommends simple Gaussian clusters, primarily detecting very coarse boundaries of actions longer than 40 seconds. These observations explain the significantly lower recall scores of the Reduced Feature and BIC approaches, as over 50% of the dataset's ground-truth actions fall within the 0 to 2-second range (0-50 frames), where only the AIC model can effectively identify action transitions.

Action Level Results

The overall action-level results, presented in Table 6.6, are significantly worse across all three models compared to the Breakfast dataset. This indicates that, although the framework detects action transitions with high accuracy, capturing entire actions using two consecutive boundaries proves to be highly inaccurate in this challenging dataset.

Table 6.6: **Action Level Results:** Average action-based Precision@0.5 and Recall@0.5 over all tested videos of the TSU Dataset.

	AIC	AIC (Red. Features)	BIC
Precision@0.5	0.158	0.207	0.218
Recall@0.5	0.301	0.240	0.166

As previously noted, the dataset contains many very short action sequences, with over 50% of actions being under 50 frames in length. As shown in Figure 6.7, the average IoU for these small actions falls well below the true positive detection threshold of 0.5, explaining the low action-based precision and recall scores. The strict nature of the IoU metric poses a particular challenge for these smaller actions; for instance, in actions like "Put something on the table" with an average length of 20 frames, missing the boundaries by just 10 frames (equals to one-third of a second) results in an IoU below 0.5. This leads to the action being classified as undetected, even though the boundaries are correctly identified. The Tables B.4 and B.5 in Appendix B highlight this issue, showing that even though IoU-based action metrics are low, the AIC-based approach is still able to detect at least one or even both boundaries of most smaller actions.

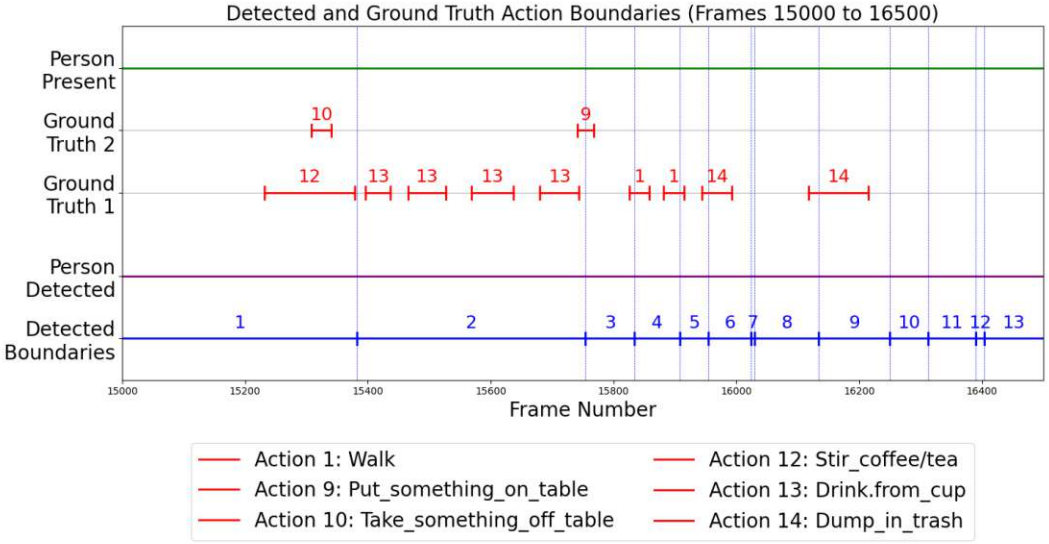

Figure 6.8: **Boundary Detection on TSU Dataset:** Comparison of ground-truth action intervals (red) with the retrieved action boundaries by the AIC-based UnSTABL framework (blue). Our framework only detects distinct action changes and, therefore, identifies consecutive identical action segments as one (actions 1 and 13). Additionally, it occasionally misses a start or end transition between fore- and background action (actions 9 and 12)..

Another contributing factor to the poor action-based recall performance is illustrated in Figure 6.8. Longer action sequences in the TSU dataset are often divided into individual, consecutive action intervals (e.g., ground-truth actions 13 and 1 in Figure 6.8). Our algorithm, however, primarily detects distinct action changes, identifying only the outer boundaries and merging these subsequent, identical action segments into a single continuous interval. As a result, the IoU for these smaller ground-truth action segments frequently falls below 0.5, leading to numerous false negatives and thereby reducing the action-level recall, while still detecting the outer, most important action boundaries.

A further issue arises from the overlapping and non-sequential action structure of the TSU dataset. As illustrated in Figure 6.9, the UnSTABL framework is still able to detect overlapping action boundaries if the action change is sufficiently large. However, these overlapping action boundaries cause our proposed action intervals to split, yielding an IoU below 0.5 for the longer ground-truth segments and contributing further to false negatives. Furthermore, the dataset contains many segments without annotated actions, known as background action segments. The transition between foreground and background actions is sometimes not clear enough, causing the algorithm to occasionally miss a start or end boundary (e.g. actions 9 and 12 in Fig.6.8). As a result, the framework merges background and foreground actions, leading to a misidentification of the ground-truth action.

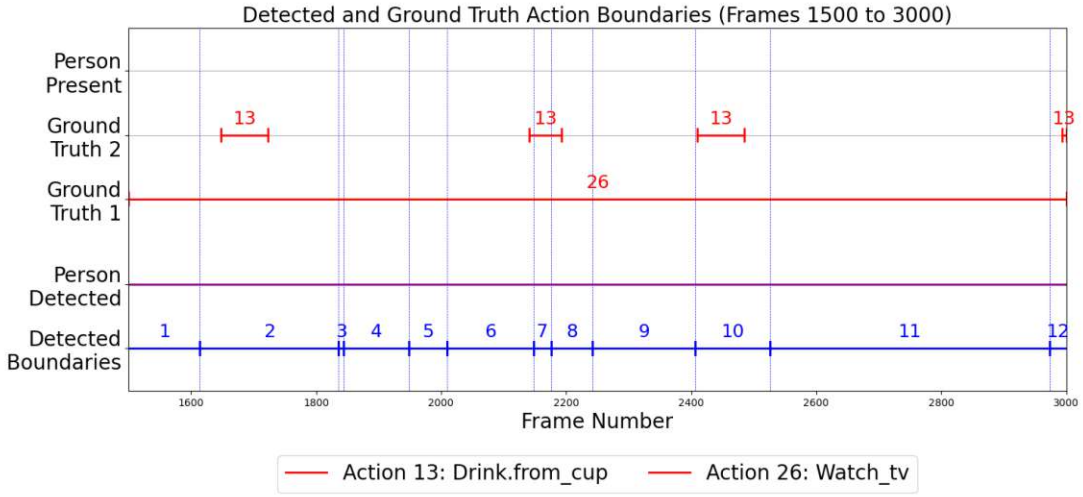

Figure 6.9: **Boundary Detection on TSU Dataset:** Comparison of ground-truth action intervals (red) with the retrieved action boundaries by the AIC-based UnSTABL framework (blue). Our framework splits longer, ground-truth action segments (action 26) into multiple, smaller action intervals by detecting overlapping ground-truth actions (action 13) or non-labeled actions or movements.

Regarding the poor performance of the action-based precision, we encounter challenges similar to those observed in the Breakfast dataset. In addition to the over-detection of longer actions (e.g. action 26 in Fig.6.9), there is also the issue of over-detecting background action segments, as shown in Figure 6.8. During these longer foreground or background action segments, the framework can detect minor, unintended movements, such as arm or hand motions, as action boundaries. This leads to a high number of false positives, significantly lowering the boundary-level and action-level precision scores.

To conclude the quantitative evaluation of both datasets, we highlight the strengths and limitations of the UnSTABL framework in boundary detection and action proposal generation. The high boundary-level recall across both datasets demonstrates the effectiveness of our models in detecting ground-truth action transitions. While all three approaches deliver state-of-the-art results in terms of F1-Score, they differ significantly in the granularity of the detected boundaries. By incorporating boundary-level precision and action-based metrics into our evaluation, we observe that both AIC-based methods excel at detecting fine-grained action transitions, whereas the BIC model focuses primarily on broader transitions. Both AIC-based models serve a distinct purpose:

- **The AIC Approach** detects very fine-grained action transitions, ensuring that fewer labeled ground-truth transitions are missed. It effectively detects actions ranging from one to 20 seconds. However, the higher recall comes at the cost of introducing additional, unnecessary boundaries, which lower the overall precision.
- **The AIC Reduced-Feature Approach** generalizes better, improving the boundary-level precision by reducing the over-detection of subtle, non-labeled action changes. By focusing on the most critical transitions, this approach is particularly proficient at detecting short and moderately long actions, ranging from several seconds up to 50 seconds.

However, the action-based precision and IoU metrics reveal limitations when generating temporal action proposals from two consecutive boundaries, especially on non-sequential datasets like TSU. Our framework struggles to handle overlapping and background actions, making it challenging to form accurate action proposals directly from the detected boundaries. These limitations suggest the need for an additional refinement stage to transform boundary detections into coherent action proposals, especially for non-sequential datasets.

Despite these limitations in temporal action proposal generation, our framework performs exceptionally well for unsupervised summary generation, effectively detecting key action transitions even under complex conditions. We argue that the Reduced Feature approach delivers the best result, providing an optimal trade-off between detecting most of the significant action changes while minimizing excessive over-detections.

6.2.3 OR Dataset

The third part of our evaluation is conducted on our custom OR dataset. As explained in Section 6.1.1, this experiment is performed on ten videos divided into three classes, each representing progressively higher levels of difficulty and crowdedness. By manually assessing the accuracy of the produced summaries, we aim to qualitatively validate whether the results previously observed in single-person environments remain similar in more complex multi-person surgical settings. Based on the previous results, we employ the Reduced Feature approach for boundary detection to achieve precise boundary detection with fewer over-detections, which is essential for creating concise and accurate summaries.

The boundary results, shown in Table 6.7, support our claim that the UnSTABL framework, through person-specific feature extraction and boundary detection, can produce strong results even in challenging multi-person environments. The performance and accuracy were consistent across all three video classes, further validating its robustness in

Table 6.7: **Boundary Performance on our OR Dataset:** Counts and percentages of correctly detected boundaries, incorrectly detected boundaries (split into redundant/wrong and person-collision-induced) and missed boundaries relative to the correctly detected ones.

Total det. Boundaries	correct det. Boundaries	incorrect det. Boundaries	incorrect det. Boundaries (due to Collision)	missed Boundaries
695	446 - **64.2%**	149 - **21.4%**	100 - **14.4%**	15 - **3.7%**

crowded scenes. The framework demonstrated high precision in detecting relevant and important action transitions, with nearly 65% of the detected boundaries considered useful in the summary. More importantly, our framework missed only 15 crucial transitions out of 446 correctly detected boundaries. These results are directly comparable to those achieved in both benchmark tests, where both AIC-based methods achieve a very high boundary-level recall, missing only a few significant action transitions. A full breakdown of these results can be found in Appendix C.

An example for such a good summary is given in Figure 6.10. In this instance, our UnSTABL framework successfully detects six distinct action intervals, efficiently summarizing 30 seconds of video without missing any important steps.

