Rilevamento in tempo reale delle azioni violente negli autobus urbani

ELECTRICAL AND INFORMATION ENGINEERING

Ph.D. Program

SSD: ING/INF-04 - Intelligent Transportation

Final Dissertation

Real-Time Violent Action Detection in Urban Buses Environment

---

by

Marco Gallo

Supervisor

Prof. David Naso

Prof. Paolo R. Massenio

Coordinator of Ph.D Program:
Prof. Nicola Giaquinto

XXXVIII Cycle - November 1st, 2022 - October 30st, 2025

ELECTRICAL AND INFORMATION ENGINEERING

Ph.D. Program

SSD: ING/INF-04 - Intelligent Transportation

Final Dissertation

Real-Time Violent Action Detection in Urban Buses Environment

---

by

Marco Gallo

Supervisors:

Prof. David Naso

Prof. Paolo R. Massenio

Coordinator of Ph.D Program:

Prof. Nicola Giaquinto

XXXVIII Cycle - November 1st, 2022 - October 30st, 2025

"Se torturi i dati abbastanza a lungo, finiranno per confessare qualsiasi cosa." — Ronald H. Coase

Abstract

Ensuring passenger safety is a critical requirement for sustainable mobility. Yet, despite the social relevance of this objective, the automatic detection of violent behaviors in public transportation has received limited attention compared with general-purpose surveillance. Most existing approaches are designed around benchmark settings and cloud-scale resources, while the deployment on board of vehicles—in constrained, latency-sensitive, and privacy-critical environments—remains underexplored. This dissertation addresses this gap by presenting, validating, and discussing a complete AI-based system for real-time violence detection specifically tailored to buses.

From a system-design perspective, public transport imposes a unique combination of constraints. Sensing is affected by overhead viewpoints, frequent occlusions, abrupt motion, changes in illumination across stops and routes, and the highly dynamic arrangement of passengers. Computing must be performed at the edge to avoid dependency on unreliable or costly connectivity and to meet stringent latency and privacy requirements. Finally, alarms must be generated with a balanced trade-off between sensitivity (to detect true incidents) and specificity (to limit nuisance alarms that could desensitize operators). The proposed solution is engineered around these constraints. It integrates six ceiling-mounted IP cameras with an embedded GPU server placed on the vehicle to enable on-board, edge-only inference. The architecture supports multi-camera ingestion, synchronized analysis, and a lightweight decision layer that fuses the evidence produced by the video models into actionable alerts. The end-to-end pipeline is designed to remain operational without cloud services, with all computation restricted to the vehicle and only high-level alerts exposed to external systems when available.

A key scientific challenge is domain shift. Models trained on standard violence datasets often fail to generalize to the particular visual and behavioral patterns observed on buses. To mitigate this, we constructed a composite dataset specifically oriented to public-transport scenarios. It combines established public resources—RWF-2000, UCF-Crime, SCVD, and Bus Violence—with a proprietary dataset recorded in a full-scale bus simulator reproducing real layouts, camera geometry, and crowding conditions. The simulator enables controlled acquisition of edge cases (e.g., rapid crowd movements, partial occlusions, seated interactions)

that are underrepresented in generic datasets. This composite corpus is used both to pre-adapt models and to quantify the benefit of injecting bus-specific samples into training.

On the algorithmic side, we investigate three state-of-the-art video architectures with complementary inductive biases: X3D, R(2+1)D, and SlowFast-50. All networks are initialized from Kinetics-400 to leverage large-scale motion representations and then adapted with a progressive unfreezing protocol. In this strategy, learning starts by training the classification head while early spatiotemporal blocks are kept frozen, and progressively deeper blocks are unfrozen as training stabilizes. This schedule supports stable transfer to the bus domain, avoids catastrophic forgetting, and reduces overfitting when the amount of domain-specific data is limited. We complement the fine-tuning with data treatments that reflect in-vehicle conditions, such as temporal subsampling, moderate motion blur, and compression artifacts consistent with IP camera streams, aiming to narrow the sim-to-real gap without adding computational burden at inference.

System performance is assessed through a multi-stage evaluation. First, ablation studies analyze the impact of each design choice: (i) inclusion of proprietary bus data in training, (ii) choice of backbone architecture, and (iii) the progressive unfreezing schedule. Second, we validate the system in the bus simulator to stress-test detection under controlled yet realistic variations—lighting, occupancy, and camera viewpoints—while measuring latency end to end. Third, we conduct supervised field trials on an actual 13-meter bus to evaluate the full pipeline under operational conditions (vibration, network jitter, passenger flow). This layered methodology allows us to disentangle model-centric effects from system-level factors and to quantify the reliability of the deployed solution.

Results consistently indicate that supplementing public benchmarks with bus-specific proprietary data markedly improves generalization to in-vehicle scenes. Among the tested architectures, X3D delivers the best trade-off between accuracy and efficiency, sustaining real-time analysis with sub-second end-to-end latency on the embedded GPU server while maintaining competitive detection quality. R(2+1)D and SlowFast-50 remain valuable references—particularly in scenarios with abundant compute or when higher temporal fidelity is desirable—but X3D proves more suitable for continuous, on-board operation. Importantly, these findings hold across ablations and are confirmed in the simulator and during on-vehicle trials, suggesting that the proposed training protocol and system integration are robust to deployment variations.

Beyond raw metrics, the study highlights several operational insights. First, multi-camera coverage from ceiling viewpoints is critical to mitigate occlusions and to capture interactions between standing and seated passengers; simple late fusion at the decision layer can already provide meaningful resilience without the cost of multi-view feature fusion. Second, prioritizing determinism in the video pipeline (fixed frame rates, bounded buffering, watchdogs) reduces tail latencies that could

delay alarms during critical events. Third, edge-only processing not only satisfies privacy constraints by avoiding video streaming off the vehicle but also enhances availability: the system remains functional in areas with poor connectivity and is immune to cloud outages. Finally, supervised validation in real service uncovers failure modes rarely observable in benchmarks—e.g., aggressive gestures partially hidden by grab poles, or non-violent yet energetic events such as joyful celebrations—that guide subsequent data curation.

The main contributions of this dissertation are as follows: (1) a complete design and deployment of an edge-based, multi-camera violence detection system dedicated to buses, covering sensing, embedded inference, and decision layers; (2) a composite dataset strategy that blends established benchmarks with simulator-acquired, bus-specific samples to address domain shift; (3) a principled fine-tuning pipeline based on progressive unfreezing for efficient transfer from Kinetics-400 to the bus domain; and (4) a comprehensive evaluation protocol spanning ablations, simulator validation, and real-world field trials on a 13-meter bus. To the best of our knowledge, this is among the first works to go beyond simulation and isolated benchmark testing by demonstrating a fully deployed, real-time, edge-only system for violence detection in public transport.

While the achieved performance and latency meet the operational targets of the target platform, the study also surfaces limitations that motivate future work. The rarity and diversity of violent incidents constrain data scale and label granularity; additional targeted collection, continual learning in-the-loop, and stronger out-of-distribution detection could further reduce false positives. Multi-camera reasoning is handled at the decision level; exploring mid-level or feature-space fusion could capture inter-view dynamics more effectively when compute permits. Finally, broader ethical and legal considerations remain central to large-scale adoption: transparent governance, privacy-by-design data handling, and human-in-the-loop review should be embedded into any real-world deployment pipeline.

In summary, this dissertation demonstrates that accurate, low-latency violence detection for public buses is feasible on embedded hardware through the joint design of a domain-aware dataset, an efficient fine-tuning strategy, and an edge-first system architecture. The results provide empirical evidence that tailoring both learning and engineering choices to the constraints and phenomenology of in-vehicle environments is essential to bridge the gap between promising laboratory performance and dependable operation in the field.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Violence Detection in Public Transportation: Context and Motivation

Public transportation is a cornerstone of contemporary urban life. Buses, in particular, remain among the most ubiquitous and adaptable modes of mobility, connecting peripheral neighbourhoods with central districts and enabling affordable access at scale. Yet, the same structural conditions that make bus services socially valuable—high passenger throughput, dense occupancy at peak hours, and a physically constrained interior—also increase the likelihood that interpersonal tensions escalate into aggression or violence [11, 50]. The security problem is multidimensional and spans passenger-to-passenger altercations, passenger-to-driver assaults, vandalism and property damage, theft and pick-pocketing, and broader disorder that hinders operations and erodes public confidence in transit services [11]. When incidents occur within the compact geometry of bus aisles, the consequences can be amplified by crowding, limited escape routes, and delayed intervention—especially when the driver operates alone or with restricted physical separation from passengers [11, 84].

A salient shift in the recent technical literature is the explicit framing of *real-time* violence detection as an operationally critical objective rather than a mere benchmark exercise. In a comprehensive and up-to-date review focused on physical assault, more recent survey works emphasize that real-time identification constitutes the most immediate line of defense, and they highlight the breadth of unresolved challenges and design choices that still fragment the field [49]. In parallel, efficiency-oriented contributions have argued that the predominant focus on accuracy alone is insufficient for deployment, and have proposed architectures explicitly designed to operate with low latency and modest compute budgets [35]. These positions align closely with the public-transportation setting, where the value of automation is proportional to the *time-to-alert* and to the stability of decisions

under noise, occlusion, and non-stationary backgrounds.

Historically, transportation agencies have relied on closed-circuit television (CCTV) systems to deter misconduct and support post hoc investigations. Cameras typically cover the front door, the driver's area, the aisle, and the rear door, offering a structured but partial view of the cabin. While retrospective analysis can be valuable, operational effectiveness is bounded by human attention: monitoring performance degrades under sustained load, and a substantial fraction of unfolding events can be missed, especially under multitasking conditions [41, 63]. Meta-analyses of CCTV effectiveness in public spaces show heterogeneous outcomes, strongly mediated by local context, system design, and the human monitoring practices that govern attention and response [80, 81, 27]. In vehicles, these limitations are exacerbated by motion-induced blur, rapid illumination changes (e.g., tunnels, shaded streets), and frequent occlusions among standing passengers [12].

Modern violence detection, however, cannot be reduced to clip-level binary classification alone. Inside buses, an operationally useful system often must answer *where* and *when* aggression occurs (and, implicitly, *who* is involved), because actionable alerts require localization for driver/dispatcher triage and for efficient retrieval of short evidence snippets. This motivates the conceptual proximity between onboard violence detection and *spatio-temporal action detection* (STAD), which jointly addresses action class recognition, temporal boundaries, and actor localization [77]. In this framing, the objective is not merely "violence vs. no violence" over a window, but the inference of an action tube—or a localized spatio-temporal hypothesis—robust to occlusion, scale variation along the aisle, and partial views.

From a criminological perspective, buses represent mobile micro-environments examined through routine activity theory and situational crime prevention: suitable targets, motivated offenders, and attenuated guardianship can align during high-density periods [15, 14]. Unlike fixed-location transit nodes, however, buses exhibit rapidly varying passenger composition, social norms, and guardianship across space and time. This dynamism complicates static deployment strategies and reinforces the potential value of on-vehicle automation that continuously adapts to changing conditions [11, 50].

Empirically, agencies and labor unions have reported persistent concern about assaults on drivers and anti-social behaviour in passenger compartments, with downstream impacts on staff retention, sick leave, and perceived safety that deters ridership [84, 86]. Effective systems must therefore balance rapid detection and low false alarms with privacy, proportionality, and transparency. In the European context, the legal framework for video processing in public spaces is shaped by GDPR and supervisory guidance for video devices that emphasizes data minimization, purpose limitation, and security-by-design [22, 21]. For onboard analytics, these requirements translate into architectural choices such as on-device inference, ephemeral buffering, and strict access controls, aiming to reduce unnecessary data movement while still enabling safety functions [21, 53].

The literature on violence detection spans classical computer vision, statistical pattern recognition, and deep learning. Early approaches leveraged motion energy, trajectories, and audiovisual cues such as elevated loudness or spectral signatures associated with shouting [64, 26]. Subsequent works introduced discriminative spatio-temporal descriptors [42, 16, 75], while modern methods learn video representations end-to-end and dominate generic benchmarks [38, 60, 9]. Yet, multiple surveys stress that deployment remains constrained by domain shift, scarce realistic data, and hardware/latency requirements that are rarely first-class in benchmark reporting [49, 35, 67]. A practical introduction for buses must therefore go beyond generic accuracy numbers and address real-time processing, multi-camera coordination, error costs, and human-in-the-loop escalation procedures [53, 85].

In this dissertation, the focus is on public-transportation use cases, with buses as the primary venue of study. The overarching objective is to synthesize evidence from surveillance, transportation, and modern video-understanding research, articulate computational and operational challenges specific to onboard violence detection, and ground system design choices in empirical findings as well as regulatory obligations. The chapter proceeds by positioning intelligent surveillance systems within this context, reviewing traditional approaches and their limitations, and then surveying modern AI-based frameworks through the lens of constraints and opportunities unique to bus environments [11, 53, 49].

1.2 Action Recognition and Action Detection

The automatic interpretation of human actions in video streams represents one of the most challenging and impactful problems in contemporary computer vision. The unprecedented growth of video data generated by surveillance systems, public transportation monitoring infrastructures, and smart city deployments has dramatically increased the demand for intelligent systems capable of understanding complex human behaviors in real time.

In this context, action recognition and action detection have emerged as foundational tasks enabling higher-level reasoning, decision making, and proactive intervention. Applications range from entertainment and sports analytics to safety-critical domains such as autonomous driving and public security. Among these, violence detection in surveillance footage constitutes a particularly demanding application, as it requires the accurate identification of rare, ambiguous, and context-dependent events under severe environmental constraints.

From a methodological perspective, violence detection systems are deeply rooted in the broader literature on action recognition, temporal action detection, and spatio-temporal action detection. A rigorous understanding of these foundational concepts is therefore essential to properly frame the research problem addressed in this dissertation. This section introduces the theoretical and methodological basis

of action understanding in videos, synthesizing and extending the analyses presented in recent comprehensive surveys on deep-learning-based violence detection and spatio-temporal action detection [49, 77].

The section is organized as follows. First, the concept of action recognition is formally defined and contextualized within its historical evolution. Subsequently, the transition from recognition to temporal and spatio-temporal detection is discussed, highlighting the increasing complexity of real-world surveillance scenarios. Different analysis paradigms and task taxonomies are then introduced, followed by a detailed description of the general processing pipeline adopted by modern action recognition and detection systems.

1.2.1 Action Recognition: Definitions, Scope, and Historical Evolution

Action recognition is conventionally defined as the task of assigning one or more semantic labels to a video sequence that describe the human action or activity being performed. Given a video

$$V = \{I_t\}_{t=1}^{T} \tag{1.1}$$

composed of T frames, the goal of action recognition is to estimate the most likely action class $c \in \mathcal{C}$, where $\mathcal{C}$ denotes a predefined set of action categories.

This formulation implicitly assumes that the temporal extent of the action is known in advance, meaning that the input video is temporally trimmed and contains a single dominant action. While this assumption simplifies the learning problem, it is rarely satisfied in real-world applications, particularly in surveillance contexts where videos are untrimmed, continuous, and dominated by long periods of non-action.

A key characteristic distinguishing action recognition from static image classification is the intrinsic role of time. Human actions are not defined solely by spatial appearance, but by the evolution of body posture, motion trajectories, and interactions across consecutive frames. As a result, temporal modeling constitutes a fundamental requirement for effective action recognition systems. Prior to the widespread adoption of deep learning, action recognition relied heavily on handcrafted spatio-temporal features. Classical approaches attempted to explicitly encode motion and appearance through descriptors such as Space-Time Interest Points (STIP), Histograms of Oriented Gradients (HOG), Histograms of Optical Flow (HOF), and improved dense trajectories.

While these methods achieved moderate success on controlled datasets, they suffered from several limitations. First, handcrafted features were often sensitive to noise, illumination changes, and camera motion. Second, their representational capacity was limited, preventing them from capturing complex, high-level semantic patterns. Finally, these approaches exhibited poor generalization across datasets,

particularly when applied to unconstrained surveillance footage.

As emphasized by Wang et al., the inability of handcrafted pipelines to scale and adapt to diverse real-world conditions ultimately motivated the transition toward learning-based representations [77].

The introduction of convolutional neural networks (CNNs) marked a turning point in action recognition research. Early deep learning approaches extended 2D CNNs, originally designed for image classification, to video by processing frames independently and aggregating features across time using pooling or averaging strategies. Although conceptually simple, these methods failed to fully exploit temporal dynamics.

To overcome this limitation, researchers explored architectures capable of jointly modeling spatial and temporal information. Two main directions emerged. The first involved the use of optical flow to explicitly encode motion, leading to two-stream architectures where RGB frames and optical flow were processed in parallel. The second direction introduced three-dimensional convolutions, enabling networks to learn spatio-temporal features directly from raw video volumes.

Inflated 3D convolutional networks (I3D) represented a major advancement in this area, demonstrating that spatio-temporal convolutions could effectively capture motion patterns while leveraging pretraining on large-scale image datasets. Subsequent architectures, such as SlowFast and X3D, further refined this paradigm by explicitly disentangling spatial semantics from temporal resolution, achieving strong performance with improved computational efficiency [77].

In parallel to convolutional approaches, recurrent neural networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, were introduced to model temporal dependencies in video sequences. In typical CNN–LSTM pipelines, convolutional networks extract frame-level or clip-level features, which are then fed into recurrent modules to capture temporal evolution.

Although effective for short-term modeling, RNN-based approaches suffer from well-known limitations, including vanishing gradients and limited scalability to long sequences. These issues become particularly problematic in surveillance scenarios involving extended video streams.

More recently, attention mechanisms and transformer-based architectures have gained prominence. By leveraging self-attention, transformers can model long-range temporal dependencies and selectively focus on informative regions of a video. As discussed in the survey by Wang et al., transformer-based models have demonstrated promising results in action recognition and detection, although their computational cost and data requirements remain significant challenges for real-time deployment. Within this broader landscape, violence detection can be understood as a specialized form of action recognition, where the target classes correspond to violent and non-violent behaviors. However, as highlighted by Negre et al., violence detection introduces additional layers of complexity. Violent actions are often visually ambiguous, context-dependent, and sparsely represented in available datasets.

Moreover, the boundary between violent and non-violent interactions is not always clearly defined, leading to subjective annotations and label noise.

These challenges underscore the limitations of treating violence detection as a mere classification problem and motivate the transition toward detection-oriented and temporally aware formulations, which are discussed in the following subsections.