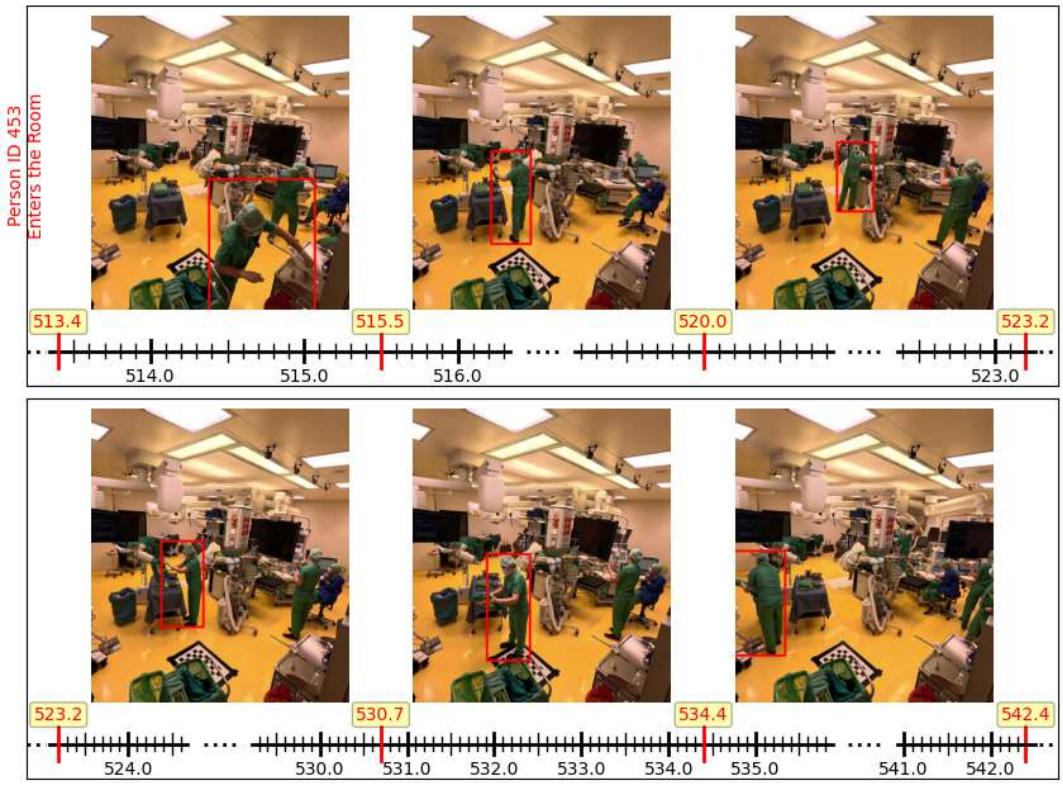

Figure 6.10: **OR Video Summary:** Our UnSTABL framework successfully detects six action segments performed by the individual: (1) placing the forceps, (2) walking to the patient and grabbing a surgical drape, (3) placing the drape on the patient's wound, (4) grabbing a new drape set, (5) unpacking it, and (6) discarding the packing paper in the garbage can.

Although the framework demonstrates high recall in detecting key action transitions, its unsupervised nature leads to around 35% of additional, incorrect detected boundaries. Of those, about 15% are caused by person collisions, as shown in Figure 6.11. When a second individual enters the bounding box of the selected person, whether in the foreground or background, the feature vector changes during the time of the collision. This happens because the SlowFast Network extracts action information from both individuals within the bounding box. Consequently, the Boundary Detection module interprets these changes as action transitions, identifying boundaries at the start and end of the collision. Furthermore, any additional action changes performed by the second individual during the collision are also most likely detected and mistakenly assigned to the selected person. To address these incorrect boundary detections, we introduced the collision-robust UnSTABL framework, whose effectiveness will be evaluated in the following section.

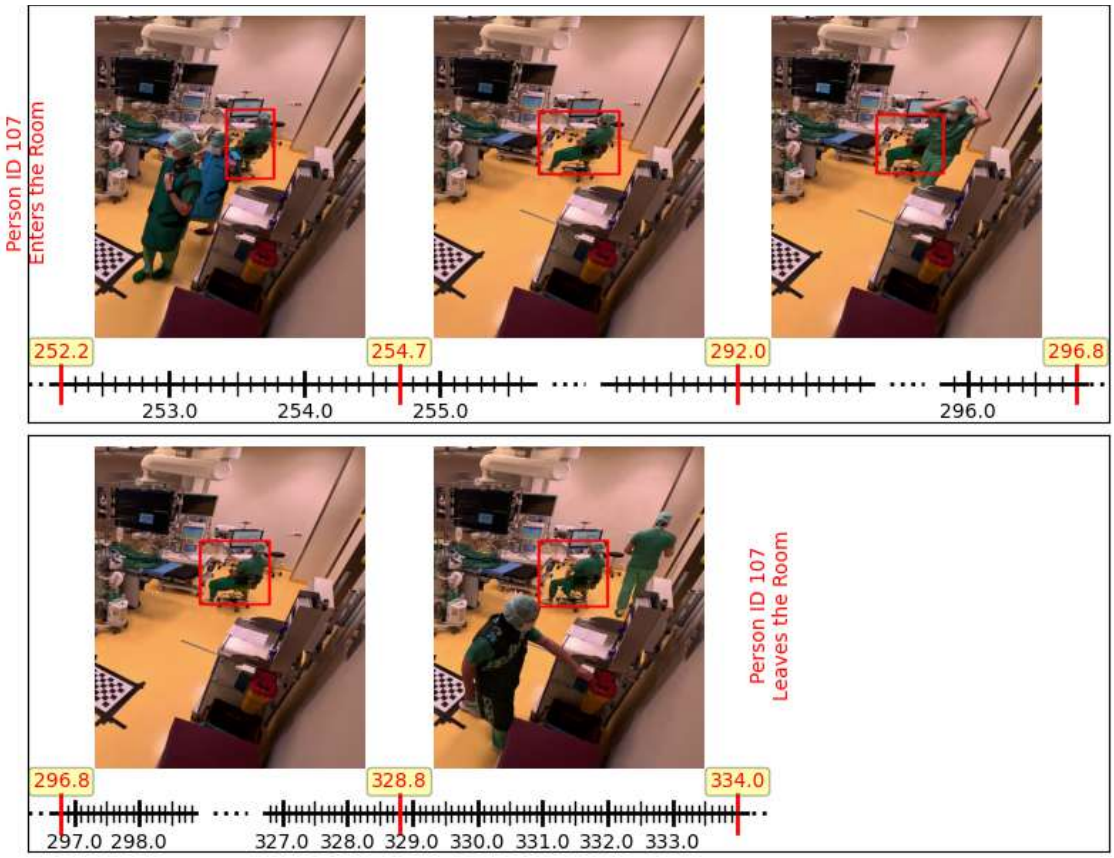

Figure 6.11: **Person Collisions:** A bounding box collision with another individual (whether in the foreground or background) leads to incorrect detection of action boundaries. In this example, the extended "working at the PC" action sequence is split by incorrectly detected boundaries caused by people walking past and overlapping with our subject (Images 1, 3, and 5).

The remaining 20% of incorrectly detected boundaries are typically caused by unexpected hand or body movements from the selected individual. For instance, in Figure 6.12, the person suddenly turns their body to look at something in the background, which our algorithm mistakenly identifies as an action transition. A similar issue arises occasionally

with abrupt hand movements. The unsupervised nature of our framework causes this behavior, as it can only detect significant action changes without any contextual understanding of the actions being performed or which transitions are relevant to the observer. However, since this happens only about 20% of the time, these incorrect detections are negligible in the produced summary.

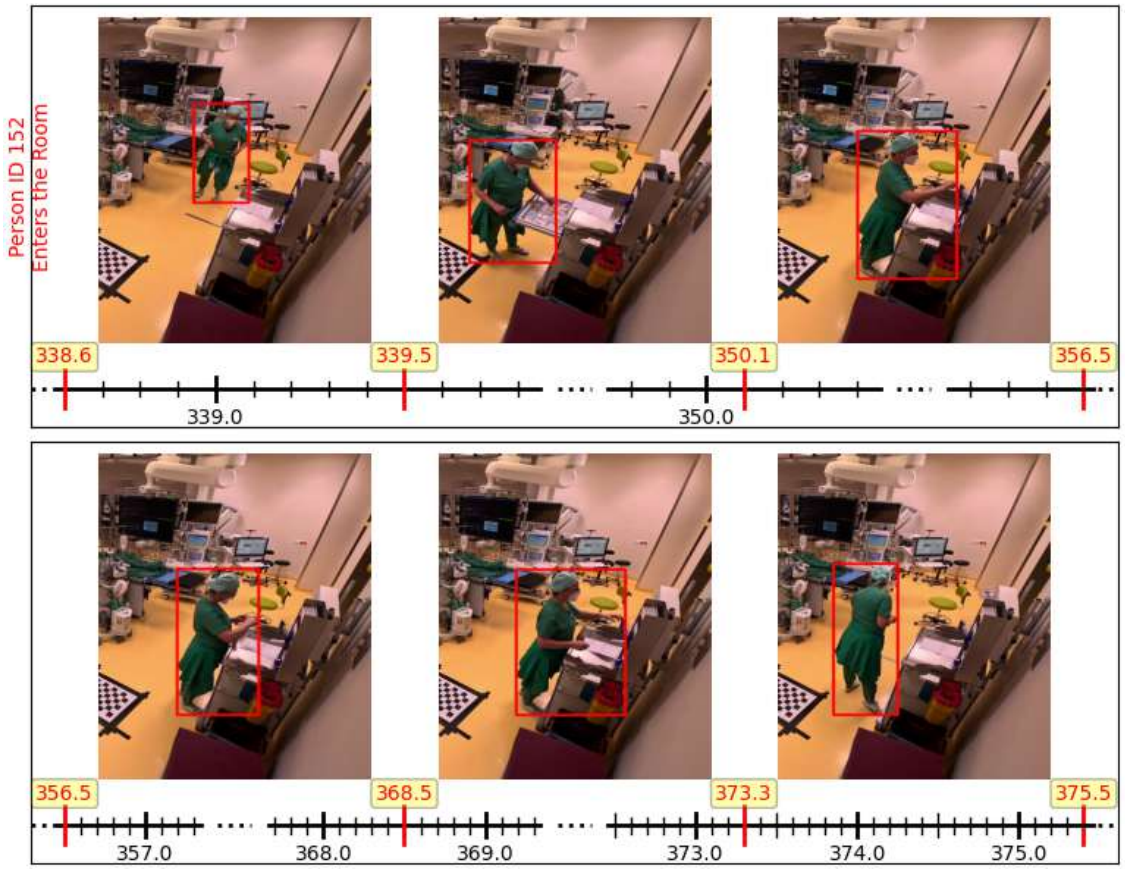

Figure 6.12: **OR Video Summary:** Our framework successfully detects five correct action transitions: (1) walking to the working table, (2) opening the drawer and taking out a syringe, (3) grabbing a needle from the shelf, (4) attaching the needle to the syringe, and (5) placing the syringe down and grabbing a new one. Additionally, one incorrect action segment was detected due to (6) body movement and rotation.

Additionally, we observed a similar issue as during our benchmark tests. Long repetitive action sequences are sometimes divided into sub-actions, as shown in Figure 6.13. As each sub-action is performed repeatedly, they generate numerous feature vectors. These vectors are then grouped into separate clusters due to their large volume, causing the action detection module to identify a transition each time a sub-action changes. In the provided example, our framework splits the long "scanning" sequence into multiple shorter action sequences of "scanning" and "confirming it on the PC", resulting in redundant images and boundaries in the produced summary. However, these sub-action boundaries could still be relevant for other use cases, such as temporal action proposal generation.

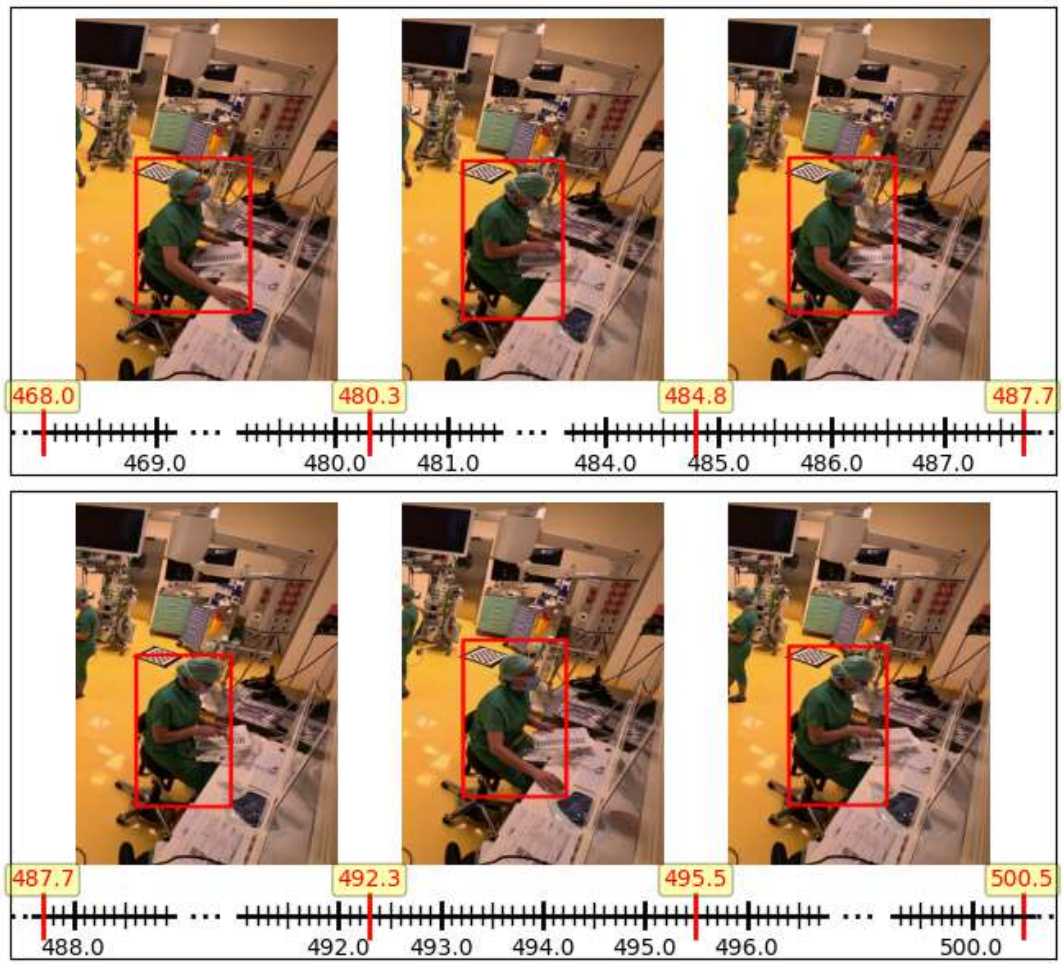

Figure 6.13: **Extended Repetitive Action Sequences:** Long sequences involving repetitive actions are often over-detected. In this example our framework detects each individual action step (1) scanning and (2) confirming it on the PC, splitting the longer "scanning" action sequence into multiple parts.

To briefly summarize, our UnSTABL framework successfully generates high-quality summaries in challenging multi-person environments without missing too many crucial action transitions. Approximately 65% of the detected boundaries were deemed valuable, while 20% included redundant or misinterpreted transitions, mainly due to over-detection of long repetitive action sequences or erratic body and hand movements. To address the remaining 15% of incorrect, collision-induced boundaries, we will evaluate the Collision Avoidance module in the next section.

Collision Avoidance

The Collision Avoidance module identifies strong person collisions and removes feature vectors containing action information of the overlaps. As shown in Table 6.8, this reduces the percentage of incorrectly detected collision induced boundaries from 14.4% to 2.3%, significantly improving the precision of the detected action transitions.

However, these improvements come at a cost. Previously detected action transitions

Table 6.8: **Collision Avoidance Results on our OR Dataset:** Percentages of collision-induced boundaries that remain incorrectly detected using the "collision-robust" UnSTABL framework, along with missed boundaries due to the collision avoidance relative to previously correctly detected ones.

		Standart	**Collision Robust**	
video class	avg. strong collisions	incorr. Boundaries (due to Collision)	incorr. Boundaries (due to Collision)	missed Boundaries (due to Coll. Avoid.)
1	15	16.2%	2.4%	8.4%
2	77	7.8%	1.8%	40.3%
3	120	16.0%	5.4%	26.8%
tot.	80	**14.4%**	**2.3%**	**28.3%**

that occur during overlaps are now more likely to be missed, as the framework eliminates all action information during collisions. As discussed in Section 5.2.2, the framework is capable of detecting action changes immediately after an overlap if the action changes only once. However, in cases of longer overlaps, all boundaries of action sequences that begin and end within the overlap are missed by our collision-robust framework.

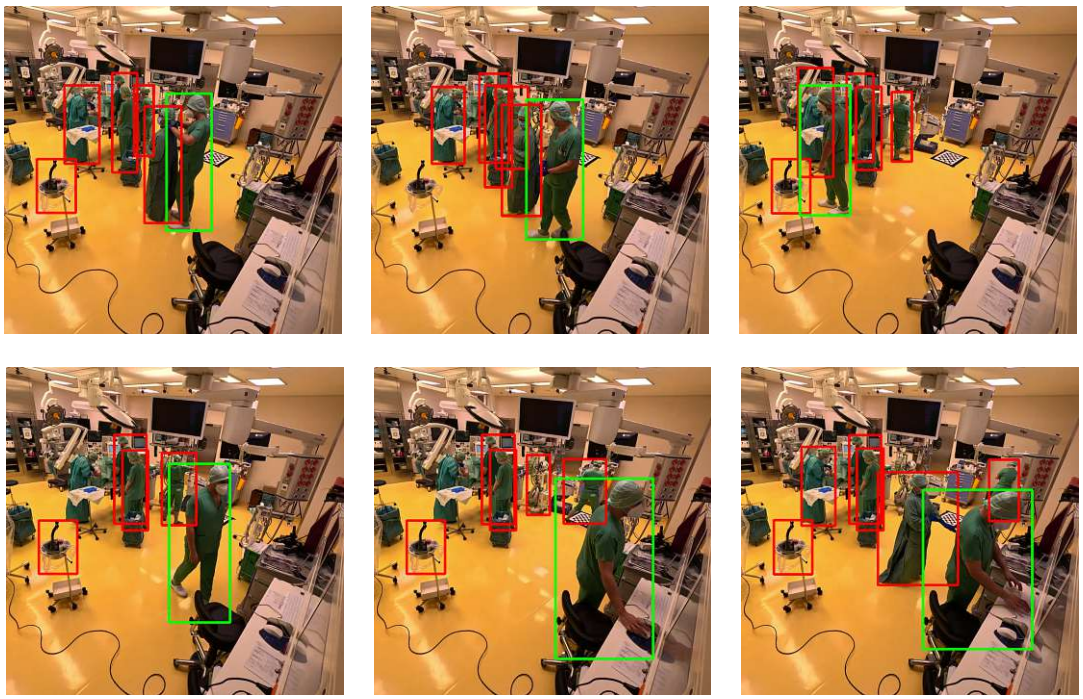

Figure 6.14: **Long Collisions:** During long collision sequences with individuals in the foreground or background (red), the Collison-Robust framework misses numerous action boundaries, such as "put on coat", "walking", etc., of the selected individual (green) due to removing the associated feature vectors.

As highlighted in Table 6.8, this issue is particularly evident in the more crowded scenarios of the video classes (2) "Two-Person Collaboration" and (3) "Crowded Operating Table Setting". In these videos, the individuals exhibit occasionally constant, strong

overlaps, either caused by the collaborating person or by people working in the background, as shown in Figure 6.14. These long overlaps result the loss of numerous action boundaries, with up to 40% missed in the collaboration videos.

In summary, our Collision Robust Boundary Detection approach performs well in scenarios with short, quick overlaps, such as in video class (1) "Single-Person in Multi-Human Environment," where people briefly pass by the selected person. In these cases, the produced summaries show significant improvements, effectively avoiding nearly all collision-induced boundaries while still capturing most of the significant action transitions. However, in scenarios involving frequent or prolonged overlaps, as seen in video classes (2) "Two-Person Collaboration" and (3) "Crowded Operating Table Setting," the advantages of the Collision Robust approach are diminished. The extended overlaps cause too many crucial action transitions to be missed, revealing the limitations of our proposed framework in such demanding environments.

6.2.4 Industrial Dataset

The Industrial video constitutes the fourth and final part of our action boundary detection evaluation. It highlights the adaptability of our UnSTABL framework to entirely different

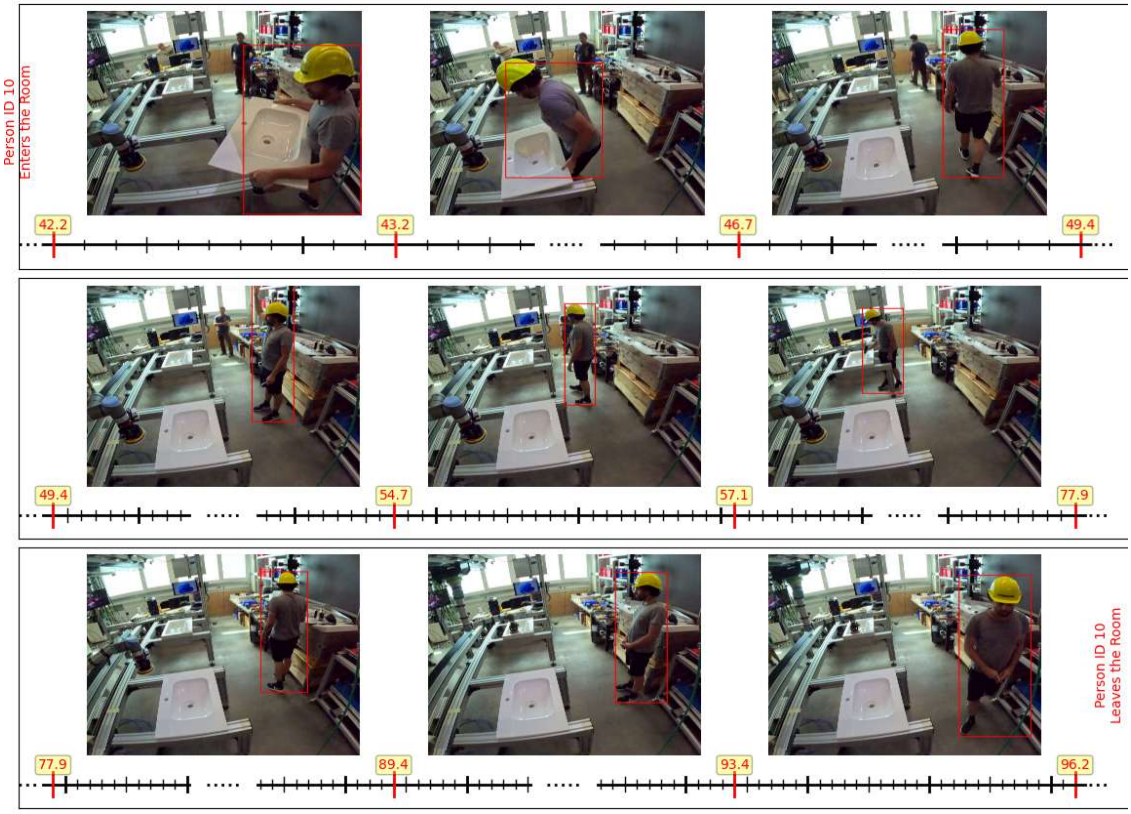

Figure 6.15: **Industrial Video Summary:** Our UnSTABL framework successfully produces a concise summary of almost one minute of video content: The person (1) enters with a sink; (2) places the sink; (3) walks to the middle; (4) raises its hand (activation signal); (5) walks to second sink; (6) draws pattern on sink; (7) walks to the middle; (8) waits in the middle; (9) leaves the room.