1.2.2 From Action Recognition to Temporal and Spatio-Temporal Action Recognition

While action recognition has traditionally focused on assigning semantic labels to temporally trimmed video clips, real-world video understanding problems, particularly in surveillance and public safety contexts—rarely conform to this simplified setting. Surveillance streams are typically untrimmed and continuous, dominated by long intervals of normal or irrelevant activity, while actions of interest, including violent behaviors, occur sparsely in time and often involve only a subset of the individuals present in the scene.

This fundamental mismatch between the assumptions underlying classical action recognition and the characteristics of real surveillance data has driven a shift toward more general action detection paradigms. Temporal Action Detection (TAD) and Spatio-Temporal Action Detection (STAD) progressively relax the trimmed-video assumption by explicitly addressing not only what action is occurring, but also when and where it takes place.

The trimmed-video setting, while effective for benchmarking, substantially simplifies the learning problem by construction: the input clip contains a single dominant action and excludes irrelevant frames, allowing models to focus exclusively on discriminative patterns without reasoning about temporal boundaries or background activity. However, as emphasized in both the violence detection literature and recent STAD surveys [49, 77], this assumption severely limits real-world applicability. In surveillance footage, violent events may emerge abruptly, last only a few seconds, and be surrounded by visually similar but non-violent interactions, causing models trained on trimmed datasets to suffer from high false-positive rates or to miss early stages of aggressive behavior when deployed on untrimmed streams.

Moreover, trimmed-video classification implicitly disregards the internal temporal structure of actions, treating the video as a short, quasi-static clip and ignoring the fact that actions evolve through distinct phases such as onset, escalation, and termination. This limitation is particularly critical for violence detection, where the timely recognition of aggressive intent is often more important than accurate classification after the event has fully unfolded. Temporal Action Detection addresses these issues by explicitly modeling the temporal localization of actions within untrimmed videos. Formally, given a video sequence V defined as in Eq. (1.1), Temporal Action Detection aims to identify a set of temporal segments

$\{(t_b^i, t_e^i)\}$, each associated with an action class c_i. Compared to action recognition, this formulation requires distinguishing action segments from background while accurately estimating their temporal boundaries, a task complicated by large variations in action duration, ambiguous transitions, and severe class imbalance due to the predominance of background frames.

In the context of violence detection, TAD offers a more realistic and operationally relevant formulation, as violent behaviors typically emerge from non-violent interactions and escalate over time. Temporal localization not only enables the detection of violent segments but also opens the possibility of identifying pre-violent phases, which is essential for proactive intervention systems.

While TAD captures the temporal dimension of actions, it does not account for their spatial extent. In many surveillance applications, knowing when an action occurs is insufficient without also determining where it occurs and which individuals are involved, motivating the formulation of Spatio-Temporal Action Detection. STAD generalizes both action recognition and TAD by jointly estimating action labels, temporal boundaries, and spatial localization across frames. As highlighted by Wang et al. [77], STAD is substantially more challenging than both action recognition and TAD, as it requires solving object detection and temporal localization simultaneously while maintaining consistency across space and time. Errors in either dimension can compromise detection quality, and their coupling further complicates model design, particularly in crowded scenes where multiple actions may occur concurrently.

For violence detection, the relevance of STAD is especially pronounced. Violent actions often involve complex interactions between multiple individuals, abrupt movements, occlusions, and close physical contact, making precise spatio-temporal localization essential for avoiding confusion between violent and non-violent behaviors.

A defining concept in STAD is the action tube, which represents the spatio-temporal trajectory of an action instance across frames. Constructing reliable action tubes requires linking detections based on spatial overlap, temporal continuity, and semantic coherence. While early approaches relied on heuristic criteria such as IoU thresholds and dynamic programming, more recent methods incorporate learned association mechanisms and long-term context modeling. Despite these advances, tube linking remains a major source of error, particularly in cluttered or crowded environments.

From a violence detection perspective, such errors can be especially detrimental, as inaccurate linking may fragment a single violent event into multiple detections or erroneously merge distinct interactions, ultimately undermining system reliability and trustworthiness.

To further clarify the conceptual progression from action recognition to spatio-temporal action detection, it is useful to adopt a unified mathematical perspective. Let $V = \{I_t\}_{t=1}^{T}$ denote a video sequence of length T, where each frame $I_t \in$

$\mathbb{R}^{H\times W\times C}$ represents a color image of height H, width W, and C channels.

Action Recognition. In classical action recognition, the learning objective can be formulated as a function

$$f_{\text{AR}} : V \to c \tag{1.2}$$

where $c \in \mathcal{C}$ is a discrete action label. The function f_{AR} implicitly assumes that the action occupies the entire temporal support of V or that irrelevant frames have been removed beforehand.

This formulation collapses the temporal dimension into a single global decision, effectively marginalizing over time. While suitable for curated benchmarks, it fails to model the temporal structure of actions and does not scale to continuous video streams.

Temporal Action Detection. Temporal Action Detection generalizes this formulation by introducing temporal localization:

$$f_{\text{TAD}} : V \to \{(c_i, t_b^i, t_e^i)\}_{i=1}^{N} \tag{1.3}$$

where N is the number of detected action instances. Here, the model must jointly infer action classes and their temporal boundaries, implicitly learning to separate foreground actions from background activity.

Spatio-Temporal Action Detection. Spatio-Temporal Action Detection further extends the output space:

$$f_{\text{STAD}} : V \to \left\{ \left(c_i, t_b^i, t_e^i, \{R_t^i\}_{t=t_b^i}^{t_e^i} \right) \right\}_{i=1}^{N} \tag{1.4}$$

where $R_t^i \subset \mathbb{R}^2$ denotes the spatial region associated with the action at time t. This formulation unifies object detection and temporal localization into a single structured prediction problem.

From an optimization standpoint, STAD represents a highly non-trivial task, as the number of action instances N is unknown *a priori*, and both spatial and temporal outputs are continuous-valued. As highlighted by Wang et al., this complexity explains the performance gap between action recognition and detection methods, despite architectural similarities.

The practical implications of these formulations can be illustrated through concrete surveillance scenarios. Consider a fixed CCTV camera monitoring the interior of a public bus.

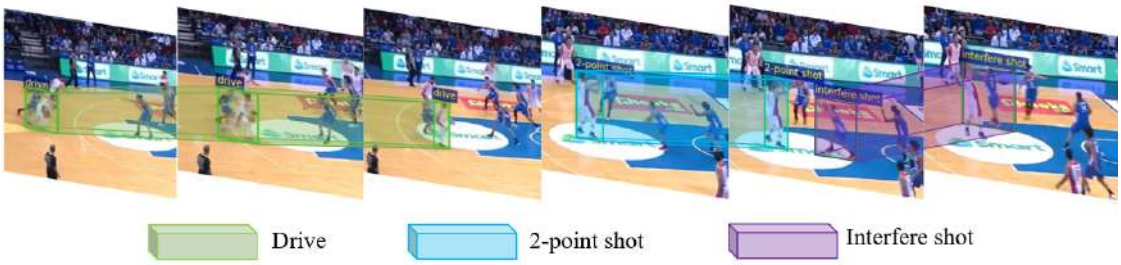


Figure 1.1: *An example of Spatio-temporal action detection (STAD): a key yet challenging video-understanding task that identifies action classes while localising them in both space and time [77, 44].*

Scenario 1: Trimmed Classification. A short video clip showing two passengers engaged in a physical altercation is manually extracted and fed into an action recognition model. The model outputs the label `Violence`. While this result may be correct, it assumes that the temporal extent of the violent action is known and ignores all preceding and following context.

Scenario 2: Temporal Detection. The full untrimmed video stream is processed by a TAD model, which outputs a temporal segment (t_b, t_e) labeled as `Violence`. This allows the system to identify when the violent interaction occurs, but it does not specify which individuals are involved, nor does it provide spatial cues for further reasoning.

Scenario 3: Spatio-Temporal Detection. An STAD model processes the same video and outputs a spatio-temporal action tube corresponding to the violent interaction. The system now knows *when* the violence occurs, *where* it occurs in the scene, and *which actors* are involved.

This distinction is not merely academic. In real deployments, spatial localization enables downstream tasks such as multi-camera tracking, re-identification, alarm visualization, and integration with other sensors, Figure 1.1 illustrates the STAD problem setting, highlighting its joint spatial and temporal localization requirements.

Violent actions exhibit a characteristic temporal structure that further motivates detection-based formulations. Rather than appearing instantaneously, violence often unfolds through a sequence of stages, including:

- a *pre-violent phase*, characterized by abnormal proximity, aggressive gestures, or rapid motion changes;
- an *escalation phase*, where physical contact or overt aggression begins;

- a *violent phase*, involving sustained physical interaction;
- a *post-violent phase*, during which the interaction subsides.

Trimmed action recognition collapses these stages into a single label, whereas temporal and spatio-temporal detection preserve their ordering and duration. This temporal decomposition is crucial for early warning systems, as it allows the detection of violence before it fully manifests.

1.2.3 Evaluation Metrics and Practical Implications

The transition from action recognition to STAD necessitates more sophisticated evaluation metrics. While classification accuracy suffices for trimmed-video recognition, detection tasks are typically evaluated using mean Average Precision (mAP) at varying temporal and spatio-temporal intersection-over-union thresholds.

These metrics reflect the joint quality of classification and localization but also reveal the sensitivity of STAD systems to boundary estimation errors. Small temporal misalignments can significantly reduce mAP, even when the semantic prediction is correct.

Negre et al. point out that many violence detection studies report high accuracy on trimmed datasets but fail to evaluate temporal localization performance, obscuring their true effectiveness in real-world surveillance settings [49]. This observation further reinforces the necessity of framing violence detection as a detection problem rather than a pure classification task. The progression from action recognition to TAD and STAD has profound implications for violence detection. Treating violence detection as an STAD problem enables systems to operate on continuous video streams, identify the onset and duration of violent events, and localize the individuals involved.

Moreover, this formulation naturally extends to early action detection and anticipation, where the goal is to predict violent behavior before it fully manifests. Such capabilities are essential for proactive safety systems in public transportation and surveillance environments.

1.2.4 Early Action Detection and Violence Anticipation

Early action detection, also referred to as action anticipation, aims to predict the occurrence of an action before its completion. Formally, given a partial observation of the video sequence $V_{1:t} = \{I_\tau\}_{\tau=1}^{t}$ with $t < t_e$, the objective is to estimate the posterior probability

$$p(c \mid V_{1:t}), \tag{1.5}$$

where c corresponds to an action that will occur in the future.

From a detection perspective, early violence detection can be framed as the problem of identifying the onset of a violent action at time t_b while observing only a fraction of the full action duration. This setting introduces an additional trade-off between timeliness and accuracy: predictions must be made as early as possible, but premature decisions increase the risk of false alarms.

The STAD formulation provides a natural foundation for early detection, as it explicitly models temporal boundaries and actor trajectories. By analyzing partial action tubes and their early evolution, a system can estimate the likelihood that an observed interaction will escalate into violence.

Most existing violence detection methods do not explicitly address this early detection problem, focusing instead on retrospective classification [49, 77]. This gap represents a critical limitation for real-world surveillance systems, where prevention and timely intervention are paramount. The formal and conceptual considerations discussed above have direct implications for model design. Systems intended for violence detection in surveillance environments should:

1. operate on untrimmed video streams rather than curated clips;
2. explicitly model temporal structure and action boundaries;
3. incorporate spatial localization to disambiguate interactions;
4. support early detection by leveraging partial observations.

These requirements strongly favor detection-oriented architectures over pure classification models. They also motivate the integration of spatio-temporal reasoning, long-term context modeling, and efficient inference mechanisms, which are examined in detail in the subsequent sections of this chapter.

1.2.5 Frame-Based, Clip-Based, and Event-Based Analysis Paradigms

The design of an action recognition or detection system is strongly influenced by the temporal granularity at which video data are analyzed. In the literature, three dominant paradigms can be identified: frame-based analysis, clip-based analysis, and event-based analysis [28, 74]. Each paradigm embodies a distinct set of assumptions regarding the temporal structure of actions and entails specific trade-offs in terms of representational power, computational complexity, and suitability for real-world surveillance applications.

This subsection provides a critical examination of these paradigms, with particular emphasis on their limitations in CCTV environments and their implications for violence detection systems.

Frame-based analysis represents the most elementary paradigm for action understanding, in which each video frame is processed independently and decisions are derived from static appearance cues [16, 32]. Its appeal lies in computational simplicity and compatibility with mature image-based architectures. However, the paradigm fundamentally ignores temporal dynamics, which are essential to disambiguate visually similar interactions in surveillance footage [75]. In crowded CCTV scenes, isolated frames rarely provide sufficient information to distinguish benign proximity from aggressive behavior, leading to high false positive rates. As emphasized in the violence detection literature, frame-level methods exhibit brittle behavior under occlusion, low resolution, and viewpoint compression, making them unsuitable as standalone solutions for real-world surveillance [72, 33].

The primary advantage of frame-based analysis lies in its efficiency. Processing frames independently enables real-time inference and simplifies system design. Moreover, frame-based methods can leverage advances in image-based object detection and classification, benefiting from mature architectures and large-scale pretraining [32, 34].

Despite these advantages, frame-based analysis suffers from severe conceptual and practical limitations. Most notably, it ignores temporal dynamics, which are essential for distinguishing between actions that share similar visual appearances. In surveillance footage, isolated frames often lack sufficient information to disambiguate between benign and aggressive interactions. For example, a single frame depicting close physical proximity between individuals may correspond to a hug, a crowded environment, or the onset of a violent altercation [4, 19].

Frame-level violence detection systems are highly susceptible to false positives and false negatives due to their inability to capture motion patterns and temporal evolution [25, 57]. This limitation is exacerbated in CCTV environments characterized by low resolution, occlusions, and fixed camera viewpoints [55, 20].

Furthermore, frame-based methods are inherently ill-suited for early violence detection. Since they lack temporal context, they cannot model the progression from pre-violent behavior to overt aggression. As a result, such systems tend to react only after violence becomes visually explicit, undermining their preventive potential [46, 49].

Clip-based analysis introduces short-term temporal context by operating on fixed-length sequences of consecutive frames [39]. This paradigm enables the extraction of local motion patterns and underpins most modern video architectures, including 3D CNNs and two-stream networks [68, 60]. Empirically, clip-based models dominate benchmark performance in violence detection, as they better capture motion cues associated with physical aggression [62, 3]. However, their reliance on fixed temporal windows implicitly assumes alignment between clip boundaries and action boundaries, an assumption rarely satisfied in untrimmed surveillance streams [51]. As a result, clip-based models often struggle with temporal localization and exhibit degraded stability when deployed on continuous CCTV footage

[12].

Clip-based analysis strikes a balance between representational power and computational feasibility. By capturing short-term temporal dynamics, clip-based models can differentiate between visually similar actions based on motion cues. Modern deep learning architectures, such as 3D CNNs and two-stream networks, are predominantly designed for this paradigm [10, 24].

Despite their success, clip-based methods remain constrained by their limited temporal receptive field. The choice of clip length introduces a trade-off: short clips may fail to capture meaningful action dynamics, while long clips increase computational cost and dilute discriminative information [76].

More critically, clip-based models implicitly assume that actions are temporally centered within the clip or that the clip boundaries align with action boundaries. This assumption is rarely satisfied in untrimmed surveillance footage [65]. Consequently, clip-based models often struggle with temporal localization and exhibit degraded performance when applied to continuous video streams [77].

From an early detection perspective, clip-based models are reactive rather than predictive. They typically require a substantial portion of the action to be observed before producing confident predictions, limiting their usefulness for early intervention [49].

Event-based analysis models actions as temporally extended processes embedded within long, untrimmed video streams [77]. Rather than classifying isolated clips, event-based approaches aim to detect action instances, estimate their temporal extent, and, in spatio-temporal formulations, localize the actors involved [29]. This paradigm aligns naturally with the characteristics of violent behavior, which unfolds through progressive stages rather than instantaneous cues [8]. While computationally demanding, event-based formulations provide the necessary foundation for robust surveillance systems operating in complex, crowded environments.

The event-based paradigm aligns naturally with the characteristics of violent behavior. Violence is not an instantaneous phenomenon but a process that evolves over time, often preceded by observable cues [73]. Modeling violence as an event enables systems to capture this progression and supports early detection and anticipation.

Furthermore, event-based analysis accommodates overlapping and concurrent actions, which are common in crowded surveillance scenes [44]. By maintaining explicit action instances, systems can reason about multiple interactions simultaneously.

The primary challenge of event-based analysis lies in its complexity. Detecting events in untrimmed videos requires models to handle long temporal dependencies, severe class imbalance, and ambiguous action boundaries [31]. Additionally, event-based systems must operate efficiently to be viable for real-time surveillance [47].

As highlighted in the STAD survey by Wang et al., current event-based methods often rely on sophisticated linking algorithms, proposal generation mechanisms, and

multi-stage pipelines, which increase system complexity and sensitivity to errors [77].

When evaluated in the context of CCTV surveillance, the limitations of frame-based and clip-based paradigms become particularly pronounced. Fixed camera angles, low spatial resolution, and frequent occlusions reduce the reliability of static appearance cues [70]. Temporal reasoning becomes essential to disambiguate actions and suppress false alarms.

Many violence detection studies report promising results on curated datasets but fail to generalize to real-world surveillance footage, precisely because they rely on clip-based or frame-based assumptions [67, 33].

Event-based approaches, while more demanding, offer a principled solution to these challenges. By explicitly modeling temporal structure and action evolution, they provide the necessary foundation for robust violence detection in operational environments [54].

Early violence detection fundamentally requires a paradigm shift from classification to event modeling. Frame-based and clip-based methods are inherently retrospective, responding only after violence becomes visually salient. Event-based analysis, in contrast, enables the detection of subtle precursors and supports anticipatory decision-making [77].

This observation underscores the necessity of adopting spatio-temporal detection frameworks for violence detection in public transportation and surveillance systems.

1.2.6 Taxonomy of Action Detection Tasks

The diversity of real-world action understanding problems has led to the formulation of multiple variants of action detection tasks, each characterized by distinct objectives, assumptions, and levels of supervision. A rigorous taxonomy is essential not only to organize existing methods, but also to clarify their applicability and limitations in specific domains such as violence detection in surveillance environments.

This subsection introduces a multi-dimensional taxonomy of action detection tasks. The taxonomy is structured along four principal axes: temporal scope, spatial scope, supervision level, and temporal anticipation capability.

One of the most fundamental distinctions among action understanding tasks concerns the temporal extent of the modeled actions. From this perspective, actions can be categorized as instantaneous, short-term, or long-term.