environments, made possible by its unsupervised design, as shown in Figure 6.15. The framework only requires surveillance-like videos of individuals to detect action boundaries accurately. It produces similar results across all four datasets without needing to re-train the network. This adaptability over different environments is particularly beneficial in dynamic OR settings, where the room layout can vary significantly from surgery to surgery. Additionally, it enables a broader usage beyond the OR, as shown in this example.

6.3 ID Swap Detection

The final part of the evaluation is again performed on the OR dataset to determine the accuracy of the ID Correction module. At each strong collision or permanent ID disappearance, the module compares the action information of all involved or newly emerged IDs and identifies the most likely permutation. This module aims to improve the tracking performance of the Deep-OC-Sort algorithm in the challenging OR environment.

As a first part, we evaluated the overall performance of our ID Correction module by calculating the percentage of the correctly proposed ID permutations of our selected individual across all checked collisions and ID disappearances. As shown in Table 6.9, in 90% of the cases, the module made the correct decision to either trust the Deep-OC-Sort tracker or propose an ID Swap, demonstrating the high accuracy of our action-based ID swap detection approach.

Table 6.9: **ID Swap Detection Results on OR Dataset:** The tabel provides the percentage of correctly detected, not detected and wrong detected ID Swaps, showing the improvements over the Deep-OC-Sort tracker (total ID Swaps). Additionally it provides the number of checked potential ID Swaps and the percentage of correct decisions, proposing the correct permutation.

		ID Swap Detection			**Likelihood Decision**	
video class	total ID Swaps	detected ID Swaps	not detected ID Swaps	wrong ID Swap det.	checked collisions	correct decision
1	15	53.3%	40.0%	6.7%	54	90.7%
2	5	40.0%	40.0%	20.0%	29	90.0%
3	15	46.7%	20.0%	33.3%	78	90.3%
total	35	**48.6%**	**31.4%**	**20.0%**	**160**	**90.4%**

As a second part, we directly demonstrate the improvements over the Deep-OC-Sort tracker. As shown in Table 6.9, the Deep-OC-Sort tracker assigned a new ID to the selected person 35 times across all videos, either due to losing track or making a wrong re-identification after a strong overlap. By comparing the action information before and after the overlap, the ID Correction module is able to correct nearly 50% of Deep-OC-Sorts wrong ID assignments. This approach proves highly effective for short overlaps, where the actions of the involved individuals are less likely to change, as shown in Figure 6.17. The module identifies the permutation from ID 3 ("sitting") to ID 94 ("sitting") and ID 93 ("walking") to ID 93 ("walking") as the most likely one, successfully detecting the ID swap.

However, as explained in Section 5.3, if the overlap is too long, the actions of the involved individuals are likely to change during the overlap. Consequently, to avoid additional

wrong ID assignments, we check only short, strong overlaps for potential ID swaps.

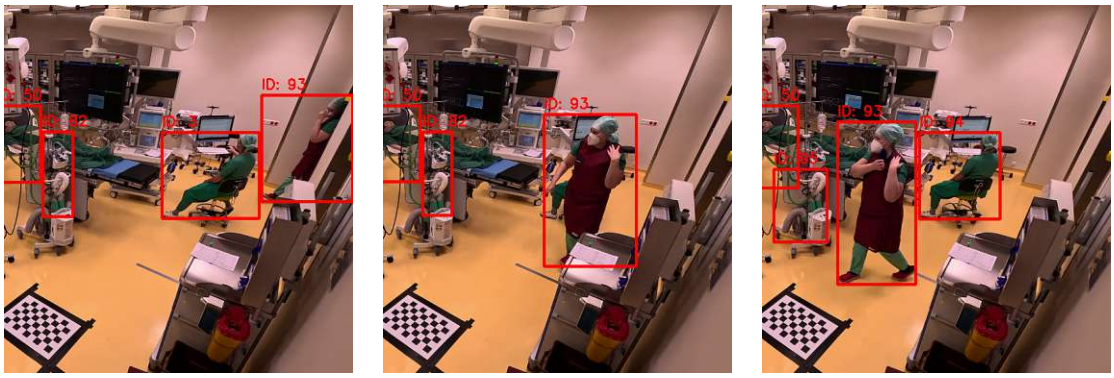

Figure 6.16: **Detected ID Swap:** The ID Swap Detection module is able to identify the ID Swap from 3 to 94 since the action of both individuals remains the same before and after the collision (ID 3,94 "sitting" and ID 93 "walking").

On the other hand, about 30% of the time, our framework is unable to detect the ID Swap, therefore making the same incorrect ID assignments as the Deep-OC-Sort tracker. This mainly occurs when both involved individuals perform the same or similar actions or if they change their actions during the overlap, as illustrated in Figure 6.17. In such cases, the result is either similar permutation scores (since all actions are similar) or scores that are too low (since no actions are similar), making it impossible to identify an ID swap. Under these circumstances, the module wrongfully trusts the original tracking data.

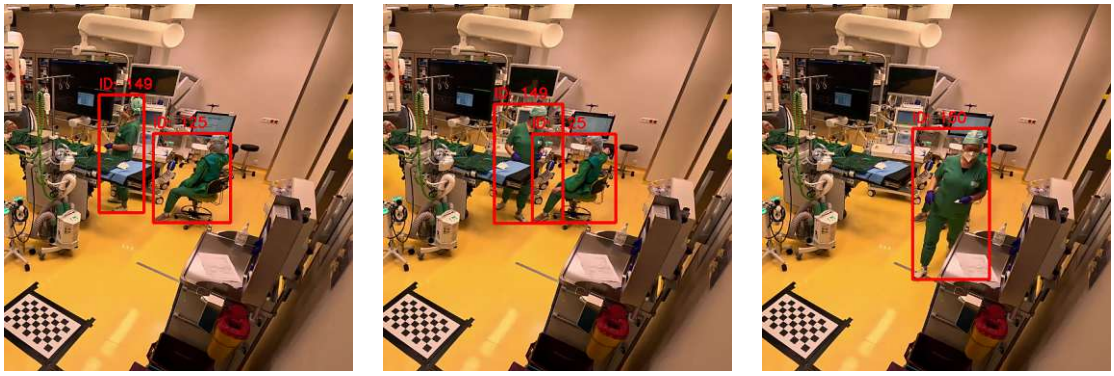

Figure 6.17: **Not Detected ID Swap:** In this case, the ID Swap Detection module is unable to identify the ID swap, because the actions before the collision (ID 149 "standing, watching monitor" and ID 125 "sitting, watching monitor") differ significantly from the action after the collision (ID 150 "walking"), making both permutations unlikely.

The biggest problem are wrong detected ID swaps. About 20% of the time, the ID Correction module suggests wrong ID swaps and mistakenly identifies two different individuals as the same. As shown in Figure 6.18, similar to before, the selected person must change their action during the overlap. However, if the action before the collision is now similar to the second person's action after the collision, this incorrect permutation

becomes the most likely one. As visible in Table 3, this scenario occurs more frequently in densely crowded and collaborative videos, where many individuals work in close proximity and are performing similar actions (e.g., patient preparation), increasing the likelihood of a wrong detected ID swap.

Figure 6.18: **Wrong detected ID Swap:** ID 1 sits before the collision and, when standing up, reveals a new person with ID 6 (also "sitting") and initiates the collision. Consequently, after the collision, the ID Swap Detection module identifies the incorrect permutation of ID 1 to ID 6 as the most likely one, since ID 1 performs a different action "standing, watching monitor".

To briefly summarize this section, the ID Swap detection module significantly improves the tracking performance of the Deep-OC-Sort tracker. Although it introduces a couple of additional ID swaps in crowded settings, its ability to correct around 50% of the incorrect ID assignments across all video classes and achieve a decision accuracy of 90% demonstrates its effectiveness in this challenging OR environment.

7 Conclusion and Outlook

Over recent years, the demand for action detection software has grown significantly as industries rely on automated tools to analyze high amounts of video data. In the OR, for instance, detected action boundaries can be used to create concise summaries for documentation purposes, which can help save time in post-operative reviews and support training by highlighting critical steps. However, industrial and OR environments come with increased complexity due to multiple individuals and numerous concurrent actions, which most state-of-the-art action detection models can not to handle effectively.

To develop an action boundary detection algorithm adaptable to various real-world and multi-person environments without needing extensive annotated datasets or intensive training, we introduced the Unsupervised Spatio-Temporal Action Boundary Localization (UnSTABL) algorithm. UnSTABL automatically identifies action boundaries for each detected individual across spatial and temporal dimensions, enabling precise, person-specific action detection without supervision. To achieve this, our pipeline consists of three stages:

- **Person Detector and Tracker:** We use a pre-trained person detection and tracking model, called Deep-OC-Sort, to identify and track individuals across frames, enabling a person-specific action boundary analysis.
- **Feature Extractor:** Action-specific feature vectors are extracted from each individual using a slightly modified and pre-trained SlowFast network to retrieve action information for the boundary detection stage.
- **Boundary Detector:** We apply a GMM clustering algorithm to group similar action features over time, allowing our framework to identify action changes without supervision.

However, frequent person occlusions and crossovers introduce an additional challenge in multi-person environments, often resulting in wrong-detected action boundaries and ID swaps. To tackle this, we propose a "collision-robust" UnSTABL extension that introduces two additional modules:

- **Collision Avoidance:** This module detects overlaps between individuals and removes the associated feature vectors. This way, we are able to prevent miss-assignments of action information, eliminating collision-induced boundaries.
- **ID Correction:** By comparing action information before and after overlaps, this module is able to identify potential ID swaps, improving the performance of state-of-the-art person trackers in challenging environments.

Extensive benchmark tests revealed that our UnSTABL framework achieves state-of-the-art results in single-person environments. Our approach is able to detect action boundaries

with a high recall, successfully capturing the majority of critical action transitions, which is crucial for accurate summary generation.