Instantaneous actions are characterized by brief, visually salient events that can often be recognized from a small number of frames. Examples include gestures such as waving or pointing. While such actions are common in curated datasets, they are relatively rare in violence detection scenarios, where meaningful patterns typically unfold over longer temporal intervals.

Short-term actions span a limited temporal window, typically ranging from a fraction of a second to several seconds. Many violent behaviors, such as punches or pushes, fall into this category. However, even within this class, the temporal boundaries of actions are often ambiguous, complicating detection and evaluation.

Long-term actions or activities extend over longer durations and may involve multiple sub-actions or phases. In surveillance footage, violence often manifests as a long-term activity composed of preparatory, escalation, and resolution phases. Modeling such actions requires systems capable of capturing long-range temporal dependencies and reasoning over extended sequences.

This temporal taxonomy highlights a key limitation of many existing violence detection methods, which implicitly assume short-term actions and therefore struggle to capture the full temporal structure of violent events.

Action detection tasks also differ in terms of spatial localization requirements. As discussed in Section 1.2.2, these differences give rise to distinct problem formulations.

Temporal Action Detection focuses exclusively on identifying the temporal boundaries of actions. Spatial information is either ignored or assumed to be implicitly encoded in the video representation. While TAD simplifies the detection problem, it is insufficient for applications requiring actor localization or interaction analysis.

Spatio-Temporal Action Detection extends TAD by explicitly localizing actions in space and time. STAD models output action tubes that track the spatial extent of actions across frames.STAD is the most general and challenging formulation, subsuming both action recognition and TAD.

For violence detection, spatial localization is critical. Violent actions often involve close physical interactions between specific individuals, and the ability to localize these interactions enables downstream reasoning, visualization, and response mechanisms. The absence of spatial reasoning is a major shortcoming of many violence detection approaches.

Another important dimension of the taxonomy concerns the level of supervision required during training. This aspect has significant practical implications, particularly in domains where annotated data are scarce or costly to obtain.

Fully supervised approaches rely on precise temporal and, in the case of STAD, spatial annotations. While these methods often achieve the highest performance on benchmark datasets, their reliance on dense annotations limits scalability and generalization. In violence detection, obtaining accurate spatio-temporal annotations is especially challenging due to ethical concerns, subjectivity, and the rarity of violent events.

Weakly supervised methods aim to reduce annotation costs by relying on coarse labels, such as video-level or clip-level annotations. These approaches typically infer temporal or spatial localization implicitly during training. Although promising, weakly supervised methods often exhibit reduced localization accuracy and remain sensitive to dataset bias.

Self-supervised and unsupervised approaches seek to leverage large volumes of unlabeled video data by exploiting intrinsic temporal and spatial structure. While still an active area of research, these methods hold particular promise for surveillance applications, where labeled data are limited but raw video streams are abundant.

Beyond retrospective detection, an increasingly important class of tasks focuses on anticipating actions before they fully occur. This dimension is particularly relevant for violence detection in safety-critical environments.

Most existing action detection methods operate retrospectively, identifying actions only after they have occurred. While suitable for offline analysis, such approaches offer limited utility for prevention and real-time intervention.

Online detection methods process video streams incrementally, producing predictions with minimal delay. These approaches must balance accuracy with latency and computational efficiency. In violence detection, online detection enables faster response but still often reacts after violence has become explicit.

Early action detection aims to predict actions based on partial observations. Formally, the task involves estimating the probability of an action occurring in the future given incomplete temporal information. This formulation aligns naturally with the concept of early violence detection, where the objective is to identify precursors of violent behavior before physical harm occurs.

1.2.7 General Pipeline for Action Recognition and Detection Systems

Despite the wide variety of architectures and learning paradigms proposed in the literature, most action recognition and detection systems can be described within a common end-to-end processing pipeline. This pipeline defines the sequence of transformations applied to raw video data in order to extract meaningful representations, model temporal dynamics, and ultimately infer action-related decisions.A high-level overview of this processing pipeline is shown in Figure 1.2.

In surveillance and violence detection scenarios, the design of this pipeline is particularly critical. Each stage must balance representational richness with computational efficiency, robustness to noise, and the ability to operate on untrimmed video streams in real time. Drawing from the analyses presented in recent surveys on spatio-temporal action detection and violence recognition, this subsection provides a detailed description of the general pipeline, with explicit emphasis on early violence detection.

1.2.8 Input Acquisition and Video Characteristics

The pipeline begins with the acquisition of raw video streams, typically captured by fixed surveillance cameras operating continuously over extended periods. In public

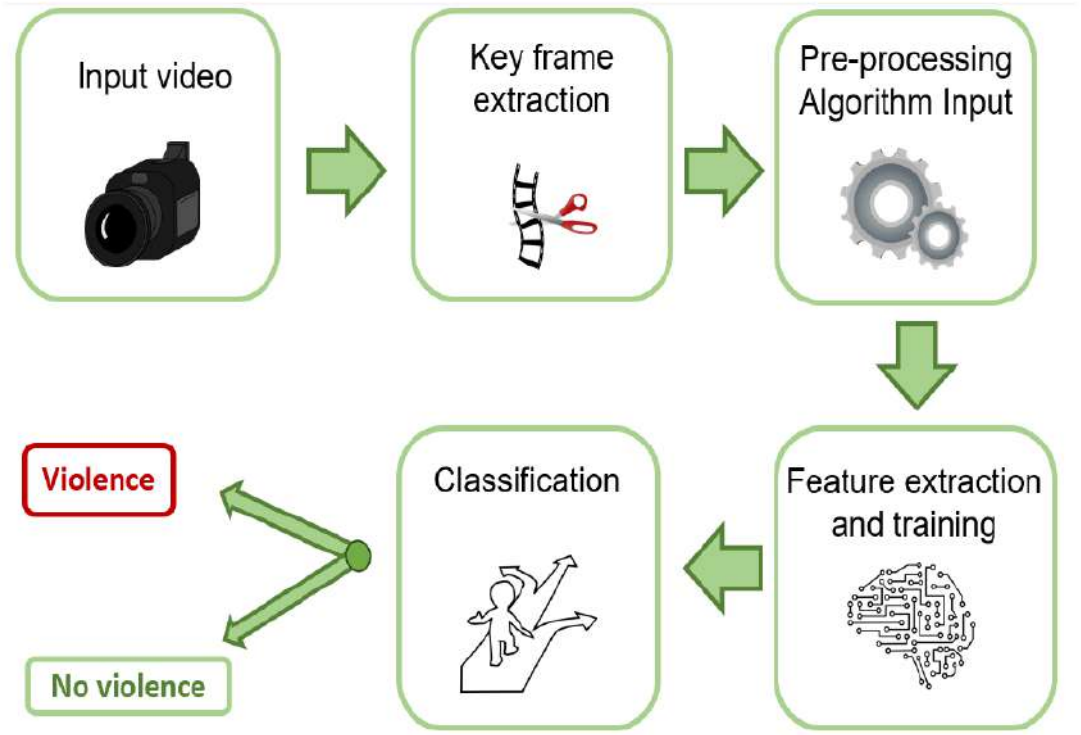


Figure 1.2: *General pipeline and taxonomy of deep-learning-based violence detection systems, including input modalities, preprocessing, feature extraction, and classification stages [49].*

transportation and CCTV environments, these streams are characterized by low to moderate spatial resolution, fixed viewpoints, compression artifacts, and highly variable illumination conditions.

Unlike curated datasets, real surveillance footage is dominated by background activity, with violent events occurring infrequently and unpredictably. This imbalance fundamentally shapes the downstream stages of the pipeline, as models must learn to discriminate rare and subtle patterns from overwhelming amounts of non-informative data.

1.2.9 Temporal Sampling and Key Segment Selection

Given the redundancy inherent in video data, temporal sampling constitutes a crucial preprocessing step. Sampling strategies determine which frames or clips are forwarded to subsequent stages and directly influence both performance and computational cost.

Common approaches include uniform frame sampling, sliding-window clip extraction, and adaptive sampling based on motion or scene changes. In detection-oriented systems, sampling strategies must preserve temporal continuity to avoid missing the onset of actions. Aggressive subsampling may reduce latency but risks eliminating early indicators of violent behavior.

The choice of temporal granularity plays a pivotal role in spatio-temporal detection performance, particularly for actions with short or ambiguous temporal extents.

1.2.10 Pre-processing and Signal Enhancement

Pre-processing aims to normalize input data and enhance salient cues relevant to action understanding. Typical operations include spatial resizing, normalization, background suppression, and motion enhancement through optical flow computation.

In surveillance environments, pre-processing must address challenges such as camera noise, motion blur, and occlusions. Inadequate pre-processing can significantly degrade violence detection performance, particularly in low-quality CCTV footage.

Importantly, pre-processing decisions influence the type of features that can be effectively learned. For instance, optical flow emphasizes motion patterns but increases computational complexity, whereas raw RGB processing preserves appearance cues but may obscure subtle motion dynamics.

1.2.11 Feature Extraction and Representation Learning

Feature extraction constitutes the core of the pipeline, transforming preprocessed video data into compact representations suitable for learning and inference. Modern systems employ deep neural networks to learn spatial, temporal, or spatio-temporal features directly from data.

Frame-based representations capture static appearance information, while clip-based representations encode short-term motion patterns. In STAD frameworks, spatio-temporal feature maps are often extracted using 3D convolutions or hybrid architectures, enabling joint reasoning over space and time.

For violence detection, representation learning must capture both low-level motion cues and high-level interaction patterns. Violent actions are often defined more by relational dynamics between individuals than by isolated poses or objects.

1.2.12 Temporal Modeling and Context Integration

Beyond local feature extraction, effective action detection requires modeling temporal dependencies and contextual relationships. Temporal modeling modules aggregate features across time to capture action evolution, suppress noise, and disambiguate visually similar interactions.

Approaches range from recurrent neural networks and temporal convolutions to attention-based and transformer architectures. In STAD systems, temporal modeling is often intertwined with spatial localization, enabling the construction of action tubes that track actors over time.

In surveillance scenarios, long-term context is particularly valuable. The ability to relate current observations to past behavior enables systems to distinguish between transient anomalies and meaningful precursors of violence. Iinsufficient temporal context is a primary cause of fragmented or unstable detections in spatio-temporal action detection.

1.2.13 Classification, Localization, and Detection Heads

At this stage, learned representations are mapped to action predictions. Depending on the task formulation, this may involve classification heads, temporal boundary regression, spatial bounding box regression, or combinations thereof [77].

In STAD frameworks, detection heads output candidate action instances along with confidence scores [29]. These outputs may be generated at the frame level or clip level and subsequently linked to form complete action tubes. The design of detection heads directly affects the system's ability to localize actions accurately and efficiently [77].

For violence detection, detection heads must cope with severe class imbalance and ambiguous boundaries. Huillcen Baca et al. highlight that lightweight detection heads and carefully designed loss functions are essential for achieving real-time performance without sacrificing accuracy [35]. In this context, loss re-weighting strategies and focal loss formulations have proven effective in mitigating class imbalance [31, 45].

Post-processing operations refine raw model outputs to produce coherent and actionable predictions. These operations may include non-maximum suppression, temporal trimming, smoothing, and confidence thresholding [25].

Temporal refinement is particularly important in detection tasks, as initial predictions often span larger temporal regions than the true action extent [65]. Improper trimming can lead to delayed detection or fragmented action instances, both of which are undesirable in surveillance applications.

The final stage of the pipeline concerns early violence detection and decision-making. Rather than waiting for an action to fully unfold, early detection systems aim to raise alerts based on partial observations and evolving evidence [49].

Formally, early violence detection can be viewed as estimating the probability of a violent event given a prefix of the video stream [77]. This requires integrating temporal cues, contextual information, and uncertainty estimation. Systems must trade off timeliness against reliability, minimizing both detection latency and false alarms.

As emphasized in recent studies, most existing violence detection pipelines are optimized for retrospective analysis and lack explicit mechanisms for early detection [49, 77]. Addressing this limitation requires rethinking the pipeline as a continuous, adaptive process rather than a static classification task.

1.3 Deep Learning Approaches for Action and Violence Recognition

The problem of automatic violence detection in video surveillance has progressively evolved from heuristic, handcrafted pipelines toward deep learning–based approaches capable of learning spatio-temporal representations directly from data. This evolution mirrors the broader trajectory of action recognition research, where the shift from manual feature engineering to end-to-end learning has enabled significant performance gains, albeit at the cost of increased computational complexity and data requirements.

From a conceptual standpoint, most deep learning approaches for violence recognition can be interpreted as specific instantiations of the general action recognition and detection pipeline discussed in Section 1.2.7. However, the peculiar characteristics of violent behavior, combined with the constraints of surveillance environments, impose additional challenges that differentiate violence recognition from generic action recognition tasks.

In particular, as extensively analyzed in the recent survey, violence detection methods must cope with: (i) strong visual ambiguity between violent and non-violent interactions, (ii) severe class imbalance due to the rarity of violent events, (iii) limited availability of large-scale, well-annotated datasets, and (iv) the need for real-time or near-real-time inference in operational settings. These factors strongly influence architectural choices, training strategies, and evaluation protocols.

This section provides a comprehensive technical survey of deep learning approaches for action and violence recognition, with a critical analysis of their strengths, limitations, and suitability for early violence detection in CCTV environments. The discussion integrates the taxonomy and methodological insights from the spatio-temporal action detection literature [77] with the domain-specific analysis of violence recognition methods presented in Section 4 of the survey by Negre et al. [49].

1.3.1 Design Space of Deep Architectures for Violence Recognition

A central design question in video-based violence recognition concerns how temporal dynamics are modeled and integrated with spatial information. Formally, given a video sequence $\mathbf{X}_{1:T}$, deep learning approaches aim to approximate the conditional distribution $p(y \mid \mathbf{X}_{1:T})$, where y denotes a violence-related label, such as a binary violent/non-violent classification or a finer-grained action category. Different architectural families correspond to different factorizations and approximations of this distribution, reflecting distinct assumptions about the relative importance and interaction of spatial and temporal cues.

Negre et al. [49] categorize existing deep learning–based violence detection methods into several major families, each characterized by a specific strategy for temporal modeling. In the following, these families are discussed in a unified framework, highlighting their relevance to surveillance scenarios and early violence detection. A quantitative synthesis of the literature further reveals strong architectural regularities and a marked sensitivity to dataset realism. Figures 1.3–1.5 summarize the distribution of architectural families and their combinations, while Figures 1.6–1.8 report representative benchmark accuracies across three widely used datasets.

These results highlight a structural gap between reported benchmark performance and robustness under real-world surveillance conditions, motivating detection-oriented and domain-aligned approaches.

1.3.2 2D CNN-Based Approaches with Temporal Aggregation

In this paradigm, a CNN encoder $\phi(\cdot)$ operates independently on each frame $\mathbf{X}_t$, producing a sequence of spatial embeddings

$$\mathbf{z}_t = \phi(\mathbf{X}_t)), \tag{1.6}$$

that are subsequently aggregated over time using simple pooling or voting strategies, implicitly assuming temporal independence between frames.

This class of methods constitutes one of the earliest and most widely explored approaches in the violence detection literature, largely due to its simplicity and compatibility with pretrained image classification backbones. Representative works employ architectures such as VGG, ResNet, or MobileNet to extract frame-level features, followed by shallow temporal fusion.

While computationally efficient, these approaches suffer from fundamental limitations. By marginalizing temporal structure through simple aggregation, they discard motion information that is often critical for distinguishing violent interactions from visually similar but benign behaviors. As a consequence, frame-based

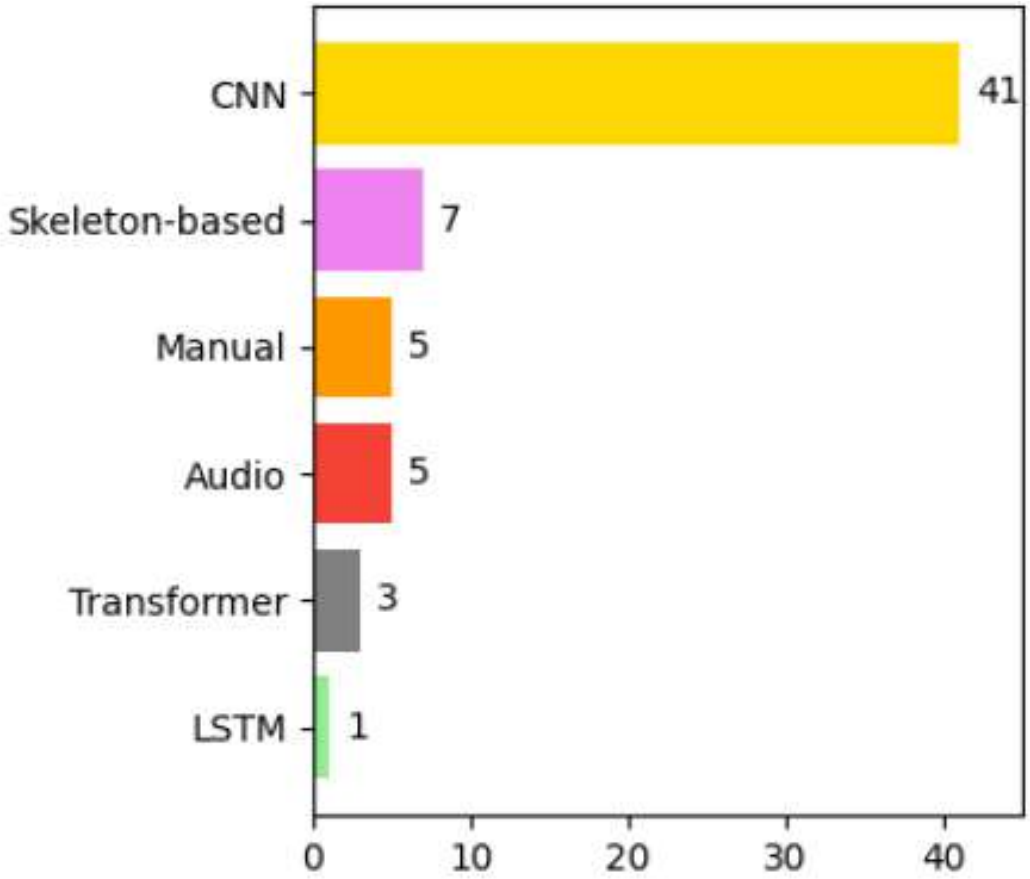


Figure 1.3: *Count of the types of algorithms used in violence detection phase 1 in the selected articles grouped by category [49].*

CNN methods are particularly prone to false positives in crowded scenes or during non-violent physical contact, a limitation repeatedly observed across multiple datasets.

From the perspective of early violence detection, purely frame-based CNN approaches are inherently reactive. They typically respond only when violence becomes visually explicit, offering limited predictive capability with respect to pre-violent behavior.

1.3.3 Two-Stream and Optical Flow–Based Architectures

Another prominent family of approaches exploits the separation between appearance and motion by employing two-stream architectures. In these models, one stream processes RGB frames, while the other processes motion representations, typically optical flow. The outputs of the two streams are fused at different stages to produce a final prediction.

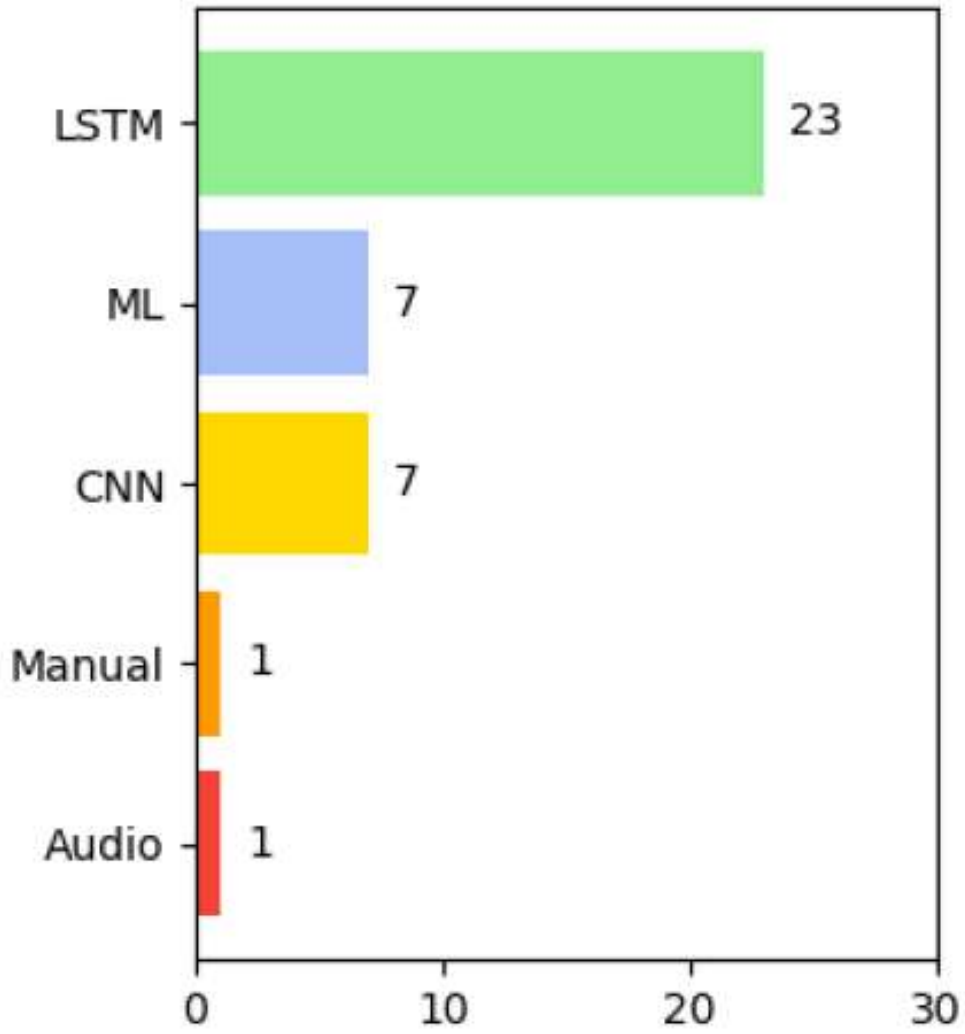


Figure 1.4: *Count of the types of algorithms used in violence detection phase 2 in the selected articles grouped by category [49].*

Two-stream architectures often achieve improved performance compared to single-stream CNNs, particularly on datasets where motion cues are discriminative of violent behavior. Optical flow emphasizes rapid, irregular movements and collisions, which are characteristic of many violent actions.

Despite these advantages, flow-based approaches introduce significant computational overhead and are sensitive to noise, camera motion, and compression artifacts—factors that are ubiquitous in CCTV footage. As a result, their deployment in real-time surveillance systems is often impractical without substantial optimization. Moreover, optical flow primarily captures short-term motion and does not

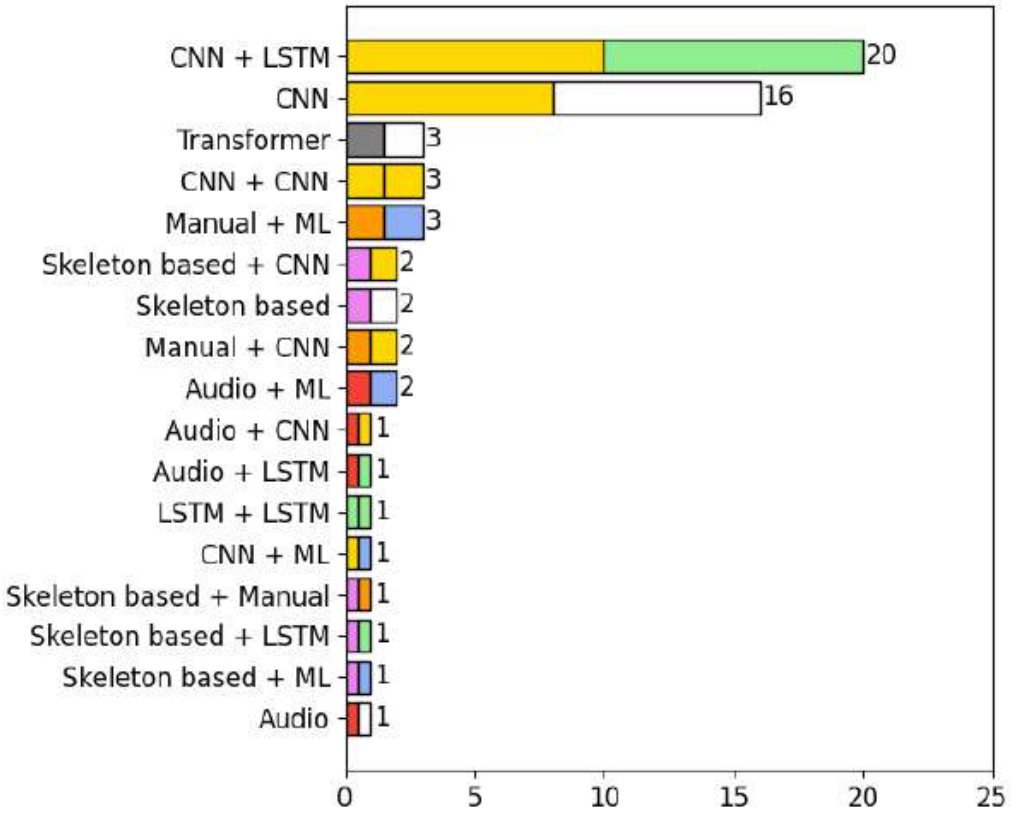


Figure 1.5: *Count of the types of algorithm combinations used in the selected articles grouped by subcategory. Colors: Yellow corresponds to the use of CNNs, purple to skeleton-based algorithms, orange to manual techniques, red to those utilizing audio, gray to those employing transformers, and green to LSTM [49].*

explicitly encode higher-level interaction semantics, which are crucial for early violence detection.

1.3.4 3D Convolutional Neural Networks

Three-dimensional convolutional neural networks extend 2D convolutions to the temporal dimension, enabling joint learning of spatial and temporal features directly from video volumes. Architectures such as C3D, I3D, R(2+1)D, and their lightweight variants have been widely adopted for action recognition and, more recently, for violence detection. 3D CNNs generally outperform 2D-based methods on benchmark violence datasets, as they better capture motion dynamics without requiring explicit optical flow computation. However, their high computational cost and memory footprint pose challenges for deployment in resource-constrained environments.

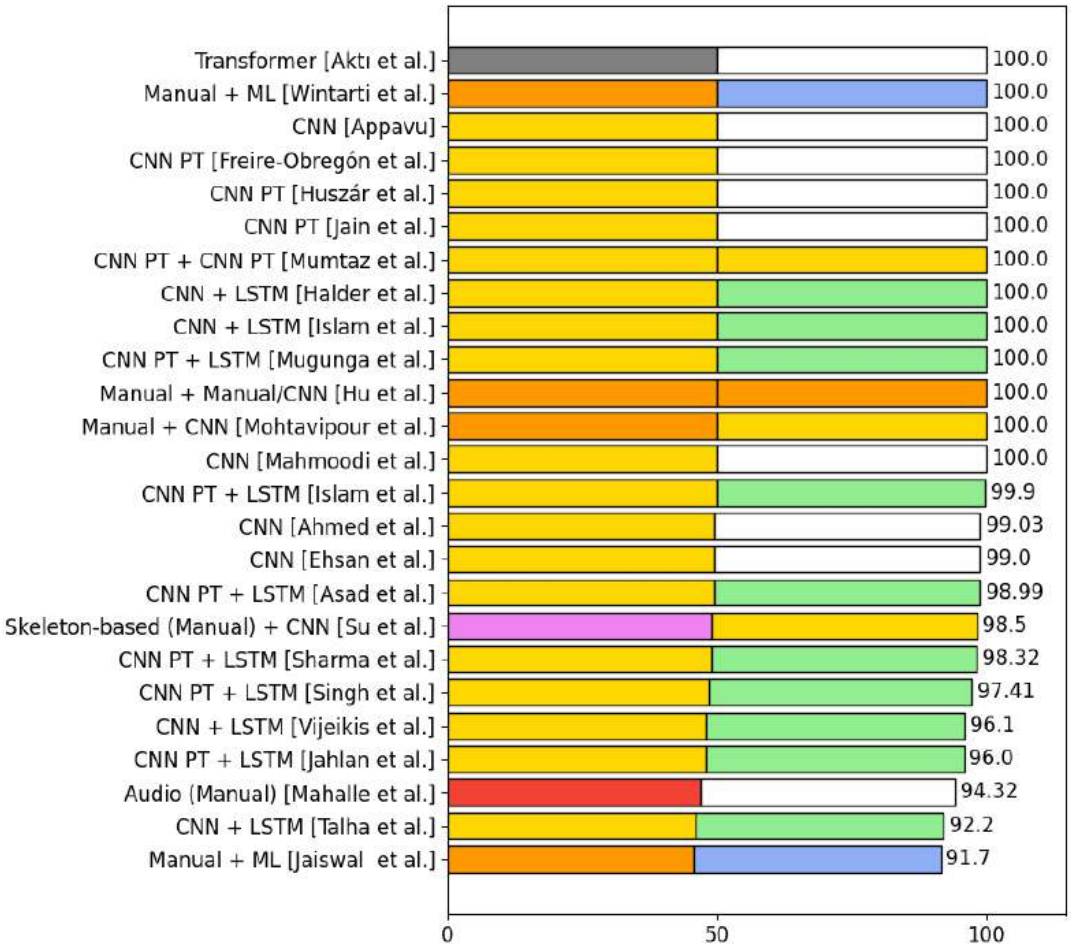


Figure 1.6: *Accuracy obtained by selected items in the Action Movies dataset. Colors: Yellow corresponds to the use of CNNs, purple to skeleton-based algorithms, orange to manual techniques, red to those utilizing audio, gray to those employing transformers, and green to LSTM [49].*

Three-dimensional convolutional neural networks (3D CNNs) extend 2D kernels along time, enabling joint reasoning over appearance and motion within short clips [38, 68]. Two-stream networks decouple appearance and motion via separate RGB and optical-flow streams [60], while I3D inflates pretrained 2D filters into 3D kernels to exploit transfer learning [10]. Factorized models such as R(2+1)D reduce compute by separating temporal and spatial operators [69], and multi-path designs such as SlowFast capture both slow semantic context and fast motion transients [24]. Efficiency-centric designs like X3D further improve accuracy-to-FLOPs trade-offs, which is pivotal for embedded multi-stream inference [23].

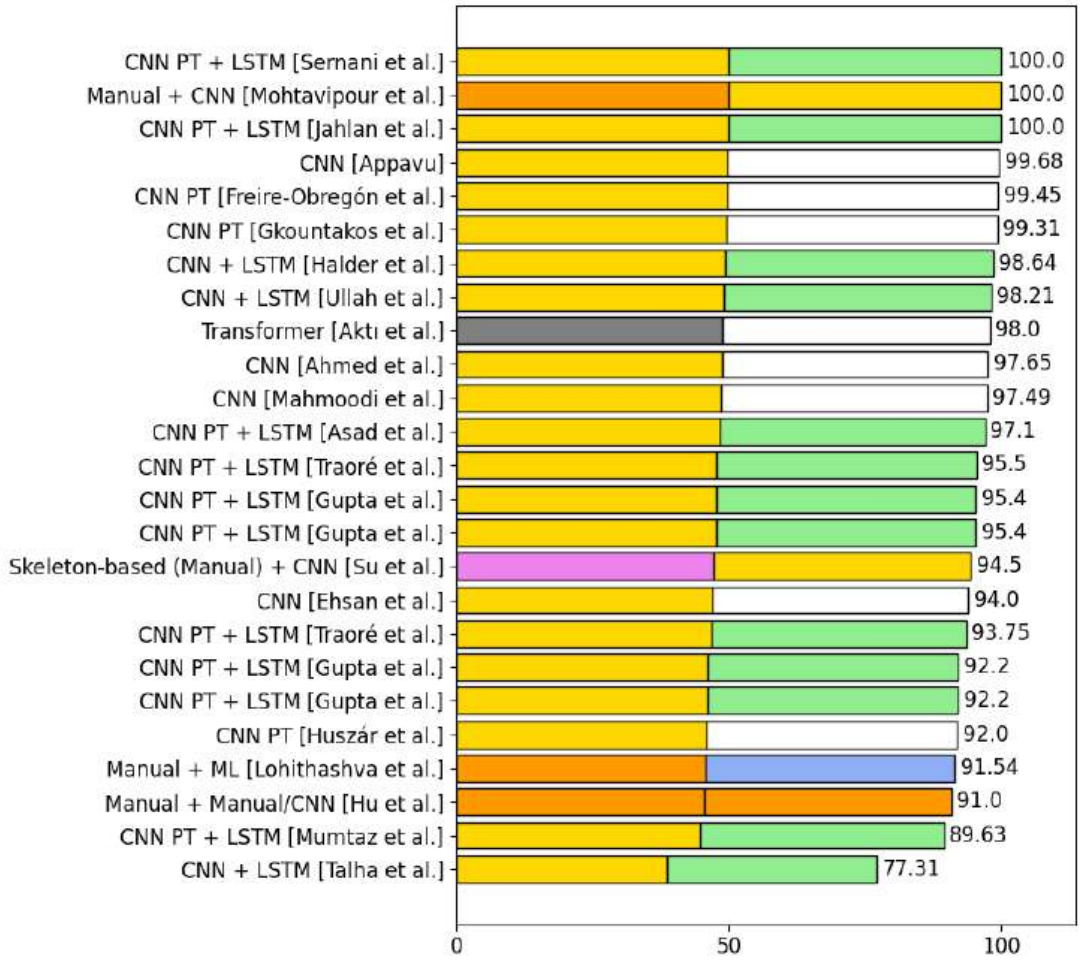


Figure 1.7: *Accuracy obtained by selected items in the Violent Flow dataset. Colors: Yellow corresponds to the use of CNNs, purple to skeleton-based algorithms, orange to manual techniques, red to those utilizing audio, gray to those employing transformers, and green to LSTM [49].*

When the task is framed as STAD, architectural choices are coupled to post-processing. STAD methods are commonly categorized into frame-level and clip-level approaches: frame-level methods produce per-frame boxes and then link them across time, whereas clip-level methods output tubelets (short sequences of boxes) that are subsequently linked into full action tubes [77]. Linking algorithms typically combine spatial overlap (IoU) and classification/actionness scores, often solved with dynamic programming/Viterbi-style optimization; additional temporal trimming methods aim to refine the start/end boundaries to mitigate transition-state ambiguity [77]. This matters for buses, where transitional states (e.g., pushing through a crowd) can be visually similar to early-stage aggression.