In our tests, we evaluated three boundary detection models. The Reduced Feature approach delivered the best performance, combining a high recall with an improved precision over the AIC approach, significantly reducing over-detections. This method is particularly effective at detecting short to moderately long actions, ranging from several seconds to almost a minute. The BIC method on the other hand mainly detects broader action transitions, making it too coarse for generating accurate summaries.

In moderately crowded, multi-person environments, the model delivers similar results to the one observed during the benchmark tests, highlighting the ability of our person-based approach to handle real-world complexities. The collision avoidance module effectively suppresses collision-induced boundaries while still capturing the most critical action transitions in these settings. Furthermore, the ID Correction module is able to identify and correct almost 50% of the tracker's incorrect ID assignments, significantly improving the tracking performance in these challenging settings.

However, densely crowded scenes and collaborative tasks pose a limitation to our approach. The high amount of person overlaps leads to numerous wrong collision-induced boundaries, significantly reducing the quality of the produced summaries. While the collision-robust framework is able to suppress most incorrect boundaries, the long and repeated overlaps cause it to miss too many crucial action transitions.

Despite these limitations, our UnSTABL framework sets a new standard by making unsupervised action boundary detection feasible in moderately crowded multi-person environments without sacrificing accuracy. Additionally, we were able to show that its unsupervised nature makes it adaptable to various real-world environments without any additional effort.

To conclude this thesis, we propose two potential directions for future work. The first proposal aims to improve the action boundary detection in crowded multi-human environments. Similarly to the improvements of Mask R-CNN over Faster R-CNN, focusing on masks instead of bounding boxes could be beneficial. This approach would allow the SlowFast Network to extract action information specific to masked individuals rather than capturing all actions within a bounding box.

The second proposal would be to develop a subsequent stage that combines the detected action boundaries into temporal action proposals and integrates an action recognition algorithm. This would transform the framework into a complete action detection pipeline, capable of not only identifying boundaries but also recognizing specific action instances. As a result, summary generation could be further optimized, allowing summaries to be filtered by specific keywords or action classes to display only the most relevant frames and boundaries.

A Results on the Breakfast Dataset

In this appendix we provide the full results of the evaluation on the Breakfast dataset. The tested subset of the Breakfast dataset includes all videos from the specified directories given in Table A.1, representing a wide range of different action classes and environments.

Table A.1: **Subset of the Breakfast Dataset:** The selected subset for the evaluation consists of 198 videos found in the following Folders. Some videos where however excluded from the Evaluation due to featuring multiple Persons.

Folder	Excluded Videos
P03/cam01	-
P04/cam01	-
P07/stereo	-
P08/cam01	-
P09/cam01	-
P10/cam01	-
P13/stereo	-
P16/cam01	P16_friedegg_ch1
P16/stereo	-
P17/stereo	P17_pancake_ch1
P18/stereo	-
P19/stereo	-
P22/stereo	-
P23/stereo	-
P24/stereo	-
P25/cam01	-
P26/cam01	-
P27/cam01	-
P28/cam01	-
P29/cam01	P29_salat
P30/stereo	-
P32/cam01	P32_friedegg, P32_pancake, P32_scrambledegg
P33/stereo	-
P36/stereo	P36_pancake_ch1
P37/stereo	P37_pancake_ch1
P38/stereo	P38_scrambledegg_ch1
P39/stereo	P39_pancake_ch1
P42/stereo	P42_friedegg_ch1, P42_salat_ch1
P43/stereo	P43_friedegg_ch1, P43_scrambledegg_ch1
P45/webcam01	P45_friedegg, P45_pancake, P45_salat, P45_scrambledegg
P47/webcam01	P47_friedegg, P47_pancake
P53/cam01	-
P54/cam01	-

We selected only videos in which the person was fully visible, ensuring that the Person Detector could accurately identify the individual across all frames, thus providing a reliable evaluation. Videos with partially visible individuals were excluded because missed detections would prevent our UnSTABL framework from accurately identifying action boundaries. Additionally, we removed 20 videos containing multiple persons for most of the duration, as this would result in numerous incorrect detected boundaries since these actions are not labeled in the ground-truth data.

Overall Performance

We start by assessing the overall boundary- and action-level performance of all three Boundary Detection models on the selected subset, given in Table A.2.

Table A.2: **Average Results over all tested videos:** This Table provides the average boundary- and action-based results over all Breakfast videos.

	Precision	Recall	F1	Mean IoU	Prec@0.5	Recall@0.5
AIC	0.356	0.658	0.438	0.481	0.282	0.497
Red. Feat.	0.370	0.606	0.444	0.474	0.299	0.486
BIC	0.446	0.512	0.456	0.488	0.433	0.466

Boundary Based Results

As the second part of our evaluation, we calculated the average boundary-level metrics (Precision, Recall, and F1-Score) across all videos within each video class. The results for each Boundary Detection model are given in Table A.3.

Table A.3: **Video-class based Boundary Results:** This Table provides the average boundary-level metrics per video class. The video classes are (1) tea, (2) coffee, (3) cereals, (4) milk, (5) juice, (6) sandwich, (7) scrambledegg, (8) friedegg, (9) salat, and (10) pancake.

		AIC			**Reduced Features**			**BIC**		
class	avg.len	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
1	612	0.485	0.662	0.549	0.431	0.604	0.494	0.613	0.537	0.564
2	631	0.416	0.643	0.498	0.399	0.649	0.484	0.507	0.621	0.548
3	646	0.478	0.648	0.543	0.502	0.658	0.560	0.554	0.562	0.552
4	917	0.474	0.657	0.540	0.494	0.677	0.562	0.552	0.561	0.537
5	1552	0.314	0.566	0.396	0.321	0.487	0.380	0.379	0.456	0.397
6	1652	0.302	0.641	0.397	0.322	0.628	0.412	0.358	0.512	0.405
7	2757	0.277	0.707	0.385	0.359	0.643	0.449	0.434	0.614	0.477
8	3149	0.179	0.719	0.277	0.227	0.587	0.311	0.243	0.414	0.283
9	3718	0.223	0.694	0.324	0.268	0.527	0.341	0.364	0.377	0.335
10	6518	0.140	0.705	0.228	0.177	0.531	0.251	0.207	0.346	0.245

Action Based Results

For the third part of our evaluation, we computed the average action-level metrics (mean IoU, Precision@0.5, and Recall@0.5) across all video classes. The summarized results of each Boundary Detection model are presented in Table A.4.

Table A.4: **Video-class based Action Results:** This Table provides the average action-level metrics per video class. The video classes are (1) tea, (2) coffee, (3) cereals, (4) milk, (5) juice, (6) sandwich, (7) scrambledegg, (8) friedegg, (9) salat, and (10) pancake.

		AIC			**Reduced Features**			**BIC**		
cl.	avg.l.	IoU	P@0.5	R@0.5	IoU	P@0.5	R@0.5	IoU	P@0.5	R@0.5
1	612	0.52	0.437	0.600	0.50	0.348	0.507	0.49	0.726	0.591
2	631	0.38	0.271	0.355	0.37	0.234	0.331	0.38	0.362	0.369
3	646	0.49	0.357	0.494	0.47	0.378	0.499	0.45	0.486	0.474
4	917	0.54	0.415	0.587	0.55	0.433	0.615	0.47	0.556	0.550
5	1552	0.46	0.270	0.509	0.41	0.289	0.446	0.41	0.403	0.448
6	1652	0.50	0.236	0.500	0.52	0.276	0.532	0.49	0.343	0.535
7	2757	0.48	0.202	0.471	0.52	0.331	0.567	0.51	0.418	0.551
8	3149	0.41	0.109	0.379	0.39	0.155	0.395	0.32	0.189	0.309
9	3718	0.51	0.162	0.527	0.46	0.233	0.482	0.32	0.373	0.372
10	6518	0.47	0.103	0.444	0.46	0.153	0.403	0.28	0.162	0.271

In the fourth part of our evaluation, ground-truth actions were categorized based on their length. For each action length category, we then calculated the average Intersection over Union (IoU) of our detected action proposals relative to the ground-truth actions. This analysis highlights which action (based on their lengths) are best captured by each Boundary Detection model. The results are given in Table A.5.

Table A.5: **Average IoU per Action length**: This Table provides the average Intersection over Union of detected action proposals to ground-truth actions.

Action length	Action count	AIC Average IoU	Red. Features Average IoU	BIC Average IoU
0-25	120	0.086	0.070	0.051
25-50	123	0.370	0.317	0.225
50-100	184	0.593	0.527	0.401
101-300	504	0.632	0.599	0.517
301-500	161	0.534	0.572	0.520
501-1000	109	0.364	0.462	0.478
1001-1500	38	0.251	0.395	0.510
>1500	27	0.131	0.247	0.344

The final two tables A.6 and A.7 show which ground-truth actions are most and least effectively detected across all tested videos. A ground-truth action is considered successfully detected if it achieves an IoU of at least 0.5 with a detected action proposal.

The ground-truth actions are ordered by average lenght to illustrate that shorter actions tend to be detected more accurately than longer, more coarsely defined actions.

Table A.6: **Action-class based Results Part 1:** This Table provides the first part of the percentages of successfully detected ground-truth actions (with an IoU over 0.5).

class	occur.	avg.len	AIC Det.[%]	Red.F.Det.[%]	BIC Det.[%]
walk_in	1	23	0.0	0.0	0.0
take_squeezer	3	65	0.0	66.7	33.3
SIL	396	68	40.9	38.6	30.1
put_bunTogether	12	72	58.3	66.7	33.3
walk_out	20	72	50.0	30.0	55.0
take_knife	8	75	50.0	12.5	37.5
take_butter	1	100	0.0	0.0	0.0
take_cup	22	104	72.7	63.6	68.2
take_bowl	14	105	78.6	50.0	28.6
put_fruit2bowl	51	109	70.6	49.0	21.6
take_glass	12	116	41.7	41.7	25.0
stir_coffee	3	138	66.7	100.0	100.0
stir_cereals	8	138	12.5	25.0	12.5
cut_orange	20	165	70.0	65.0	65.0
pour_juice	28	172	96.4	89.3	92.9
pour_sugar	1	195	0.0	0.0	0.0
add_teabag	27	198	66.7	59.3	74.1
take_plate	31	200	61.3	45.2	41.9
pour_oil	19	204	57.9	57.9	21.1
pour_milk	78	206	66.7	62.8	57.7
stir_milk	22	218	90.9	90.9	68.2
pour_water	27	219	63.0	55.6	81.5
put_egg2plate	29	226	58.6	72.4	62.1
take_topping	8	226	62.5	75.0	62.5
put_pancake2plate	10	227	50.0	60.0	50.0
spoon_sugar	3	241	66.7	100.0	33.3
pour_cereals	24	260	33.3	41.7	62.5
stir_fruit	3	270	100.0	66.7	100.0
add_saltnpepper	23	278	47.8	47.8	60.9

Table A.7: **Action-class based Results Part 2:** This Table provides the second part of the percentages of successfully detected ground-truth actions (with an IoU over 0.5).

class	occur.	avg.len	AIC Det.[%]	Red.F.Det.[%]	BIC Det.[%]
take_eggs	2	284	0.0	0.0	100.0
crack_egg	43	286	53.5	51.2	44.2
spoon_powder	25	295	56.0	52.0	60.0
pour_coffee	20	299	40.0	25.0	60.0
cut_bun	23	303	73.9	78.3	60.9
pour_egg2pan	6	307	50.0	83.3	33.3
spoon_flour	10	357	50.0	40.0	30.0
pour_flour	3	366	33.3	66.7	33.3
put_toppingOnTop	26	371	42.3	53.8	69.2
stir_egg	7	437	71.4	71.4	57.1
pour_dough2pan	11	497	45.5	54.5	18.2
cut_fruit	60	576	40.0	36.7	41.7
butter_pan	15	641	13.3	20.0	33.3
stirfry_egg	20	647	55.0	60.0	80.0
smear_butter	24	652	33.3	37.5	54.2
stir_dough	18	696	66.7	77.8	33.3
squeeze_orange	28	715	25.0	28.6	28.6
peel_fruit	20	752	30.0	80.0	55.0
fry_egg	22	1374	9.1	13.6	13.6
fry_pancake	12	2803	0.0	8.3	33.3

Parameters

This last section provides the parameter of the Person Tracker and the SlowFast Network used during our Evaluation, as detailed in Table A.8.