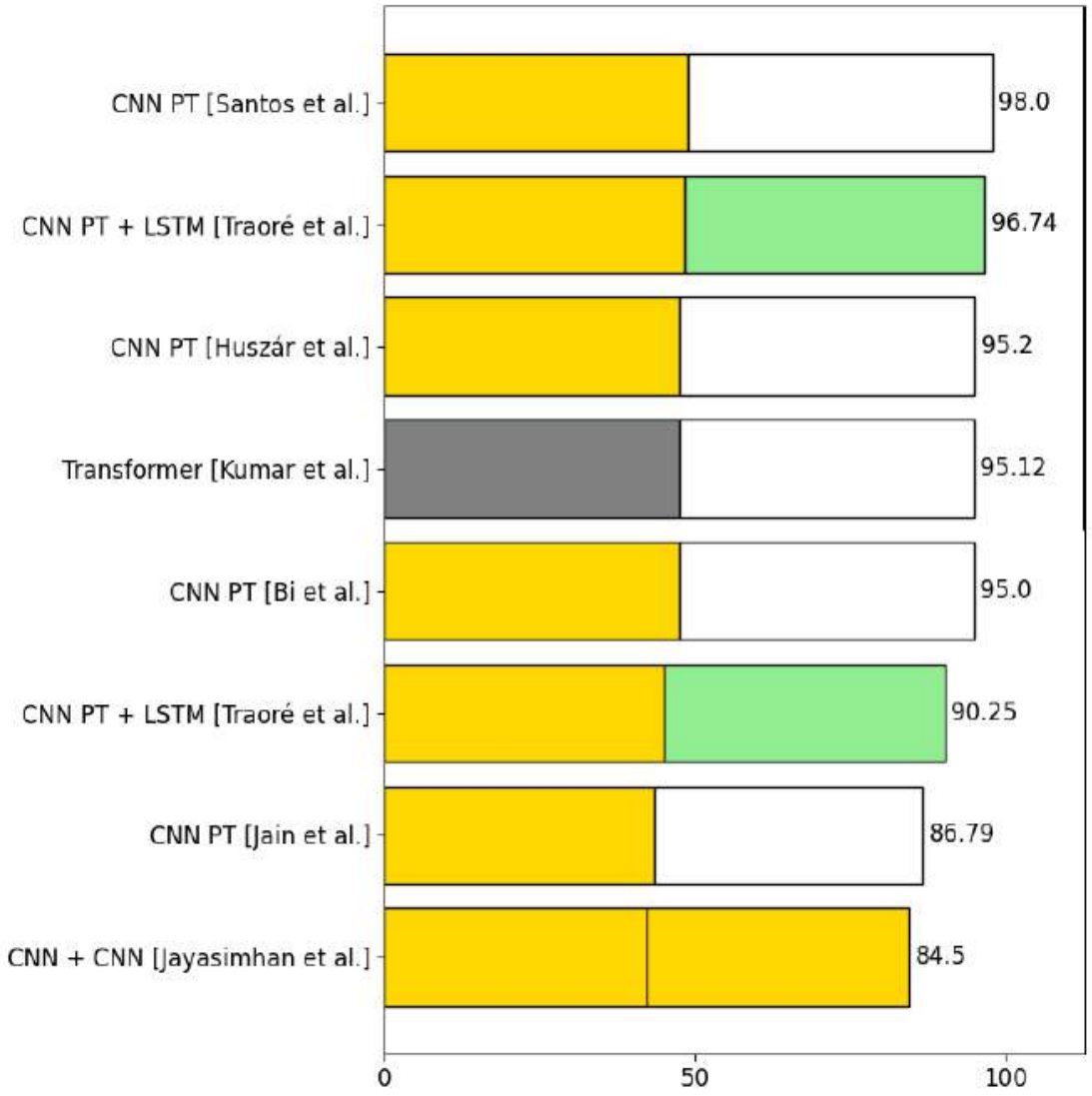


Figure 1.8: *Accuracy obtained by selected items in the Real Life Violent Scenes Dataset. Colors: Yellow corresponds to the use of CNNs, purple to skeleton-based algorithms, orange to manual techniques, red to those utilizing audio, gray to those employing transformers, and green to LSTM [49].*

Transformer-based architectures have also become influential in STAD, motivated by their capacity to model long-range dependencies and relations among actors and context. Surveyed frameworks include transformer-style tubelet detectors that directly output action tubelets from a single representation, and relation-centric models that integrate spatio-temporal attention or graph reasoning over actor tubelets [77]. While transformer inference can be compute-intensive, hybrid designs—combining efficient backbones with limited attention windows—are

promising for embedded surveillance, provided their latency and calibration behavior are explicitly validated.

For CCTV and bus interiors, these architectures must be evaluated not only by benchmark accuracy but by resilience to occlusions, compression, and viewpoint shifts, and by their behavior under severe class imbalance. Multi-rate and factorized designs are attractive because they offer control over temporal resolution versus compute load, enabling real-time operation without discarding temporal cues critical for disambiguating aggressive interactions [24, 23].

1.3.5 CNN–RNN Architectures: Explicit Temporal Modeling

To overcome the lack of temporal reasoning in frame-based CNN approaches, a substantial body of work integrates recurrent neural networks, such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) architectures, on top of CNN feature extractors. In this formulation, the CNN encodes each frame or short clip into a feature representation, while the recurrent module explicitly models temporal dependencies across the sequence according to

$$\mathbf{h}_t = \mathrm{RNN}(\mathbf{z}_t, \mathbf{h}_{t-1}), \qquad \hat{y} = g(\mathbf{h}_T), \tag{1.7}$$

where $\mathbf{h}_t$ denotes the hidden state at time t and $g(\cdot)$ represents a task-specific classification head.

Negre et al. [49] report that CNN–LSTM architectures represent one of the most common design choices for violence recognition, particularly in early works addressing datasets such as Hockey Fight, Violent-Flows, and Movies. These models explicitly capture temporal evolution and can, in principle, recognize patterns associated with escalation toward violence.

However, their effectiveness is highly dependent on the quality of the extracted features and the length of the temporal window. In long surveillance videos, recurrent models often struggle to maintain stable representations, and their training can be affected by vanishing gradients and overfitting. Moreover, CNN–RNN pipelines tend to focus on retrospective recognition rather than anticipation, limiting their applicability to early detection scenarios.

1.3.6 Transformer-Based Models and Attention Mechanisms

More recent approaches leverage attention mechanisms and transformer architectures to model long-range temporal dependencies. By allowing each temporal token to attend to all others, transformers provide a powerful framework for capturing complex interactions and contextual cues.

While transformer-based models have shown promising results in generic action recognition and spatio-temporal action detection [77], Negre et al. [49] note that

their adoption in violence detection remains limited. This is primarily due to their high data requirements and computational complexity, which are difficult to reconcile with the scarcity of violence datasets and real-time constraints.

Nevertheless, attention-based mechanisms are conceptually well aligned with early violence detection, as they can highlight subtle precursors and relational patterns over extended temporal horizons.

1.3.7 Skeleton-based and Pose-driven Approaches

Skeleton-based approaches first estimate 2D/3D keypoints and then model their temporal evolution to recognize actions or interactions [59, 83, 74]. Conceptually, this abstraction can reduce sensitivity to illumination and appearance and yields low-dimensional sequences amenable to graph convolutional networks or temporal models [83, 58]. For violence recognition, pose dynamics may capture characteristic patterns such as punches, kicks, or pushes when keypoints are reliable [43].

In surveillance practice, however, pose pipelines face two fundamental constraints that are amplified in buses: (i) keypoint estimation quality degrades sharply under far-field views, low resolution, and heavy occlusion; (ii) the computational overhead of keypoint extraction can be prohibitive for real-time, multi-stream deployments. Efficiency-focused surveillance work explicitly notes that skeleton extraction entails high computational cost and is often unsuitable for unconstrained surveillance because cameras are not focused on the violent scene, yielding partial bodies and ambiguous joints [35]. In bus interiors, where poles, seat structures, and standing passengers frequently occlude limbs, skeleton signals can become intermittent and systematically biased.

For these reasons, while skeleton-based methods remain valuable as complementary signals when pose quality is sufficient, RGB-based spatio-temporal models generally offer more graceful degradation under crowding and partial visibility. In this thesis, pose-driven representations are therefore treated as auxiliary modalities rather than primary detection channels for onboard violence detection. Finally, Section 4 of [49] emphasizes the growing interest in lightweight architectures tailored for real-time surveillance. These methods prioritize efficiency, often sacrificing some accuracy to achieve low latency and reduced resource consumption.

The work by Huillcen Baca et al. [35] exemplifies this trend, proposing an efficient pipeline for violence recognition designed explicitly for real-time deployment. Such approaches are particularly relevant for early detection, where delayed predictions may negate the practical value of the system.

Across all architectural families, a recurring theme emerges: most deep learning approaches to violence recognition are optimized for clip-level classification and retrospective analysis. As a result, they often fail to address the core requirements of early violence detection in surveillance environments.

Integrating insights from spatio-temporal action detection [77] and the detailed

analysis of violence recognition methods in [49], it becomes evident that future systems must move beyond static classification toward detection-oriented, temporally aware, and computationally efficient models. This observation motivates the more detailed analysis of specific architectural families and training strategies presented in the following subsections.

1.4 Comprehensive Analysis of Deep Learning Approaches for Violence Detection

The analysis of deep learning–based violence detection methods reveals that the majority of existing research converges on a relatively narrow interpretation of the problem, primarily focused on the recognition of physical aggression in video data. Pphysical aggression constitutes the most visually explicit and therefore most commonly addressed form of violence in surveillance-oriented datasets. Actions such as punching, kicking, pushing, or group fighting dominate the literature, largely because they can be more easily identified from visual cues compared to psychological or verbal aggression. This focus, however, introduces a structural bias toward late-stage and overt manifestations of violence, implicitly neglecting the early and often ambiguous phases that precede physical confrontation.

From a methodological standpoint, most violence detection systems frame the task as a binary classification problem at the clip or video level, where short, temporally trimmed sequences are labeled as violent or non-violent. While this formulation simplifies training and evaluation, it obscures the inherently temporal and progressive nature of violent events. In real surveillance scenarios, violence does not occur instantaneously but emerges through a sequence of interactions, posture changes, and escalating behaviors. Treating violence as a static label rather than a temporally evolving process fundamentally limits the capacity of models to support early detection and preventive intervention.

The challenges associated with detecting physical aggression in surveillance footage are multifaceted and extend well beyond model architecture. CCTV videos typically suffer from low spatial resolution, fixed viewpoints, compression artifacts, and poor or variable illumination. Physical aggression often involves rapid movements, partial occlusions, and close proximity between individuals, making it difficult to distinguish violent interactions from benign physical contact, such as crowding or playful behavior. Moreover, surveillance scenes frequently contain multiple actors and overlapping activities, further complicating the extraction of reliable spatio-temporal cues. These visual and environmental constraints significantly degrade the effectiveness of methods that perform well on curated or staged datasets.

Temporal ambiguity represents another fundamental challenge. Violent events rarely exhibit well-defined temporal boundaries, and the transition from non-violent

to violent behavior is often gradual. This ambiguity leads to inconsistent annotations across datasets and undermines the reliability of supervised learning. In many cases, different annotators may disagree on when violence begins or ends, introducing label noise that propagates through the training process. This issue is particularly critical for detection and early detection tasks, where temporal precision is essential. Despite this, most existing datasets provide only coarse temporal annotations or clip-level labels, limiting the development of models capable of fine-grained temporal reasoning.

Dataset design plays a central role in shaping the behavior and perceived performance of violence detection algorithms. Widely used datasets vary significantly in terms of acquisition conditions, video duration, annotation granularity, and realism. Many datasets consist of short clips extracted from movies, sports broadcasts, or staged scenarios, which tend to exaggerate violent actions and present them in visually clean conditions. While such datasets facilitate benchmarking, they fail to capture the complexity and variability of real-world surveillance footage. More realistic datasets, such as those collected from CCTV-like environments, often contain longer, untrimmed videos but provide limited annotation detail and suffer from severe class imbalance. Crucially, none of the commonly used datasets are explicitly designed to support early violence detection, as they lack annotations for pre-violent phases or escalation dynamics.

In response to the length and redundancy of surveillance videos, most violence detection pipelines incorporate mechanisms for selecting relevant frames or temporal segments. Simple approaches rely on uniform sampling or sliding-window extraction of fixed-length clips, which are computationally efficient but risk missing critical moments or diluting discriminative information with irrelevant content. More advanced strategies exploit motion cues, optical flow statistics, or visual saliency to identify potentially informative segments. While these methods can focus computation on dynamic regions, they are sensitive to noise and camera motion and often emphasize frames where violence is already apparent. Learning-based selection mechanisms, including attention models, offer greater flexibility but increase system complexity and are rarely evaluated under real-time constraints. Importantly, the majority of frame and segment selection strategies are designed for retrospective recognition rather than anticipation, limiting their effectiveness for early detection.

The analysis of violence detection algorithms reveals a strong predominance of clip-based approaches, reflecting both dataset design and evaluation practices. Frame-based methods, typically built on 2D CNNs, process each frame independently and aggregate predictions over time. Although computationally efficient, these methods are fundamentally limited by their inability to capture motion dynamics and temporal evolution. Clip-based approaches, including 3D CNNs, CNN RNN hybrids, and two-stream architectures, incorporate short-term temporal information and generally achieve higher accuracy on benchmark datasets. However,

they remain constrained by fixed temporal windows and struggle to generalize to untrimmed video streams. Event-based and detection-oriented approaches, which model violence as a temporally extended process, are conceptually better aligned with surveillance requirements but are significantly less explored in the literature.

A critical trend identified across the reviewed works is the prioritization of classification accuracy over robustness, generalization, and operational viability. High reported accuracies often reflect dataset-specific biases rather than genuine advances in understanding violent behavior. Cross-dataset evaluation is rarely performed, and when it is, performance typically degrades substantially. Moreover, relatively few studies explicitly address real-time constraints, false alarm rates, or system-level considerations, despite their central importance for deployment in public safety applications. Lightweight and efficient architectures are occasionally proposed, but their evaluation is often limited to controlled settings, leaving open questions about their reliability in complex surveillance environments.

Taken together, the literature analysis highlights a substantial gap between the current state of research and the requirements of practical violence detection systems. Early violence detection, in particular, remains largely unaddressed. Most existing methods are reactive, triggering alarms only after physical aggression becomes visually explicit. Bridging this gap requires a fundamental shift in problem formulation, from static clip-level classification toward spatio-temporal detection and anticipation. Such a shift necessitates not only new model architectures, but also new datasets, annotation protocols, and evaluation methodologies explicitly designed to capture the temporal evolution of violent behavior. This observation motivates the methodological choices and contributions presented in the subsequent chapters of this dissertation.

1.5 Action Recognition in CCTV Environments

The majority of advances in action recognition and violence detection have been driven by progress on curated benchmark datasets, often characterized by trimmed videos, controlled viewpoints, and relatively clean visual conditions. However, when these methods are deployed in real-world CCTV environments, their performance frequently degrades, revealing a substantial gap between laboratory settings and operational surveillance scenarios. Understanding this gap is essential for framing violence detection as a practical problem rather than a purely academic one.

CCTV-based action recognition differs fundamentally from generic action recognition along multiple dimensions, including video quality, scene dynamics, temporal structure, and operational constraints. As highlighted across the surveys by Negre et al. [49], Wang et al. [77], and the real-time–oriented study by Huillcen Baca et al. [35], these differences profoundly influence both algorithm design and evaluation methodology.

1.5.1 Characteristics of CCTV Surveillance Data

CCTV footage is typically acquired from fixed cameras operating continuously over long periods. Unlike videos collected for action recognition benchmarks, surveillance streams exhibit low and heterogeneous spatial resolution, strong compression artifacts, limited frame rates, and fixed or elevated viewpoints. In public transportation scenarios, cameras are often mounted at ceiling level, producing top-down or oblique views that distort human appearance and reduce the visibility of discriminative body cues.

Illumination conditions vary widely due to time of day, weather, and artificial lighting, while motion blur and noise are common in low-cost camera systems. Furthermore, surveillance scenes are dominated by background activity, with actions of interest occupying only a small fraction of the temporal axis. This imbalance exacerbates false positives and makes naive application of clip-level classifiers impractical.

In such settings, the visual signatures of actions are weak and context-dependent. Violent and non-violent interactions may appear visually similar when observed from a distance or under occlusion. These characteristics undermine the assumptions underlying many action recognition models trained on high-quality, human-centric datasets.

1.5.2 Limitations of Benchmark-Oriented Action Recognition Models

Most action recognition architectures implicitly assume that the input video contains a single, temporally localized action, centered within a short clip. This assumption is violated in CCTV footage, where actions emerge unpredictably and may overlap in time and space. As a result, models trained on trimmed datasets often exhibit unstable behavior when applied to untrimmed streams, producing fragmented detections or spurious activations.

The reliance on clip-level supervision further limits generalization. Benchmark datasets typically provide precise labels aligned with clip boundaries, whereas surveillance datasets often rely on weak or coarse annotations. This mismatch leads to overfitting to dataset-specific biases, such as scene layout, camera angle, or actor appearance, rather than learning robust representations of human behavior.

Wang et al. [77] explicitly note that the majority of action recognition models are ill-suited for continuous video analysis, motivating the shift toward temporal and spatio-temporal action detection frameworks. In the context of CCTV, this shift is not optional but necessary, as meaningful system behavior depends on the ability to process long streams and reason about temporal structure.

1.5.3 Action Recognition versus Action Detection in CCTV

In surveillance environments, the distinction between action recognition and action detection becomes particularly pronounced. While recognition answers the question of what action is present, detection additionally addresses when and where the action occurs. For CCTV-based violence detection, temporal localization is critical: operators and automated systems must know not only that violence has occurred, but also its duration and evolution.

Spatio-temporal action detection (STAD) offers a principled framework for modeling actions as temporally extended and spatially localized events. However, as discussed in [77], STAD methods are computationally demanding and sensitive to annotation quality, both of which pose challenges in surveillance contexts. Moreover, most STAD benchmarks still differ substantially from real CCTV footage in terms of scene complexity and video quality.

Despite these challenges, detection-oriented formulations align more naturally with the requirements of CCTV systems. They enable reasoning about multiple concurrent actions, tracking of interacting individuals, and integration with downstream components such as multi-camera tracking and alarm generation.

1.5.4 Real-Time Constraints and System-Level Considerations

A defining characteristic of CCTV-based action recognition is the need for real-time or near-real-time operation. Surveillance systems must process continuous video streams with minimal latency, often on resource-constrained hardware. This requirement imposes strict limits on model complexity, memory usage, and preprocessing overhead.

Huillcen Baca et al. [35] emphasize that achieving real-time violence recognition necessitates careful architectural design, including lightweight backbones, efficient temporal modeling, and streamlined pipelines. Models that achieve state-of-the-art accuracy on benchmarks may be unsuitable for deployment if their inference latency exceeds the temporal window required for intervention.

Real-time constraints also interact with early violence detection. A system that detects violence accurately but with significant delay may fail to prevent harm. Consequently, timeliness becomes as important as classification performance, shifting the evaluation focus toward latency-aware metrics and online processing capabilities.

1.5.5 Early Violence Detection in CCTV Environments

Early violence detection represents one of the most challenging and underexplored aspects of CCTV-based action recognition. Unlike retrospective recognition, early

detection requires predicting the likelihood of violence based on partial observations, often before overt physical aggression occurs. This task is inherently uncertain and demands models capable of capturing subtle behavioral cues and temporal trends.

Current datasets and evaluation protocols provide limited support for early detection. Most violence annotations correspond to fully developed events, offering no supervision for pre-violent phases. As a result, models are typically optimized to recognize explicit violence rather than anticipate escalation.

Integrating insights from action detection and anticipation research with the domain-specific suggests that early violence detection should be framed as a spatio-temporal prediction problem on untrimmed streams. Such a framing encourages the development of models that operate incrementally, maintain temporal context, and explicitly trade off prediction confidence against timeliness.

1.5.6 Real-Time Constraints and Practical Challenges

Real-time operation is the defining constraint of CCTV violence detection. From a systems perspective, end-to-end latency aggregates across encoding, decoding, buffering, clip sampling, inference, post-processing, and alert dispatch. Inside buses, where multiple IP cameras stream concurrently to an onboard compute unit, the pipeline must balance throughput and low jitter under constrained power and thermal envelopes. Efficient media pipelines (e.g., zero-copy decoding) and careful batching can improve utilization, but excessive batching risks head-of-line blocking that increases time-to-alert [85, 56].

Model-side constraints interact with these systems issues. Larger temporal windows improve recognition but delay the earliest possible decision; sparse temporal sampling mitigates this by distributing a fixed number of frames across a longer interval [76]. Multi-rate architectures can trigger provisional alerts via a fast branch while refining decisions as more context accrues [24]. Quantization and mixed precision can further reduce compute, but calibration after compression is crucial for setting operational thresholds [37].

An important practical point raised by efficiency-oriented surveillance work is that "real-time capability" is often asserted without standardized measurement. Huillcen Baca *et al.* note the lack of a formal method and adopt a pragmatic protocol: measuring processing time per 30 frames assuming 30 fps as a default for surveillance; under this protocol they report an average latency of 0.072 s to process 30 frames (i.e., roughly one second of video) on a modest GPU-equipped laptop [35]. They further implement a local surveillance prototype that loops over 30-frame buffers, overlays predicted labels, and releases buffers iteratively, providing a deployment-oriented validation beyond offline benchmarking [35].

While the exact numbers are hardware-dependent, the methodological point is central for buses: latency metrics must be reported in a way that maps to

operational constraints (alerts per second, streams per device, thermal limits), not only to FLOPs or parameter counts.

Operationalization introduces evaluation criteria that standard academic metrics rarely capture: false alerts per hour per camera, temporal stability of predictions across sliding windows, localization within the scene, and mean time to detection from incident onset. These criteria become even more important in buses, where driver distraction costs constrain tolerable false positives. In addition, privacy and governance requirements push architectures toward on-board inference with compact, encrypted alerts and short pre/post buffers rather than continuous streaming to control centers [22, 21, 85]. In essence, real-time CCTV violence detection is a joint optimization over representation learning, systems engineering, and governance; for buses, these constraints are the preconditions for safe and responsible deployment.

1.6 Public Datasets for Violent and Aggressive Behavior Detection

Datasets are the substrate of progress in computer vision. For violence detection, dataset choice is particularly consequential because the semantics of "violence" depend on subtle, context-dependent cues that are unevenly represented across domains. Recent review work underscores both the growth and fragmentation of available resources: dozens of datasets and a broad diversity of algorithm inputs and preprocessing strategies, reflecting the absence of a single dominant data standard for violence detection [49]. This diversity matters for buses, where domain shift is often the principal failure mode.

A rigorous treatment distinguishes four axes. First, label granularity ranges from clip-level binary labels to spatio-temporal localization of atomic actions (e.g., AVA bounding boxes) [29]. Second, temporal structure ranges from trimmed clips to untrimmed streams, shifting the task from classification to detection/localization [65]. Third, modality can be visual-only or audio-visual, the latter offering complementary cues at the cost of increased privacy sensitivity and acoustic confounds in vehicles. Fourth, domain determines transferability: movies and sports offer high signal-to-noise exemplars but diverge from CCTV and onboard video statistics [67].

1.6.1 Overview of Existing Datasets

The *Hockey Fight* dataset provides balanced fight vs. non-fight clips from broadcast hockey videos [52]. Its value lies in clear motion patterns and reproducibility, but it is strongly biased: fights typically occupy much of the frame and the visual context is homogeneous. Such biases are explicitly noted in efficiency-oriented surveillance work as reasons why sports data are weak proxies for real surveillance [35].

Movie-derived corpora, including MediaEval violence-scene detection resources, offer shot-level annotations and support multimodal research [18, 17, 61]. However, cinematography and choreography diverge from fixed CCTV viewpoints, limiting transfer to buses.

Surveillance-oriented datasets provide closer alignment. *Violent Flows* emphasizes crowded scenes and global motion statistics [30]. *RWF-2000* offers CCTV-style clips of fights and non-fights [13], but it is curated from online sources and may underrepresent night scenes and unedited CCTV noise; this limitation is also discussed in applied work that contrasts RWF-2000 with truly in-domain surveillance footage [35]. Untrimmed anomaly datasets such as *UCF-Crime* promote weakly supervised temporal localization in long videos [65]. Audio-visual datasets such as *XD-Violence* support multimodal weak supervision, though their deployment relevance depends on audio governance [82].

A practical addition to this landscape is *VioPeru*, introduced as a dataset of real violence and non-violence extracted from municipal CCTV records in Peru. It comprises hundreds of short clips collected from multiple municipalities and includes additional non-violent videos for false-positive analysis, with explicit attention to ethical authorization and dataset legitimacy [35]. The motivation is precisely to validate whether models that perform well on common benchmarks remain effective under real surveillance conditions such as night scenes, varied resolutions, and occlusion [35].

Finally, for this dissertation's bus-specific scope, the *Bus Violence* benchmark targets bus interiors directly [7]. Its camera geometry, occlusion patterns, and motion/illumination dynamics more closely match vehicle-mounted CCTV, providing a more honest estimate of performance in the target domain. The remaining caveat is ecological validity: staged incidents may not fully capture the distribution of genuine aggression.

1.6.2 Limitations for Real-World Deployment

The principal barrier to deployment is not the absence of datasets, but mismatch. Most public corpora favor trimmed clips, binary labels, and salient exemplars to simplify annotation and release, whereas operational systems must handle untrimmed, multi-stream inputs, low signal-to-noise ratios, and asymmetric error costs. This mismatch manifests as brittle time-to-alert behavior and poor calibration when models are exposed to bus-specific noise sources.

Label granularity is a persistent limitation. Binary labels do not capture onset, duration, or spatial locus, even though these properties matter for escalation and evidence retrieval. Fine-grained datasets (e.g., AVA) support localization but differ in camera placement and behavior statistics from buses [29]. Weakly supervised anomaly datasets provide temporal localization but can induce shortcut learning and suffer label noise [65]. The STAD literature reinforces that spatio-temporal

annotations are expensive, motivating label-efficient learning and online detection methods as key future directions [77]—a point that is directly relevant to buses where annotation is both costly and sensitive.

Domain bias compounds these issues. Sports and movies concentrate discriminative motion centrally and present canonical views; CCTV distributes signal sparsely across cluttered scenes and subjects it to compression, glare, and low light. Torralba and Efros' dataset bias observations remain apt: models internalize the peculiarities of their training corpora and generalize poorly outside them [67]. Even CCTV datasets can be biased if collected from edited online sources. VioPeru-style data illustrate the extent of variability in real surveillance—night scenes, occlusion, varying resolutions, and diverse forms of violence beyond fistfights—properties that better match the operational domain [35].

A further limitation is the misalignment between academic metrics and operational requirements. While clip accuracy or framewise AUC are useful, transit agencies care about false alerts per hour, temporal stability, event-level detection latency, and interpretability. Recent reviews categorize challenges in violence detection into groups that explicitly include hardware and real-time constraints, detection/monitoring issues (e.g., occlusion and scale variation), image quality/lighting, and environmental changes [49]. These categories map directly onto bus interiors, where camera position and viewpoint compression, heavy occlusion, and illumination variation are routine.

In sum, no single dataset satisfies all desiderata for bus deployment. A practical strategy is staged: pretrain on large generic corpora, adapt on CCTV-oriented data, fine-tune on bus-specific benchmarks, and—where governance allows—leverage self-supervised learning on unlabeled in-domain streams to reduce residual domain shift [40, 29, 7, 77].

1.7 Research Gaps in the Literature

Despite progress in action recognition and intelligent surveillance, several gaps still limit reliable onboard violence detection in buses. First, a persistent domain gap remains between academic benchmarks and vehicle-mounted CCTV. Bus interiors exhibit fixed elevated viewpoints, strong perspective compression along narrow aisles, frequent occlusions by poles and passengers, and non-stationary backgrounds due to vehicle motion. Reviews of violence detection explicitly list camera position, occlusion, and scale variation among core challenges, alongside illumination variation and non-stationary backgrounds. These factors systematically undermine models trained on non-vehicle domains.

Second, label granularity rarely matches operational needs. Binary labels do not support onset/duration reasoning, while weak localization labels can promote context-driven shortcuts. Although STAD provides a principled framework to

jointly address class, temporal extent, and actor localization, STAD annotation is expensive, motivating label-efficient learning as an open problem. For buses, where privacy and annotation costs are high, the literature lacks consensus on practical protocols that combine minimal supervision, robust localization, and calibrated alerting.

Third, real-time constraints remain under-modeled. Surveys and applied work point out that many proposals optimize accuracy without rigorous, standardized measurement of latency and throughput under realistic surveillance settings [35, 49]. In buses, the true bottleneck is end-to-end time-to-alert across multi-stream pipelines, not only FLOPs. Furthermore, confidence calibration after compression/quantization is underexplored in surveillance despite its importance for threshold setting under asymmetric error costs [37].

Fourth, multi-camera reasoning for onboard environments is not yet mature. While multi-view surveillance has a long history [78], bus cameras are short-baseline, sometimes partially overlapping, and subject to independent exposure/compression dynamics and mild temporal drift. Simple late-fusion rules do not account for per-view reliability or asynchronous streams, and principled cross-view models that incorporate geometry and temporal uncertainty remain scarce.

Finally, the gap between academic metrics and operational outcomes persists. Transit deployments require reporting and optimizing for false alerts per hour, temporal persistence, event-level latency, and interpretable localization. The STAD literature explicitly highlights online real-time STAD as a future direction for surveillance applications, requiring systems to process incoming frames/clips using only history and current data and to report detections as soon as possible [77]. Translating these requirements to buses demands evaluation protocols and datasets that make latency, calibration, and cross-view consistency first-class, alongside privacy-by-design constraints mandated by regulation [22, 21].

Taken together, these gaps motivate the dissertation's focus: bus-specific domain alignment, deployment-oriented efficiency and latency analysis, principled (and privacy-aware) localization consistent with STAD-style objectives, and evaluation protocols tied to operational risk rather than benchmark convenience.

1.8 Objectives and Contributions of This Dissertation

This dissertation responds to the aforementioned gaps by advancing a domain-aligned approach to violence detection that is expressly tailored to bus interiors and to the realities of embedded, multi-camera deployment. The overarching objective is to design, implement, and rigorously validate a real-time system that transforms raw multi-stream CCTV inputs into reliable aggression alerts with bounded latency,

under the legal and operational constraints of public transport. Achieving this objective requires contributions that span representation learning, dataset construction and standardization, systems engineering at the edge, evaluation methodology, and governance-aware design.

The first contribution is a comprehensive, domain-specific synthesis that reinterprets the state of the art in action recognition through the lens of bus surveillance. Rather than presenting models in isolation, the dissertation articulates how architectural biases—including multi-rate temporal pathways, factorized spatio-temporal convolutions, and attention over long-range dependencies—interact with constraints such as occlusion patterns, subject scale variation with aisle distance, camera mounting geometry, and bandwidth-induced compression [69, 24, 23, 5, 2]. This synthesis yields concrete design criteria for selecting and adapting video encoders that can be realized on embedded hardware without forfeiting the temporal cues that disambiguate aggressive interactions.

The second contribution is methodological and concerns data. The work assembles and standardizes a training and validation corpus that integrates surveillance-origin violence datasets with bus-specific material. Public corpora are curated and normalized in frame size, temporal sampling, and clip duration to reduce confounds during training and comparison [13, 65]. In parallel, a laboratory dataset is produced in an instrumented environment that replicates the geometry and camera placement of a bus interior; this dataset augments public resources by introducing in-domain exemplars with controlled illumination regimes, passenger densities, and occlusion patterns that are representative of operating conditions [7]. The standardization protocol and the in-domain augmentation are documented to enable reproducibility and to provide a template for other agencies or researchers to construct compliant, domain-aligned datasets under local policy constraints [71, 22, 21].

The third contribution is a systems architecture for onboard, multi-camera inference that prioritizes end-to-end latency and stability. The architecture integrates efficient video encoders with a zero-copy media pipeline, hardware-accelerated decoding, and batched inference across streams to meet real-time constraints within power and thermal envelopes typical of vehicle deployments [47, 36]. Particular emphasis is placed on the design of the decision layer. Rather than treating streams independently, the system implements a cross-view fusion mechanism that accounts for view-specific reliability, temporal asynchrony, and the spatial layout of the cabin as derived from mounting positions. The fusion policy is calibrated to suppress spurious detections at the periphery and to compensate for confidence decay at increasing subject distance from each camera, addressing a failure mode repeatedly observed in preliminary trials and field tests. The dissertation provides a rigorous account of this fusion logic and analyzes its effect on alert stability relative to single-view baselines [78].

The fourth contribution is an evaluation framework aligned with deployment. Beyond conventional metrics such as precision, recall, F1-score, and frame-level

AUC, the framework reports event-level measures that reflect operational concerns: mean time to detection from incident onset; stability of decisions across overlapping windows; cross-view consistency; and calibrated false alerts per hour per camera under asymmetric error costs appropriate for driver workload. To make these measures comparable, the evaluation protocol fixes clip sampling, latency budget, and alert hysteresis, and it explicitly accounts for post-training quantization when relevant to embedded inference [76, 37]. The framework is applied first in a controlled laboratory setting, then in a live bus deployment, enabling a transparent accounting of the drop in accuracy and confidence that accompanies the transition from lab to field and the identification of environmental regimes in which the system retains acceptable performance.

Field results are analyzed in terms of distance-dependent confidence decay, occlusion-induced misclassifications, illumination-driven distribution shifts, and compression artifacts. These analyses are linked to design choices in representation learning, sampling, and fusion, and they motivate a set of targeted improvements that include fine-tuning on in-domain data, self-supervised pretraining on unlabeled onboard streams subject to governance constraints, and adaptive fusion that modulates per-view weights using quality estimators derived from exposure and motion statistics [82]. Rather than treating these as generic suggestions, the dissertation quantifies their expected impact within the latency and compute budgets of the onboard platform. The system architecture and evaluation protocols are developed with explicit reference to privacy-by-design principles: on-vehicle inference with minimized data egress, ephemeral buffering around incidents, encryption in transit and at rest for short alert snippets, strict access control, and auditable retention policies, all aligned with regulatory guidance on video devices [71, 22, 21]. The dissertation argues that such constraints are not merely externalities but shape algorithmic design choices and data strategy, and it offers a set of reproducible configurations and documentation templates to facilitate responsible adoption.

Collectively, these contributions move the field toward robust, real-time violence detection in bus interiors by combining domain-aligned representation learning with embedded systems engineering, principled multi-camera fusion, and deployment-oriented evaluation. The result is a coherent approach that narrows the gap between laboratory benchmarks and the operational realities of public transportation, while satisfying the legal and social constraints that appropriately govern surveillance technologies [54, 6, 47].

1.9 Thesis Outline

This dissertation is organized to move from problem framing and scientific background to methodology and system design, and finally to empirical validation in both laboratory and in-service conditions. The narrative progression is deliberate:

each chapter resolves specific uncertainties that emerge in bus interiors when violence detection is treated not as an offline recognition task but as a real-time, multi-camera system problem conducted under operational and governance constraints. In doing so, the dissertation aims to bridge the gap between the general literature on intelligent surveillance and action recognition and the practicalities of deployment on public transport vehicles [54].

Chapter 2 presents the methodology and the overall system architecture. It formalizes the end-to-end processing stack, beginning with the acquisition of six synchronized IP-camera streams onboard the vehicle and proceeding through decoding, temporal sampling, inference, and decision logic, all hosted on an embedded server engineered for sustained real-time operation. The chapter details the spatial and temporal standardization applied to publicly available corpora relevant to aggression and violence, clarifying how heterogeneous sources are harmonized in resolution, frame rate, clip duration, and label semantics in order to produce comparable training and validation sets [13, 65]. It then motivates the construction of a laboratory dataset that reproduces the geometry and occlusion patterns of a bus cabin and explains how this in-domain resource complements public datasets by providing controlled yet realistic exemplars of interpersonal aggression in the target environment [7]. The chapter frames model selection in terms of architectural biases that the literature has shown to be effective for spatio–temporal reasoning—factorized 3D convolutions, multi-rate temporal pathways, and efficient expanded designs—and discusses how these families can be adapted to bus interiors without exceeding the constraints of onboard compute [69, 24, 23]. It describes the training and validation protocol, including pretraining and fine-tuning choices on generic action datasets where appropriate [40, 29], and it specifies the evaluation criteria used throughout the dissertation, which couple conventional accuracy measures with deployment-facing indicators such as time-to-first-alert and false alerts per vehicle-hour. Because the feasibility of multi-stream inference depends as much on systems engineering as on representation learning, the chapter also documents the media pipeline, batching and buffering policies, and optimization techniques—mixed precision, quantization-aware compilation, and zero-copy transport—used to sustain low latency and thermal stability under realistic load [47, 36]. Finally, it justifies the choice of a late fusion strategy across cameras, reflecting the partial overlap and asynchronous behavior typical of vehicle interiors, and positions this design within the broader multi-camera surveillance literature [78].

Chapter 3 reports the experimental results. It begins with laboratory evaluations that isolate model behavior under controlled conditions, comparing representative spatio–temporal architectures trained on the standardized dataset mixture and on the laboratory bus corpus. The analysis characterizes precision, recall, and calibration as functions of subject distance, crowd density, and occlusion severity, thereby linking performance differences to the inductive biases discussed in Chapter 2. The chapter then transitions to the field deployment on an operating bus,

where the full system processes six streams in real time. Here the focus shifts to end-to-end behavior: the latency budget from frame acquisition to alert issuance, the stability of decisions across overlapping clips, the robustness of cross-view fusion when streams are imperfectly synchronized, and the rate of false alerts per vehicle-hour under thresholds appropriate for driver workload and dispatch triage. Results are interpreted against the backdrop of the constraints identified in Chapter 1 and the dataset biases highlighted in the literature, including the tendency of models trained on generic corpora to lose confidence as subjects recede from the camera or when illumination drops during evening service [67, 13]. The chapter includes an error analysis that enumerates failure modes observed in practice—distance-dependent confidence decay, occlusion-induced misclassifications, and compression artefacts—and it quantifies the marginal gains achieved by multi-camera late fusion relative to single-stream baselines, placing these findings in the context of prior multi-view work [78]. Where relevant, the chapter also revisits the effectiveness of pretraining on large action datasets and discusses the residual gap that might be reduced through self-supervised pretraining on privacy-compliant, in-domain footage [66].