Table A.8: **Parameters Breakfast Dataset:** This table provides the parameters used during our evaluation.

Deep-OC-Sort	training model	mot17
	det_threshold	0.3
SlowFast	sample_rate	2
	num_frames	32
	alpha	4

B Results on the TSU Dataset

This second appendix provides the complete evaluation results for second benchmark, conducted on the Toyota Smarthome Untrimmed Dataset. We selected 77 videos from the TSU Dataset, representing an approximate 80/20 split. The videos feature all three different environments, different camera angles and persons. A complete list of the selected videos is given in Table B.1.

Table B.1: **Subset of the TSU Dataset:** The selected TSU subset features 77 videos, 31 of those in the kitchen, 31 in the living room, and 15 in the dining room.

Environment	Videos
Kitchen	P02T01C06, P02T01C07, P02T02C03, P02T02C06, P02T02C07 P02T03C03, P02T03C07, P02T10C06, P02T13C06, P02T14C03 P02T16C06
	P03T15C03, P03T15C06, P03T15C07, P03T16C03, P03T16C07 P03T18C03, P03T18C07
	P04T14C03, P04T14C06, P04T15C03, P04T15C06, P04T15C07 P04T16C03, P04T16C06, P04T16C07, P04T17C03, P04T17C06 P04T17C07, P04T18C03, P04T18C06
Living Room	P02T04C05, P02T05C04, P02T06C05, P02T07C04, P02T07C05 P02T08C04, P02T08C05, P02T17C05, P02T18C05
	P03T02C04, P03T02C05, P03T03C04, P03T03C05, P03T04C04 P03T04C05, P03T05C04, P03T06C04, P03T06C05, P03T07C04 P03T07C05, P03T08C04, P03T08C05, P03T19C05
	P04T03C05, P04T04C04, P04T04C05, P04T05C04, P04T06C05 P04T07C05
	P06T02C04, P06T02C05
Dining Room	P02T11C01, P02T12C02
	P03T10C01, P03T10C02, P03T12C02, P03T13C01, P03T13C02
	P04T09C01, P04T09C02, P04T10C01, P04T10C02, P04T11C02
	P04T12C01, P04T13C01, P04T13C02

Overall Performance

As a first part, the Table B.2 provides the overall boundary- and action-based performance of all three Boundary Detection models on the previously defined TSU subset.

Table B.2: **Average Results over all selected TSU Videos:** This Table provides the average boundary- and action-based results over all TSU videos.

	Precision	Recall	F1	Mean IoU	Prec@0.5	Recall@0.5
AIC	0.417	0.652	0.487	0.347	0.158	0.301
Red. Feat.	0.426	0.420	0.402	0.300	0.207	0.240
BIC	0.392	0.274	0.298	0.300	0.218	0.166

Boundary Based Results

As second part of the TSU evaluation, we determined the average boundary-level metrics (Precision, Recall, and F1-Score) per video class. The results of all three Boundary Detection models are given in Table B.3.

Table B.3: **Video-class based Boundary Results:** This Table provides the average boundary-level metrics per video class. The video classes are (1) Dining Room, (2) Kitchen, and (3) Living Room.

		AIC			Reduced Features			BIC		
class	avg.len	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
1	23777	0.27	0.596	0.347	0.312	0.397	0.317	0.286	0.172	0.178
2	15191	0.499	0.708	0.569	0.496	0.486	0.476	0.474	0.381	0.393
3	24319	0.406	0.621	0.472	0.409	0.363	0.367	0.359	0.215	0.259

In addition to the class based results, we determined also the percentage of detected ground-truth boundaries for each action class. However, we only provide the results for the AIC based approach, since it delivers by far the best result on the TSU dataset. The Tables B.4 and B.5 provide the percentage of the AIC based UnSTABL framework detecting both boundaries, only one boundary (start or end) or no boundary for each action class.

Table B.4: **AIC-based Boundary Detection of the Action classes on TSU Part 1:** "Total Det." indicates the percentage of detecting both boundaries of the ground-truth action, "One Det." shows the percentage where either the start or end boundary is detected, and "No Det." represents the percentage where no boundaries are detected.

Action-class	occ.	avg.len	Total Det.[%]	One Det.[%]	No Det.[%]
Take_something_off_table	871	16	69.1	11.9	18.9
Use_glasses	15	19	66.7	13.4	20.0
Put_something_on_table	862	20	59.6	19.1	21.3
Cook.Use_stove	19	26	47.4	31.6	21.1
Get_up	323	26	85.8	11.8	2.5
Sit_down	309	28	79.9	14.8	5.2
Eat_snack	51	34	64.7	25.5	9.8
Drink.From_glass	26	58	57.7	30.7	11.5

Table B.5: **AIC-based Boundary Detection of the Action classes on TSU 2**

Action-class	occ.	avg.len	Total Det.[%]	One Det.[%]	No Det.[%]
Lay_down	34	60	29.4	44.2	26.5
Pour.From_can	4	74	100.0	0.0	0.0
Drink.From_cup	284	78	35.9	38.4	25.7
Drink.From_can	76	81	35.5	40.8	23.7
Clean_dishes.Put_smthing_in_sink	10	81	40.0	40.0	20.0
Dump_in_trash	22	82	68.2	27.3	4.5
Drink.From_bottle	95	93	29.5	35.8	34.7
Stir_coffee/tea	6	98	33.3	16.7	50.0
Pour.From_bottle	45	99	22.2	46.7	31.1
Walk	839	106	70.7	26.4	3.0
Pour.From_kettle	26	120	30.8	46.2	23.1
Get_water	4	122	50.0	50.0	0.0
Use_fridge	41	133	85.4	12.2	2.4
Take_pills	57	144	43.9	42.1	14.0
Breakfast.Cut_bread	11	150	72.7	18.2	9.1
Wipe_table	14	165	64.3	35.7	0.0
Use_cupboard	29	186	75.9	17.2	6.9
Cook.Use_oven	21	188	90.5	4.8	4.8
Make_coffee.Get_water	7	190	85.7	14.3	0.0
Make_coffee.Pour_water	17	192	35.3	23.5	41.2
Make_coffee.Pour_grains	9	216	22.2	55.5	22.2
Use_Drawer	144	217	59.7	24.3	16.0
Make_tea.Boil_water	12	254	91.7	8.3	0.0
Breakfast.Eat_at_table	41	255	39.0	29.3	31.7
Cook.Stir	48	285	64.6	22.9	12.5
Breakfast.Spread_jam_or_butter	6	297	83.3	0.0	16.7
Breakfast.Take_ham	1	313	100.0	0.0	0.0
Make_tea	11	362	63.6	36.4	0.0
Clean_dishes.Clean_with_water	11	363	72.7	27.3	0.0
Clean_dishes.Dry_up	51	518	80.4	19.6	0.0
Cook.Cut	22	586	45.5	36.4	18.2
Use_telephone	76	628	50.0	36.8	13.2
Use_laptop	69	751	39.1	34.8	26.1
Watch_TV	135	797	57.0	31.9	11.1
Clean_dishes	103	968	68.0	25.2	6.8
Make_coffee	17	1042	11.8	29.4	58.8
Use_tablet	56	1492	71.4	25.0	3.6
Write	76	1541	55.3	31.6	13.2
Read	260	1831	58.5	30.8	10.8
Cook	12	4847	75.0	16.6	8.3
Breakfast	4	6012	25.0	50.0	25.0

Action Based Results

As a third part of our TSU evaluation, we provide the action-based metrics for each video class. The results are given in Table B.6.

Table B.6: **Video-class based Action Results:** This Table provides the average action-level metrics per video class. The video classes are (1) Dining Room, (2) Kitchen, and (3) Living Room.

		AIC			Reduced Features			BIC		
cl.	avg.l.	IoU	P@0.5	R@0.5	IoU	P@0.5	R@0.5	IoU	P@0.5	R@0.5
1	23777	0.30	0.080	0.258	0.27	0.109	0.209	0.24	0.145	0.123
2	15191	0.40	0.209	0.360	0.37	0.307	0.317	0.37	0.299	0.243
3	24319	0.31	0.143	0.262	0.25	0.153	0.174	0.26	0.171	0.108

Additionally, we also determined the average IoU per action lenght, measuring how well our three Boundary Detection models capture shorter as well as longer action instances. The results are provided in Table B.7.

Table B.7: **Average IoU per Action length on TSU:** This Table provides the average Intersection over Union of detected action proposals to ground-truth actions.

Action length	Action count	AIC Average IoU	Red. Features Average IoU	BIC Average IoU
0-25	1997	0.174	0.053	0.031
25-50	868	0.351	0.161	0.112
50-100	929	0.491	0.314	0.214
101-300	746	0.522	0.480	0.282
301-500	213	0.453	0.519	0.367
501-1000	221	0.461	0.534	0.462
1001-1500	112	0.399	0.494	0.452
>1500	206	0.252	0.320	0.443

As a last part of our evaluation, we provide show which ground-truth actions are most and least effectively detected across all tested videos using the action-based metrics. A ground-truth action is considered successfully detected if it achieves an IoU of at least 0.5 with a detected action proposal (two consecutive boundaries). The results are given in Table B.8 and B.9.

Table B.8: **Action-class based Results on TSU Part 1**: This Table provides the percentages of successfully detected ground-truth actions (IoU over 0.5).