Chapter 4 concludes the dissertation by synthesizing scientific and technological contributions, limitations, and future directions. It distills the lessons learned about how architectural choices interact with the geometry and social dynamics of bus interiors, and it articulates the boundary conditions under which the current generation of models remains effective. The chapter reflects on the implications of the evaluation protocol and argues for deployment-oriented metrics as first-class outcomes in violence detection research, complementing conventional accuracy scores with measures that track notification burden and time-to-intervention. It then outlines next steps that logically extend from the empirical findings: reliability-weighted cross-view fusion and temporal voting strategies tailored to the partial overlap and asynchronous behavior of onboard cameras; domain adaptation and self-supervised pretraining to reduce distribution shift between public datasets and bus interiors; and extensions to the governance-aware system design, with on-vehicle inference, minimal data egress, and strengthened privacy controls aligned with transport-agency policies [71, 47]. The chapter closes by discussing pathways to scale, including operational integration and monitoring, and by describing how the resources produced in this work—dataset protocols, preprocessing pipelines, evaluation scripts, and system configurations—can be reused to accelerate further research and responsible deployment in public transport.

The dissertation is accompanied by appendices that provide materials required for reproducibility and audit. They include detailed descriptions of the laboratory dataset design, annotation guidelines, and consent and safety procedures; hyperparameter tables and training logs for each model configuration; extended quantitative results such as full confusion matrices and calibration plots; and engineering documentation for the onboard deployment, including component specifications,

thermal characterization, and software configurations. Where agency policy permits, the appendices also record the exact preprocessing code and evaluation scripts used to produce the figures and tables in Chapter 3. Together with the chapter structure summarized above, these materials are intended to make the scientific claims testable and the engineering results verifiable, closing the loop between academic method and fielded system in the domain of violence detection for buses [54, 7].

1.10 List of Scientific Publications

1.10.1 Journals

Mariano G. Paganelli, Marco Gallo, Paolo R. Massenio, and David Naso. Integrated passenger flow analysis and street-level monitoring for public transport management using deep learning and IoT. *IEEE Access*, 13:143401–143413, 2025.

Marco Gallo, Mariano G. Paganelli, Paolo R. Massenio, and David Naso. Real-time Violence Detection in Urban Bus Environments. *IEEE Access*, 14:55075-55089, 2026.

1.10.2 Conference Proceedings

Mariano G. Paganelli, Marco Gallo, Paolo R. Massenio, and David Naso. Enhancing Public Transport Management with Deep Learning and IoT-Based Monitoring, 2025 13th International Conference on Traffic and logistic Engineering (ICTLE-25) Macau, Cina

M. Gallo, M. G. Paganelli, D. Naso, and P. R. Massenio, "AI-Enabled Bus Surveillance: Real-Time Passenger Counting and Violence Detection," presented at the Int. Conf. on Artificial Intelligence and Smart Environments (ICAISE'25), Hammamet, Tunisia, 2025

Chapter 2

Methodology and System Architecture

2.1 Overview of the Proposed Approach

In summary, the proposed approach integrates temporal sampling, spatio–temporal representation learning via a deep learning model, multiview reliability-weighted fusion, and a hysteresis-based alert policy within an embedded, privacy-aware system designed for buses. By specifying the sampler, backbone and head, fusion, and policy in operational terms and by committing to deployment-oriented evaluation, the method narrows the longstanding gap between benchmark performance and the requirements of real-time violence detection in public transport [54, 47]. The proposed system approaches violence detection in buses as a real-time, multi-camera perception problem executed entirely on the onboard embedded platform. Inference, cross-view fusion, and the driver-facing alert are performed locally; connectivity is required only for optionally dispatching a compact, encrypted notification to the control center. When the backhaul is unavailable, the system continues operating in local-only mode and seals a short pre/post-incident buffer for opportunistic upload upon reconnection, thereby meeting latency requirements while minimizing data egress in line with governance constraints. Figure 2.1 provides a schematic overview of the proposed real-time violence detection pipeline, highlighting the interaction between temporal sampling, inference, multiview fusion, and alerting.

Methodologically, the approach integrates representation learning and systems engineering: temporal sampling and the *deep learning model* design (a spatio–temporal backbone as feature extractor with a task-specific classification head) are chosen to preserve the high-frequency motion signatures that characterize aggressive interactions while respecting thermal and power envelopes; decoding, buffering, batching, and fusion are co-designed to control end-to-end delay [24, 76].

For each camera $c \in \{1, \ldots, C\}$, we denote the RGB frame sequence with spatial

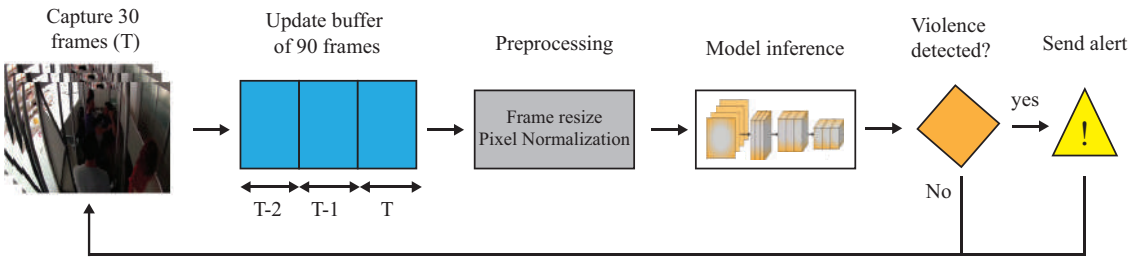


Figure 2.1: *Schematic representation of the real-time violence detection algorithm.*

resolution $H \times W$ and nominal frame rate f_v as

$$\{I_t^{(c)}\}_{t \geq 0} \in \mathbb{R}^{H \times W \times 3}. \tag{2.1}$$

In this formulation, $I_t^{(c)}$ represents the frame acquired by camera c at discrete time index t, where time is measured in frame ticks at cadence f_v. The parameters H and W correspond to the sensor-aligned image height and width in pixels, while the final dimension accounts for the three RGB color channels. Under stable acquisition conditions, the index t increases by one for each decoded frame, so that one second of video corresponds approximately to f_v increments of t.

A temporal sampler $S(\cdot)$ converts a buffer of decoded frames into a fixed-length clip of T frames at spatial resolution $H' \times W'$, spanning a temporal window of length L. Formally, the sampled clip ending at time t is defined as

$$X_{t,L}^{(c)} = S\Big(\{I_{t-L+1}^{(c)}, \ldots, I_t^{(c)}\}\Big) \in \mathbb{R}^{T \times H' \times W' \times 3}. \tag{2.2}$$

Here, the sampler S selects and resizes frames from the most recent L decoded frames of camera c. The number of frames actually fed to the network is T, with $T \leq L$ when sparse temporal sampling is employed. The dimensions H' and W' denote the network input resolution after spatial preprocessing (e.g., resizing, cropping, or letterboxing). A uniform sampler corresponds to the special case $L = T$, whereas a sparse sampler distributes the T frames across a larger temporal span L to capture longer-term motion dynamics without a proportional increase in computational cost [76].

Each clip $X_{t,L}^{(c)}$ is processed independently by a spatio–temporal backbone ϕ_θ, which acts as a deep feature extractor. The resulting representation is given by

$$z_t^{(c)} = \phi_\theta\Big(X_{t,L}^{(c)}\Big) \in \mathbb{R}^d. \tag{2.3}$$

The mapping ϕ_θ, parameterized by learnable weights θ, encodes joint motion and appearance information into a d-dimensional feature vector. Typical choices for ϕ_θ include architectures such as X3D, SlowFast-50, or R(2+1)D. The dimensionality d depends on the specific architecture and its final pooling stage. Importantly, this

backbone operates entirely on decoded video data and is decoupled from the video codec pipeline, which is handled upstream by the hardware decoder.

A task-specific classification head $\psi(\cdot)$ maps the extracted features to a per-clip posterior probability for the target event (i.e., aggressive or violent interaction). This operation is expressed as

$$p_t^{(c)} = \psi\Big(z_t^{(c)}\Big) \in [0,1]. \tag{2.4}$$

In practice, ψ is typically implemented as a linear layer followed by a sigmoid activation in the binary detection setting. The resulting score $p_t^{(c)}$ is interpreted as the model's confidence that the clip acquired by camera c and ending at time t contains an aggression event.

Due to frequent occlusions, viewpoint changes, and distance-dependent scale variations in onboard bus environments, single-view predictions may be unreliable. To improve robustness, evidence is aggregated both temporally and across multiple camera views using a temporal–multiview fusion operator $g(\cdot)$. Let the set of decision instants included in a temporal window of width W_t be defined as

$$\mathcal{T}_t = \{\, t - k\Delta \mid k = 0, \ldots, K \,\}, \tag{2.5}$$

and let the set of available cameras be

$$\mathcal{C} = \{\, 1, \ldots, C \,\}. \tag{2.6}$$

The fused confidence score at time t is then computed as

$$\widehat{p}_t = g\Big(\{\, p_\tau^{(c)} \,\}_{c\in\mathcal{C},\, \tau\in\mathcal{T}_t} \;;\; \{\, w_\tau^{(c)} \,\}\Big) \in [0,1]. \tag{2.7}$$

In this formulation, Δ denotes the decision stride (in frames) between successive fusion updates, while K determines the number of past decision instants included in the aggregation window, such that the effective temporal width satisfies $W_t \approx K\Delta$. The weights $w_\tau^{(c)} \geq 0$ modulate the contribution of each camera c at time τ based on low-cost quality indicators, including exposure statistics, blur proxies derived from inter-frame gradients, encoder-level information (e.g., quantization parameters or packet loss), and estimates of subject scale within the field of view. For numerical stability and interpretability, the weights may be normalized at each update so that

$$\sum_{c\in\mathcal{C}} \sum_{\tau\in\mathcal{T}_t} w_\tau^{(c)} = 1. \tag{2.8}$$

The operator $g(\cdot)$ is designed to be monotone and associative (for instance, a reliability-weighted average optionally combined with a max-hold across the last few updates) so that it admits streaming implementation and incremental updates without waiting for strict cross-camera synchronization.

Turning fused scores into operationally meaningful alerts requires an alerting policy h that enforces temporal persistence and hysteresis. The policy maps the score process into a binary alert state

$$A_t = h\Big(\{\,\widehat{p}_\tau\,\}_{\tau\in\mathcal{T}_t}\;;\;\tau_{\mathrm{on}},\,\tau_{\mathrm{off}},\,T_{\mathrm{hold}}\Big) \in \{0{,}1\}. \tag{2.9}$$

In this formulation, $A_t = 1$ indicates an active alert at time t. The parameter $\tau_{\mathrm{on}} \in (0{,}1)$ is the raise threshold for the fused score, while $\tau_{\mathrm{off}} \in (0{,}1)$ with $\tau_{\mathrm{off}} < \tau_{\mathrm{on}}$ is the clear threshold; their asymmetry implements hysteresis and prevents oscillation near decision boundaries. The parameter $T_{\mathrm{hold}} \geq 0$ is a minimum dwell time (expressed in decision steps of size Δ or, equivalently, in seconds) that requires persistence of evidence before raising or clearing the alert. The policy may include a cross-view consistency test that demands contemporaneous support from at least two distinct cameras when fields of view overlap, thus leveraging geometry to suppress spurious reflections and local motion bursts [78]. Driver notifications are emitted locally with no backhaul dependency; only the optional control-center dispatch depends on network availability and is designed as a store-and-forward channel.

The timing properties of the pipeline are captured by a latency budget that decomposes end-to-end delay from photon to alert as

$$\ell_{\mathrm{tot}} = \ell_{\mathrm{acq}} + \ell_{\mathrm{net}} + \ell_{\mathrm{dec}} + \ell_{\mathrm{buf}} + \ell_{\mathrm{inf}} + \ell_{\mathrm{fuse}} + \ell_{\mathrm{alert}}. \tag{2.10}$$

In this decomposition, ℓ_{acq} covers camera exposure and sensor readout; ℓ_{net} is the transport delay over the onboard network from camera to server; ℓ_{dec} is hardware-accelerated decoding time for compressed video; ℓ_{buf} is the buffering delay to assemble the T frames of the next clip under the chosen sampler; ℓ_{inf} is the deep model inference time per (micro-)batch of clips; ℓ_{fuse} is the multiview aggregation and policy evaluation time; ℓ_{alert} is any remaining decision/dispatch overhead (UI update, CAN or dashboard message, and optional packaging of a short encrypted pre/post snippet). All terms are measured in milliseconds and reported as medians with jitter bands in Chapter 3. The design aims to minimize $\ell_{\mathrm{buf}} + \ell_{\mathrm{inf}} + \ell_{\mathrm{fuse}}$ by combining sparse temporal sampling, micro-batch inference across streams, and streaming fusion that does not await synchronized clip boundaries when unnecessary [76, 24]. In embedded environments, the latency budget is shaped by thermal ceilings and sustained power, motivating mixed-precision inference and integer quantization with post-quantization calibration to preserve the mapping between scores and probabilities used by the hysteresis policy [37, 47].

Generalization across conditions is approached by a data strategy that combines pretraining and in-domain tuning. The deep learning model is initialized from large action corpora to endow early layers with generic motion–appearance sensitivities [40, 10]. Fine-tuning proceeds on CCTV-origin datasets with surveillance artefacts to reduce the domain gap, and, where policy allows, self-supervised pretraining on

unlabeled onboard footage is used to align representations to the specific geometry and illumination cycles of buses without expanding the footprint of personal data beyond the vehicle [13, 66]. The in-domain component follows privacy-by-design principles: unlabeled frames are processed on-vehicle; no identifiable video leaves the bus during self-supervised stages; only model parameters and non-personal metadata are collected for fleet-wide improvements, aligning with agency policies and supervisory guidance [71].

Cross-view fusion reflects the cabin's spatial structure. Camera placement yields partially overlapping views along the aisle; mounting heights and focal lengths create heterogeneous scale–resolution trade-offs. The aggregator g incorporates view-specific priors via weights that depend on estimated subject size and region-of-interest occupancy, so that near-field cameras dominate decisions for events occurring within their high-fidelity zone while far-field cameras contribute persistence evidence. When calibration is available, the policy can enforce geometric consistency by requiring that hypothesized motion regions project to physically plausible loci in neighboring views; when it is not, late fusion with reliability weights remains effective and is simpler to implement robustly in an embedded pipeline where per-stream jitter is unavoidable [78]. In both cases, fusion is designed to be commutative and streaming so that asynchronies do not compound delay.

The evaluation protocol is defined to mirror deployment objectives. In addition to conventional metrics such as precision, recall, F1-score and framewise AUC on curated test sets, streaming evaluations report mean time to first alert from incident onset, alert persistence under sliding windows, cross-view agreement rates, and false alerts per vehicle-hour at operational thresholds. Reporting includes calibration curves before and after quantization and stratifies performance by subject distance and illumination regime so that degradations at the tails of the distribution are visible. Laboratory experiments use standardized data with consistent resolution and frame rates to isolate representation effects, while field trials on an operating bus expose the full pipeline to transport-induced noise sources, validating that the latency budget and hysteresis controls remain within acceptable limits [47, 36, 7].

2.2 Hardware Architecture

The hardware architecture described in this chapter is the exact configuration engineered for the field pilot and was finalized inside a full-scale laboratory simulator. The simulator is a 1:1 replica of the target bus interior constructed from measurement campaigns (*rilievi*) to reproduce aisle geometry, seat layout, ceiling height, handrail and pole placement, and door clearances. A structural frame supports ceiling panels and mounting rails at the same heights and offsets as the vehicle, enabling faithful reproduction of occlusions, perspective compression, and subject scale along the longitudinal axis. Within this environment, we iterated camera

Figure 2.2: *Laboratory simulator (full-scale* 1:1*) reproducing the target bus interior (aisle, seats, poles, doors).*

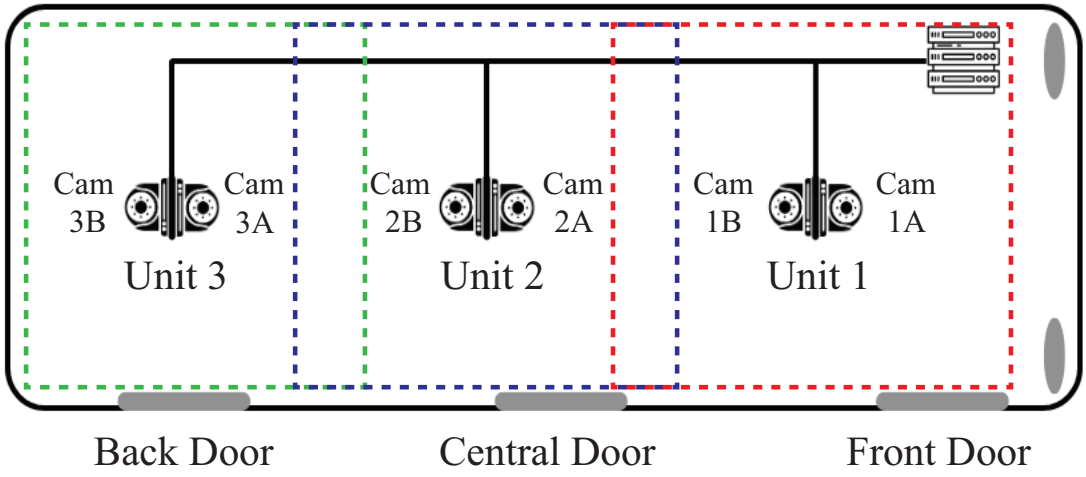


Figure 2.3: *Simulator plant layout: three camera units (Unit1, Unit2, Unit3), work areas with partial overlap (A: front, B: mid, C: rear), network/power topology, and server mount position.*

placement, optics, network topology, power distribution, and server mounting, and we validated coverage, pixel densities on subject, overlapping views, latency headroom, and thermal margins before transferring the configuration onto the bus.

Figure 2.2 shows the simulator; Figure 2.3 presents the plant layout with camera units and work areas (A: front, B: mid, C: rear); Figure 2.4 depicts the six camera footprints and their overlaps within the simulator. These artifacts directly informed the installation on the pilot vehicle.

Figure 2.4: *Top-down footprints of the six cameras in the simulator with overlap corridors across areas A (front), B (mid), and C (rear).*

2.2.1 Embedded Server Configuration

The embedded server hosts the deep learning stack, including the spatio–temporal backbone and classification head, hardware-accelerated video decoding, temporal sampling, multiview fusion, and the alert policy. The unit operates from the vehicle 24 V rail through isolated DC–DC conversion stages and is enclosed in a ruggedized chassis mounted behind an access panel above the driver, where both airflow and maintenance access are available. The design objective is to sustain six concurrent video streams while preserving the latency budget defined in Section 2.1, and maintaining power consumption and operating temperature within vehicle constraints [47, 36].