Action-class	occ.	avg.len	AIC Det.[%]	Red.F Det.[%]	BIC Det.[%]
Take_something_off_table	871	16	9.1	1.6	1.1
Use_glasses	15	19	6.7	6.7	0.0
Put_something_on_table	862	20	9.4	2.2	2.0
Get_up	323	26	18.0	5.6	0.3
Cook.Use_stove	19	26	15.8	0.0	0.0
Sit_down	309	28	29.1	6.1	2.3
Eat_snack	51	34	25.5	3.9	2.0
Drink.From_glass	26	58	50.0	7.7	7.7
Lay_down	34	60	8.8	8.8	2.9
Pour.From_can	4	74	100.0	50.0	75.0
Drink.From_cup	284	78	39.8	18.3	7.4
Drink.From_can	76	81	50.0	18.4	19.7
Clean_dishes.Put_somethg_in_sink	10	81	40.0	0.0	0.0
Dump_in_trash	22	82	59.1	27.3	18.2
Drink.From_bottle	95	93	34.7	25.3	13.7
Stir_coffee/tea	6	98	33.3	0.0	0.0
Pour.From_bottle	45	99	31.1	15.6	11.1
Walk	839	106	51.4	32.9	19.7
Pour.From_kettle	26	120	46.2	19.2	7.7
Get_water	4	122	75.0	50.0	25.0
Use_fridge	41	133	73.2	36.6	22.0
Take_pills	57	144	45.6	36.8	21.1
Breakfast.Cut_bread	11	150	81.8	54.5	36.4
Make_tea.Insert_tea_bag	11	161	63.6	45.5	0.0
Wipe_table	14	165	42.9	64.3	14.3
Use_cupboard	29	186	75.9	62.1	10.3
Cook.Use_oven	21	188	57.1	76.2	42.9
Make_coffee.Get_water	7	190	71.4	85.7	71.4
Make_coffee.Pour_water	17	192	58.8	47.1	41.2
Make_coffee.Pour_grains	9	216	55.6	33.3	33.3
Use_Drawer	144	217	52.1	45.8	24.3
Make_tea.Boil_water	12	254	41.7	41.7	16.7
Breakfast.Eat_at_table	41	255	26.8	19.5	4.9
Cook.Stir	48	285	43.8	39.6	39.6
Breakfast.Spread_jam_or_butter	6	297	50.0	100.0	16.7
Breakfast.Take_ham	1	313	0.0	100.0	0.0

Table B.9: **Action-class based Results on TSU Part 2**

Action-class	occ.	avg.len	AIC Det.[%]	Red.F Det.[%]	BIC Det.[%]
Make_tea	11	362	18.2	45.5	18.2
Clean_dishes.Clean_with_water	11	363	18.2	63.6	36.4
Clean_dishes.Dry_up	51	518	47.1	41.2	31.4
Cook.Cut	22	586	45.5	31.8	27.3
Use_telephone	76	628	34.2	48.7	31.6
Use_laptop	69	751	68.1	76.8	53.6
Watch_TV	135	797	48.1	48.9	36.3
Clean_dishes	103	968	14.6	24.3	28.2
Make_coffee	17	1042	0.0	0.0	11.8
Use_tablet	56	1492	44.6	58.9	58.9
Write	76	1541	36.8	36.8	34.2
Read	260	1831	36.5	39.2	37.7
Cook	12	4847	0.0	8.3	16.7
Breakfast	4	6012	0.0	0.0	50.0

Parameters

In this last section we provide the parameters of the Person Tracker and the SlowFast Network used during our Evaluation, as detailed in Table B.10.

Table B.10: **Parameters TSU Dataset:** This table provides the parameters used during our evaluation.

Deep-OC-Sort	training model	mot17
	det_threshold	0.5
SlowFast	sample_rate	2
	num_frames	32
	alpha	4

C Results on the OR Dataset

This chapter provides a comprehensive overview of the qualitative analysis conducted on our custom OR dataset. We assess the accuracy of detected boundaries by evaluating the generated summaries to identify correct, incorrect, and missed boundaries. Additionally, we analyze the impact of person collisions and the percentage of incorrectly detected boundaries resulting from them.

In the second part, we perform a similar evaluation using the Collision Avoidance module to determine the extent of improvement in boundary detection within these complex environments. Lastly, we assess the performance of the ID Swap Detection module, examining how much it improves the Deep-OC-Sort tracker in challenging conditions by utilizing action information to identify the optimal ID permutations following collisions.

Boundary Detection

As a first part of our evaluation we provide the number and percentage of correctly and incorrectly identified boundaries by manually assessing the produced summaries.

Table C.1: **Full Boundary Results on OR Dataset:** This table provides a qualitative analysis of the number and percentage of correctly and incorrectly identified boundaries. Incorrectly identified boundaries are categorized into two types: (1) redundant or unnecessary boundaries, and (2) incorrect boundaries resulting from collisions.

video	length	strong coll.	det. bounds	correct det. bounds		incorrect det. bounds		incorrect bounds (coll.)	
1	14500	21	46	29	63.04%	5	10.87%	12	26.09%
2	10918	17	63	41	65.08%	10	15.87%	12	19.05%
3	10440	8	52	37	71.15%	13	25.00%	2	3.85%
class 1 total		46	161	107	**66.46%**	28	**17.39%**	26	**16.15%**
7	17300	132	88	42	47.73%	37	42.05%	9	10.23%
8	11383	21	53	25	47.17%	26	49.06%	2	3.77%
class 2 total		153	141	67	**47.52%**	63	**44.68%**	11	**7.80%**
4	17536	280	69	43	62.32%	9	13.04%	17	24.64%
5	21078	57	121	88	72.73%	22	18.18%	11	9.09%
6	11200	154	58	53	91.38%	3	5.17%	2	3.45%
9	11200	23	49	39	79.59%	3	6.12%	7	14.29%
10	16200	89	96	49	51.04%	21	21.88%	26	27.08%
class 3 total		603	393	272	**69.21%**	58	**14.76%**	63	**16.03%**
TOTAL		802	695	446	**64.17%**	149	**21.44%**	100	**14.39%**

Incorrect boundaries are categorized into normal incorrect boundaries (redundant or

wrong) and those caused by person collisions. The results are presented in Table C.2. We also present the percentage of missed boundaries in our summaries relative to the number of correctly identified boundaries in Table C.2. This analysis highlights the reliability of our algorithm in detecting action transitions.

Table C.2: **Full Boundary Results on OR Dataset:** This table presents the number and percentage of important boundaries missed by our framework compared to those correctly detected.

video	length	strong coll.	correct det. bounds	missed bounds	
1	14500	21	29	2	6.90%
2	10918	17	41	1	2.44%
3	10440	8	37	3	8.11%
class 1 total		46	107	6	**5.61%**
7	17300	132	42	0	0.00%
8	11383	21	25	1	4.00%
class 2 total		153	67	1	**1.49%**
4	17536	280	43	0	0.00%
5	21078	57	88	2	2.27%
6	11200	154	53	0	0.00%
9	11200	23	39	3	7.69%
10	16200	89	49	3	6.12%
class 3 total		603	272	8	**2.94%**
TOTAL		802	402	15	**3.73%**

Collision Robust Boundary Detection

As a second part of our evaluation we determine the boundary detection improvement using our Collision Robust approach. Table C.3 presents the percentage of collision-induced boundaries that remain incorrectly detected, offering a clear comparison to the standard UnSTABL approach. In addition it also provides the percentage of previously correct detected boundaries missed due to collision avoidance.

Table C.3: **Full Collision Robust Boundary Results on OR Dataset:** This table shows the percentage of boundaries that remain incorrectly detected due to collisions, even with Collision Robust Boundary Detection, and the percentage of previously correctly detected boundaries missed due to collision avoidance.

					Collision Robust			
		strong	det.	prev. corr.	incorr.		missed	
video	length	coll.	bounds	det. bounds	coll.	bounds	bounds	
1	14500	21	43	29	2	4.65%	1	3.45%
2	10918	17	42	41	1	2.38%	4	9.76%
3	10440	8	42	37	0	0.00%	4	10.81%
class 1 total		46	127	107	3	**2.36%**	9	**8.41%**
7	17300	132	34	42	1	2.94%	17	40.48%
8	11383	21	23	25	0	0.00%	10	40.00%
class 2 total		199	57	67	1	**1.75%**	27	**40.30%**
4	17536	280	27	43	4	14.81%	15	34.88%
5	21078	57	96	88	1	1.04%	12	13.64%
6	11200	154	28	53	0	0.00%	21	39.62%
9	11200	23	35	39	1	2.86%	13	33.33%
10	16200	89	38	49	6	15.79%	12	24.49%
class 3 total		266	224	272	12	**5.36%**	73	**26.84%**
TOTAL		511	695	446	16	**2.30%**	126	**28.3%**

ID Swap Detection

The ID Swap Detection module checks each shorter strong collision for potential ID swaps and determines the most likely ID permutation following each collision. This section presents the percentage of detected ID swaps, highlighting the improvements over the Deep-OC-Sort Tracker. We also report the percentage of undetected action transitions, making the same errors as Deep-OC-Sort, and the percentage of incorrectly introduced ID swaps after collisions. These results are provided in Table C.4.

Additionally, Table C.5 provides the percentage of correct and incorrect ID swap decisions for each evaluated collision, demonstrating the accuracy of our ID Swap Detection module in correctly identifying ID permutations post-collision.

Table C.4: **Full ID Swap Detection Results on OR Dataset:** This table presents the results of ID Swap Detection across all videos and video classes. It details the percentages of correctly identified ID swaps, undetected ID swaps, and incorrectly detected ID swaps (where the algorithm assigns an incorrect ID after a collision).

					ID Swap Detection					
video	length	strong coll.	checked coll.	tot. ID Swaps	detected ID Swaps		not detected ID Swaps		wrong ID Swap det.	
1	14500	21	23	4	2	50.0%	2	50.0%	0	0.0%
2	10918	25	11	5	2	40.0%	3	60.0%	0	0.0%
3	10440	21	20	6	4	66.7%	1	16.7%	1	16.7%
class 1 total		67	54	15	8	**53.3%**	6	**40.0%**	1	**6.7%**
7	17300	123	12	4	2	50.0%	2	50.0%	0	0.0%
8	11383	44	18	1	0	0.0%	0	0.0%	1	100.0%
class 2 total		167	30	5	2	**40.0%**	2	**40.0%**	1	**20.0%**
4	17536	280	24	4	2	50.0%	1	25.0%	1	25.0%
5	21078	57	6	3	2	66.7%	1	33.3%	0	0.0%
6	11200	154	12	3	1	33.3%	1	33.3%	1	33.3%
9	11200	23	8	4	1	25.0%	0	0.0%	3	75.0%
10	16200	89	33	1	1	100.0%	0	0.0%	0	0.0%
class 3 total		603	83	15	7	**46.7%**	3	**20.0%**	5	**33.3%**
TOTAL		837	167	35	17	**48.6%**	11	**31.4%**	7	**20.0%**

Table C.5: **Full ID Swap Likelihood Decision Results on OR Dataset:** This table presents the decision performance of our ID Swap Detection module, providing the percentage of correct and incorrect decisions per checked collision. A wrong decision is either to propose a wrong or to not detect an ID Swap.

				Likelihood Decision			
		strong	checked	correct		wrong	
video	length	collisions	collisions	decision		decision	
1	14500	21	23	21	91.30%	2	8.70%
2	10918	25	11	10	90.91%	1	9.09%
3	10440	21	20	18	90.00%	2	10.00%
class 1 total		67	54	49	90.74%	5	9.26%
7	17300	123	12	10	83.33%	2	16.67%
8	11383	44	18	17	94.44%	1	5.56%
class 2 total		167	30	27	90.00%	3	10.00%
4	17536	280	24	22	91.67%	2	8.33%
5	21078	57	6	5	83.33%	1	16.67%
6	11200	154	12	10	83.33%	2	16.67%
9	11200	23	8	5	62.50%	3	37.50%
10	16200	89	33	33	100.00%	0	0.00%
class 3 total		603	83	75	90.36%	8	9.64%
TOTAL		837	167	151	90.42%	16	9.58%

Parameters

This last section, in Table C.6, we provide the parameter used during our Evaluation.

Table C.6: **Parameters OR Dataset:** This table provides the parameters used during our evaluation.

Detector	training model	dance
	det_threshold	0.7
SlowFast	sample_rate	2
	num_frames	32
	alpha	4
ID Switch	min_sim_threshold	0.6
	min_sim_difference	0.05
	weight_curr_id	0.2
	weight_other_id	0.1
	max_overlap_duration	400
	weak_overlap_th	0.05
	strong_overlap_th	0.11
Collision Avoidance	weak_overlap_th	0.05
	strong_overlap_th	0.35

Bibliography

[1] D. G. Lowe, „Object recognition from local scale-invariant features,“ in *Proceedings of the seventh IEEE international conference on computer vision*, Ieee, vol. 2, 1999, pp. 1150–1157.

[2] N. Dalal and B. Triggs, „Histograms of oriented gradients for human detection,“ in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, 886–893 vol. 1.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, 2014. arXiv: 1311.2524 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1311.2524.

[4] R. Girshick, *Fast r-cnn*, 2015. arXiv: 1504.08083 [cs.CV].

[5] S. Ren, K. He, R. Girshick, and J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, Issue: arXiv:1506.01497 arXiv:1506.01497 [cs], Jan. 2016. [Online]. Available: http://arxiv.org/abs/1506.01497 (visited on 11/29/2022).

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, 2016. arXiv: 1506.02640 [cs.CV].

[7] B. Ghanem, J. C. Niebles, C. Snoek, *et al.*, *Activitynet challenge 2017 summary*, 2017. arXiv: 1710.08011 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1710.08011.

[8] K. O'Shea and R. Nash, *An introduction to convolutional neural networks*, 2015. arXiv: 1511.08458 [cs.NE].

[9] S. Hochreiter and J. Schmidhuber, „Long short-term memory,“ *Neural computation*, vol. 9, pp. 1735–80, Dec. 1997.

[10] C. R. Wolfe. „Deep learning on video (part two): The rise of two-stream architectures.“ (2022), [Online]. Available: https://towardsdatascience.com/deep-learning-on-video-part-two-the-rise-of-two-stream-architectures-f830d5c655d0 (visited on 08/21/2024).

[11] C. Feichtenhofer, H. Fan, J. Malik, and K. He, *Slowfast networks for video recognition*, 2019. arXiv: 1812.03982 [cs.CV].

[12] R. Dai, „Action detection for untrimmed videos based on deep neural networks,“ Theses, Université Côte d'Azur, Sep. 2022. [Online]. Available: https://theses.hal.science/tel-03827178.

[13] G. Yao, T. Lei, X. Liu, and P. Jiang, „Temporal action detection in untrimmed videos from fine to coarse granularity,“ *Applied Sciences*, vol. 8, no. 10, 2018, ISSN: 2076-3417. [Online]. Available: https://www.mdpi.com/2076-3417/8/10/1924.

[14] S. Wold, K. Esbensen, and P. Geladi, „Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1, pp. 37–52, 1987, Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists, ISSN: 0169-7439. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0169743987800849.

[15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, *Generative adversarial networks*, 2014. arXiv: 1406.2661 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1406.2661.

[16] B. Wang, Y. Zhao, L. Yang, T. Long, and X. Li, „Temporal action localization in the deep learning era: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 4, pp. 2171–2190, 2024.

[17] Y. Li, L. Chen, R. He, Z. Wang, G. Wu, and L. Wang, *Multisports: A multi-person video dataset of spatio-temporally localized sports actions*, 2021. arXiv: 2105.07404 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2105.07404.

[18] X. Li, B. Zhao, and X. Lu, „Key frame extraction in the summary space," *IEEE Transactions on Cybernetics*, vol. 48, no. 6, pp. 1923–1934, 2018.

[19] J. Redmon and A. Farhadi, *Yolov3: An incremental improvement*, 2018. arXiv: 1804.02767 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1804.02767.

[20] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, *Yolox: Exceeding yolo series in 2021*, 2021. arXiv: 2107.08430 [cs.CV].

[21] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, *Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*, 2022. arXiv: 2207.02696 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2207.02696.

[22] D. Reis, J. Kupec, J. Hong, and A. Daoudi, *Real-time flying object detection with yolov8*, 2024. arXiv: 2305.09972 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2305.09972.

[23] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, *Microsoft coco: Common objects in context*, 2015. arXiv: 1405.0312 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1405.0312.

[24] G. Maggiolino, A. Ahmad, J. Cao, and K. Kitani, *Deep oc-sort: Multi-pedestrian tracking by adaptive re-identification*, 2023. arXiv: 2302.11813 [cs.CV].

[25] Y. Zhang, P. Sun, Y. Jiang, *et al.*, *Bytetrack: Multi-object tracking by associating every detection box*, 2022. arXiv: 2110.06864 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2110.06864.

[26] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, *Observation-centric sort: Rethinking sort for robust multi-object tracking*, 2023. arXiv: 2203.14360 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2203.14360.

[27] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, „Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, Sep. 2016. [Online]. Available: http://dx.doi.org/10.1109/ICIP.2016.7533003.

[28] N. Wojke, A. Bewley, and D. Paulus, *Simple online and realtime tracking with a deep association metric*, 2017. arXiv: 1703.07402 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1703.07402.

[29] P. Dendorfer, A. Ošep, A. Milan, *et al.*, *Motchallenge: A benchmark for single-camera multiple target tracking*, 2020. arXiv: 2010.07548 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2010.07548.

[30] P. Dendorfer, H. Rezatofighi, A. Milan, *et al.*, *Mot20: A benchmark for multi object tracking in crowded scenes*, 2020. arXiv: 2003.09003 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2003.09003.

[31] P. Sun, J. Cao, Y. Jiang, *et al.*, *Dancetrack: Multi-object tracking in uniform appearance and diverse motion*, 2022. arXiv: 2111.14690 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2111.14690.

[32] C. Feichtenhofer, A. Pinz, and A. Zisserman, *Convolutional two-stream network fusion for video action recognition*, 2016. arXiv: 1604.06573 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1604.06573.

[33] K. Simonyan and A. Zisserman, *Two-stream convolutional networks for action recognition in videos*, 2014. arXiv: 1406.2199 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1406.2199.

[34] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, *Learning spatiotemporal features with 3d convolutional networks*, 2015. arXiv: 1412.0767 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1412.0767.

[35] J. Carreira and A. Zisserman, *Quo vadis, action recognition? a new model and the kinetics dataset*, 2018. arXiv: 1705.07750 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1705.07750.

[36] K. Hara, H. Kataoka, and Y. Satoh, „Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?“ In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6546–6555.

[37] W. Kay, J. Carreira, K. Simonyan, *et al.*, *The kinetics human action video dataset*, 2017. arXiv: 1705.06950 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1705.06950.

[38] W. Kay, J. Carreira, K. Simonyan, *et al.*, *The kinetics human action video dataset*, 2017. arXiv: 1705.06950 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1705.06950.

[39] C. Gu, C. Sun, D. A. Ross, *et al.*, *Ava: A video dataset of spatio-temporally localized atomic visual actions*, 2018. arXiv: 1705.08421 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1705.08421.

[40] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. de Mathelin, and N. Padoy, *Endonet: A deep architecture for recognition tasks on laparoscopic videos*, 2016. arXiv: 1602.03012 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1602.03012.

[41] A. Sharghi, H. Haugerud, D. Oh, and O. Mohareri, *Automatic operating room surgical activity recognition for robot-assisted surgery*, 2020. arXiv: 2006.16166 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2006.16166.

[42] A. P. Twinanda, „Vision-based approaches for surgical activity recognition using laparoscopic and rgbd videos,“ English, ⟨NNT : 2017STRAD005⟩. ⟨tel-01557522⟩, PhD thesis, Université de Strasbourg, 2017.

[43] Z. Du, X. Wang, G. Zhou, and Q. Wang, „Fast and unsupervised action boundary detection for action segmentation,“ in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 3313–3322.

[44] H. Kuehne, A. B. Arslan, and T. Serre, „The language of actions: Recovering the syntax and semantics of goal-directed human activities,“ in *Proceedings of Computer Vision and Pattern Recognition Conference (CVPR)*, 2014.

[45] Y. Li, Z. Xue, and H. Xu, *Otas: Unsupervised boundary detection for object-centric temporal action segmentation*, 2023. arXiv: 2309.06276 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2309.06276.

[46] T. Lin, X. Zhao, and Z. Shou, *Temporal convolution based action proposal: Submission to activitynet 2017*, 2018. arXiv: 1707.06750 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1707.06750.

[47] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, *Bsn: Boundary sensitive network for temporal action proposal generation*, 2018. arXiv: 1806.02964 [cs.CV].

[48] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, *Bmn: Boundary-matching network for temporal action proposal generation*, 2019. arXiv: 1907.09702 [cs.CV].

[49] H. Su, W. Gan, W. Wu, Y. Qiao, and J. Yan, *Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation*, 2021. arXiv: 2009.07641 [cs.CV].

[50] R. Dai, S. Das, S. Sharma, *et al.*, *Toyota smarthome untrimmed: Real-world untrimmed videos for activity detection*, 2020. arXiv: 2010.14982 [cs.CV].

[51] M. Basavarajaiah and P. Sharma, „Gvsum: Generic video summarization using deep visual features,“ *Multimedia Tools and Applications*, vol. 80, no. 9, pp. 14 459–14 476, 2021. [Online]. Available: https://doi.org/10.1007/s11042-020-10460-0.

[52] A. Singh Parihar, J. Pal, and I. Sharma, „Multiview video summarization using video partitioning and clustering,“ *Journal of Visual Communication and Image Representation*, vol. 74, p. 102 991, 2021, ISSN: 1047-3203. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320320302091.

[53] H. Fan, Y. Li, B. Xiong, W.-Y. Lo, and C. Feichtenhofer, *Pyslowfast*, https://github.com/facebookresearch/slowfast, 2020.

[54] D. Peng, Z. Gui, and H. Wu, *Interpreting the curse of dimensionality from distance concentration and manifold effect*, 2024. arXiv: 2401.00422 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2401.00422.

[55] A. Mueller, *Clustering with mixture models*, Accessed: 2024-09-05, 2023. [Online]. Available: https://amueller.github.io/aml/03-unsupervised-learning/02-clustering-mixture-models.html.

[56] L. He, X. Liao, W. Liu, X. Liu, P. Cheng, and T. Mei, *Fastreid: A pytorch toolbox for general instance re-identification*, 2020. arXiv: 2006.02631 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2006.02631.

[57] A. Sharghi, H. Haugerud, D. Oh, and O. Mohareri, „Automatic operating room surgical activity recognition for robot-assisted surgery," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, A. L. Martel, P. Abolmaesumi, D. Stoyanov, *et al.*, Eds., Cham: Springer International Publishing, 2020, pp. 385–395.

[58] A. P. Twinanda, E. O. Alkan, A. Gangi, M. de Mathelin, and N. Padoy, „Data-driven spatio-temporal rgbd feature encoding for action recognition in operating rooms," *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 6, pp. 737–747, 2015, [published correction appears in Int J Comput Assist Radiol Surg. 2015 Jul;10(7):1177. doi: 10.1007/s11548-015-1227-9].

Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt wurde. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.
Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, im September 2024

Stefan Kuen