The aggregate electrical power budget of the surveillance stack is modeled as

$$P_{\text{tot}} = P_{\text{srv}} + \sum_{c=1}^{C} P_{\text{cam}}^{(c)} + P_{\text{PoE}} + P_{\text{conv}} + P_{\text{marg}}, \tag{2.11}$$

where P_{tot} denotes the total power draw (in watts); P_{srv} is the average power consumption of the embedded compute unit (CPU, accelerator/GPU, memory, and storage); $P_{\text{cam}}^{(c)}$ is the power consumption of camera c (including PoE delivery when sourced locally); P_{PoE} represents the overhead associated with PoE injection and switching; P_{conv} accounts for conversion losses in the 24 V DC–DC stages; and P_{marg} provides margin for transients, aging, and supply tolerances. The resulting sum is maintained below the fused circuit rating under all operating modes.

Thermal viability was assessed in simulation using the steady-state approximation

$$T_{\mathrm{case}} = T_{\mathrm{amb}} + \Theta_{\mathrm{JA}}\, P_{\mathrm{srv}}, \tag{2.12}$$

where T_{case} is the steady-state enclosure temperature of the compute module, T_{amb} is the ambient temperature within the installation compartment, and Θ_{JA} denotes the effective junction-to-ambient thermal resistance (in °C/W). The value of Θ_{JA} was estimated empirically by logging T_{case} under stepped computational loads, and the resulting measurements were used to dimension heat sinks and airflow so as to avoid thermal throttling at the highest expected ambient temperatures along the route.

Memory provisioning balances decoded video surfaces, model parameters and activations, temporal buffers, and operating system overhead:

$$M_{\mathrm{tot}} \;\geq\; M_{\mathrm{model}} + M_{\mathrm{act}} + M_{\mathrm{buf}} + M_{\mathrm{dec}} + M_{\mathrm{OS}}. \tag{2.13}$$

Here, M_{tot} denotes the total available RAM; M_{model} the memory required for the backbone and classification head parameters; M_{act} the peak activation footprint during inference; M_{buf} the memory allocated to clip and fusion ring buffers; M_{dec} the decode surfaces allocated across concurrent video sessions; and M_{OS} the operating system and logging overhead.

A conservative bound for the activation memory scales with the input tensor dimensions and the effective micro-batch size across streams:

$$M_{\mathrm{act}} \;\approx\; B \cdot T \cdot H' \cdot W' \cdot C_{\mathrm{in}} \cdot b, \tag{2.14}$$

where B is the effective micro-batch size (number of clips processed concurrently), T is the number of frames per clip, $H' \times W'$ is the network input resolution, $C_{\mathrm{in}}{=}3$ for RGB input, and b is the number of bytes per element (e.g., $b{=}2$ for FP16 or $b{=}1$ for INT8 quantized inference).

Throughput targets were validated in simulation by co-running six video streams and measuring decoding and inference headroom. The decoder capacity constraint is expressed as

$$\sum_{c=1}^{C} R_{\mathrm{dec}}^{(c)} \;\leq\; R_{\mathrm{dec}}^{\max}, \tag{2.15}$$

where $R_{\mathrm{dec}}^{(c)}$ is the per-stream decode throughput (frames/s) and $R_{\mathrm{dec}}^{\max}$ is the aggregate throughput of the hardware video decoder.

For inference, given a per-clip inference time τ_{inf} and a degree of parallelism Π (number of concurrent inference engines or streams), the admissible micro-batch size for a decision stride Δ satisfies

$$B \;\leq\; \left\lfloor \frac{\Pi \cdot \Delta}{\tau_{\mathrm{inf}}} \right\rfloor. \tag{2.16}$$

Mixed-precision inference and post-training quantization were tuned in the simulator until both the decoding constraint in (2.15) and the inference constraint in (2.16) held with margin, and the resulting configuration was subsequently deployed on the vehicle platform [24, 37].

2.2.2 Multi-Camera Setup with Six IP Cameras

The camera system is organized into three units, each comprising two fixed cameras with partially overlapping fields of view:

$$\text{Unit1} = \{\, c_1,\, c_2 \,\}, \qquad \text{Unit2} = \{\, c_3,\, c_4 \,\}, \qquad \text{Unit3} = \{\, c_5,\, c_6 \,\}. \tag{2.17}$$

Here, c_i indexes the six cameras. Unit 1 primarily covers the front door and the aisle adjacent to the driver; Unit 2 covers the mid-aisle; Unit 3 covers the rear door and back aisle. Coverage areas are designed with deliberate overlaps across unit boundaries to support cross-view confirmation and to mitigate occlusions.

Work areas are defined on the simulator floor plane $\mathcal{F}$ as three polygonal regions:

$$\mathcal{A} \subset \mathcal{F} \quad \text{(front)}, \qquad \mathcal{B} \subset \mathcal{F} \quad \text{(mid)}, \qquad \mathcal{C} \subset \mathcal{F} \quad \text{(rear)}. \tag{2.18}$$

These polygons were traced from the 1:1 vehicle plan and correspond to the zones illustrated in Fig. 2.3. By construction,

$$\mathcal{A} \cap \mathcal{B} \neq \emptyset, \qquad \mathcal{B} \cap \mathcal{C} \neq \emptyset, \qquad \mathcal{A} \cap \mathcal{C} = \emptyset, \tag{2.19}$$

ensuring that adjacent areas share an overlap corridor while the front and rear regions remain disjoint.

Each camera delivers a compressed IP video stream with the following nominal parameters:

$$\theta_h = 160^\circ, \qquad W_c = 1080, \qquad H_c = 720, \qquad f_v^{(c)} = 30 \text{ fps}. \tag{2.20}$$

Here, θ_h denotes the horizontal field of view, $W_c \times H_c$ the encoded spatial resolution, and $f_v^{(c)}$ the nominal frame rate. The very wide horizontal field of view maximizes cabin coverage and overlap; lens dewarping and geometric rectification are applied downstream in the software pipeline prior to model ingestion. For completeness, the rectilinear vertical field of view implied by the aspect ratio $\rho = H_c/W_c$ is given by

$$\theta_v = 2 \arctan\Big(\rho \, \tan(\theta_h/2) \Big), \tag{2.21}$$

which was used to estimate vertical pixel densities in the pixel-per-meter (PPM) analysis reported below.

Top-down coverage footprints $\Omega_c \subset \mathcal{F}$ for each camera c were computed from measured mounting positions and optical parameters (Fig. 2.4). The per-point coverage redundancy over the floor plane is defined as

$$N(x) = \sum_{c=1}^{C} \mathbf{1}\Big[\, x \in \Omega_c \,\Big], \qquad x \in \mathcal{F}, \tag{2.22}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function. The design enforces the constraints

$$N(x) \geq 2 \quad \text{for all } x \in (\mathcal{A} \cap \mathcal{B}) \cup (\mathcal{B} \cap \mathcal{C}), \tag{2.23}$$

and

$$N(x) \geq 1 \quad \text{for all } x \in \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}, \tag{2.24}$$

so that overlap corridors guarantee cross-view confirmation for the alert policy, while single-view coverage remains continuous throughout the passenger cabin.

For pixel-per-meter profiling on the floor plane, a camera with horizontal field of view θ_h and horizontal resolution W_c subtends, at longitudinal distance D, a width

$$W_{\mathrm{FOV}}(D) = 2D \tan\!\left(\frac{\theta_h}{2}\right), \tag{2.25}$$

yielding a horizontal pixel density

$$\mathrm{PPM}(D) = \frac{W_c}{W_{\mathrm{FOV}}(D)} = \frac{W_c}{2D \tan(\theta_h/2)}. \tag{2.26}$$

In these expressions, D denotes the floor-projected distance from the camera nadir, while θ_h and W_c are fixed by the lens and encoder configuration. The function $\mathrm{PPM}(D)$ was validated empirically in the simulator by translating an ArUco/checkerboard target along the aisle and verifying that the torso width of an adult subject remains above a minimum pixel threshold across critical interaction zones. Final lens selection and mounting geometry were thus chosen to preserve effective subject scale in regions where aggressive interactions most frequently occur.

To guide bandwidth provisioning, the nominal per-stream bitrate was modeled as

$$B_c = \eta \cdot H_c \cdot W_c \cdot f_v^{(c)} \cdot \kappa(QP, G), \tag{2.27}$$

where η captures scene entropy (e.g., increased motion during interactions) and $\kappa(QP, G)$ models codec efficiency as a function of the quantization parameter QP and GOP structure G. The aggregate load $\sum_c B_c$ was used to dimension the PoE switch and server network interface, with additional margins for transient bitrate peaks coincident with high-motion events.

2.2.3 Synchronization and Real-Time Acquisition Requirements

Real-time multiview fusion presumes bounded temporal misalignment across video streams. In the simulator, we validated a lightweight synchronization scheme in which each camera timestamps frames at capture time, while the server estimates an affine mapping from the camera clock to the server monotonic clock. For camera c, the time alignment model is

$$t_{\mathrm{srv}} = \alpha_c \, t_{\mathrm{cam}}^{(c)} + \beta_c, \tag{2.28}$$

where $t_{\mathrm{cam}}^{(c)}$ denotes the camera-provided timestamp, t_{srv} is the server-side monotonic clock, α_c represents the relative clock rate (skew), and β_c is the offset. The parameters (α_c, β_c) are estimated via robust regression over periodically transmitted synchronization beacons and are subsequently applied to align frames within the temporal fusion window.

The residual alignment error over the most recent estimation interval $\mathcal{T}$ is bounded as

$$\varepsilon_{\mathrm{sync}}^{(c)} = \max_{\tau \in \mathcal{T}} \left| \alpha_c \, t_{\mathrm{cam}}^{(c)}(\tau) + \beta_c - t_{\mathrm{srv}}(\tau) \right|. \tag{2.29}$$

Cross-view coincidences are accepted when aligned timestamps differ by at most $\pm\delta$, where δ is chosen relative to the decision stride Δ defined in Section 2.1. The value of δ was swept in simulation to identify the smallest admissible tolerance that avoided missed coincidences under realistic clock jitter and network variability.

Acquisition delay and its variability were characterized by decomposing the one-way per-stream latency as

$$D_{\mathrm{acq}}^{(c)} = d_{\mathrm{cap}}^{(c)} + d_{\mathrm{enc}}^{(c)} + d_{\mathrm{net}}^{(c)} + d_{\mathrm{dec}}^{(c)}, \tag{2.30}$$

where $d_{\mathrm{cap}}^{(c)}$ accounts for sensor exposure and readout, $d_{\mathrm{enc}}^{(c)}$ captures encoder and GOP buffering delay, $d_{\mathrm{net}}^{(c)}$ represents network transmission and queuing latency, and $d_{\mathrm{dec}}^{(c)}$ denotes hardware decoding time on the server. Timestamp probes at camera egress and server ingress were instrumented to estimate each component.

To satisfy the end-to-end latency budget, both the expected acquisition delay and a jitter proxy were required to remain within the acquisition allotment Λ_{acq}:

$$\mathbb{E}\left[D_{\mathrm{acq}}^{(c)}\right] + \sigma\left(D_{\mathrm{acq}}^{(c)}\right) \;\leq\; \Lambda_{\mathrm{acq}}. \tag{2.31}$$

In addition, a bounded frame-drop condition was enforced for each stream,

$$\rho_{\mathrm{drop}}^{(c)} \leq \rho_{\max}, \tag{2.32}$$

where $\rho_{\mathrm{drop}}^{(c)}$ denotes the fraction of lost frames and $\rho_{\max}$ is the maximum tolerable drop rate before temporal sampling and fusion stability degrade.

Stress tests conducted in the simulator injected network congestion and encoder complexity spikes to validate compliance with the constraints in (2.31) and (2.32) prior to deploying identical configurations on the vehicle. When supported by the hardware, time discipline mechanisms within the surveillance VLAN can further reduce $\varepsilon_{\mathrm{sync}}^{(c)}$; however, the affine correction model in (2.28) was sufficient in our trials. Importantly, the synchronization layer degrades gracefully: if alignment quality deteriorates, the alert policy increases temporal persistence and relaxes coincidence requirements, thereby maintaining stable driver alerts regardless of transient backhaul conditions [78, 47].

Chapter 3

Dataset Development

3.1 Data Structure of Datasets

The initial dataset was composed by integrating the following five sources:

1. RWF-2000 [13]: a dataset composed of 2,000 short video clips (5 s at 30 fps) evenly split between Fight and No Fight actions, captured by real-world surveillance cameras;

2. UCF-Crime [65]: a dataset of 400 untrimmed real-world surveillance videos (10-120 s, up to 3600 s, 20-60 fps), containing a variety of violent and non-violent actions distributed into Abuse, Arrest, Assault, Fighting, and Normal classes;

3. Smart-City CCTV Violence Detection Dataset (SCVD) [1]: a dataset built for violence detection scenarios in smart-city public surveillance contexts made of 481 videos (3-10 s, 20-30 fps) including Normal, Violence, and Weaponized classes;

4. Bus Violence Dataset [7]: 1,400 clips (1 s, 30 fps) of violent and non-violent scenes recorded on real buses using dome surveillance cameras;

3.1.1 RWF-2000

The RWF-2000 (Real-World Fighting) dataset [13] is a large-scale video collection specifically designed for violence detection in realistic surveillance scenarios. It was introduced to address several limitations of previous benchmarks, such as small scale, low diversity, and the lack of truly real-world footage, which often relied on acted scenes or movie clips. All videos in RWF-2000 are sourced from operational CCTV systems and depict unconstrained, in-the-wild situations rather than staged actions or studio-quality content.Representative examples from the RWF-2000 and Bus Violence datasets are shown in Figures 3.1 and 3.6, respectively.

Figure 3.1: *[13] Gallery of the RWF-2000 database.*

The dataset is constructed by crawling the YouTube platform using a set of violence-related keywords (e.g., "real fights", "violence under surveillance", "violent events") in multiple languages. From the retrieved URLs, the authors manually filter out irrelevant videos and retain only those recorded by fixed surveillance cameras in public spaces. The selection is deliberately broad in terms of violent phenomena: any subjectively identified violent activity can be included, encompassing physical fights, assaults, robberies, explosions, shootings, and scenes with visible blood or other aggressive behaviors. Each retained raw video is then temporally trimmed into a short clip of fixed duration (5 seconds at 30 fps), yielding temporally "trimmed" snippets centered around the most informative portion of the event. In its final form, RWF-2000 comprises 2,000 clips and approximately 300,000 frames. The dataset is strictly binary: each clip is annotated at video level as either *Violent* or *Non-Violent*, with an exactly balanced class distribution (1,000 clips per class). No spatial annotations (e.g., bounding boxes, masks) nor frame-level temporal boundaries for individual actions are provided. This design explicitly targets video-level violence recognition, where the model must infer the presence or absence of violent behavior from the global spatio-temporal pattern observed in the clip.

The visual characteristics of the dataset are intentionally heterogeneous. RWF-2000 aggregates material from around 1,000 distinct source videos, covering a wide variety of camera positions, zoom levels, and scene layouts.The strong heterogeneity in acquisition settings and recording hardware is reflected in the resolution distribution reported in Figure 3.2. Resolutions are not standardized: the distribution of video sizes exhibits several clusters around common CCTV formats (approximately 240p, 320p, 720p, and 1080p), reflecting the diversity of real-world recording hardware. Many clips are affected by typical surveillance artefacts such as low illumination, motion blur, compression noise, partial views where only parts of the body are visible, small targets at long distance, crowded scenes, and transient or short-lived actions. From a modeling perspective, these factors make RWF-2000

Table 3.1: *Summary of the RWF-2000 dataset.*

Property	**Value**
Total clips	2,000 video clips
Classes	Violent / Non-violent (Fight / Non-Fight)
Clips per class	1,000 violent, 1,000 non-violent
Clip duration	5 s (fixed-length)
Frame rate	30 fps
Total frames	$\approx 3.0 \times 10^5$ frames
Video type	Trimmed CCTV surveillance clips from real-world scenes
Source videos	$\sim$ 1,000 unique raw surveillance videos (YouTube)
Acquisition context	Fixed surveillance cameras in public or semi-public environments
Label granularity	Video-level binary labels (Violent vs Non-violent)
Train–test protocol	Predefined 80% / 20% split, avoiding source-video leakage

noticeably more challenging than earlier, more controlled datasets.

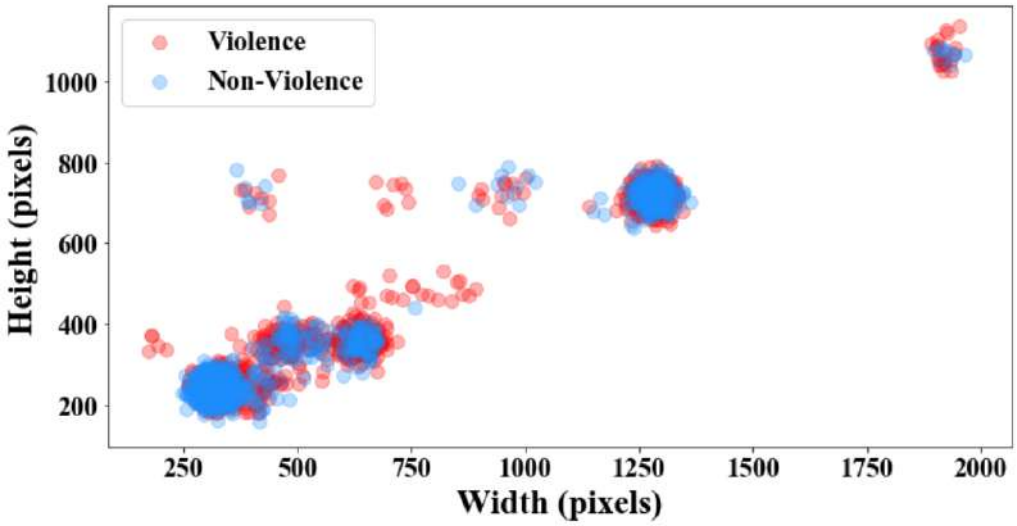


Figure 3.2: *[13] Resolution Distribution of the RWF-2000 Database.*

To this end, the authors maintain a mapping between clip identifiers and their parent video, and additionally compute color-histogram similarities between clips to identify near-duplicates. The most similar subset of clips is manually inspected to guarantee that highly redundant samples are not split across partitions. The

resulting benchmark therefore provides a realistic yet well-controlled setting for evaluating supervised violence detection methods.

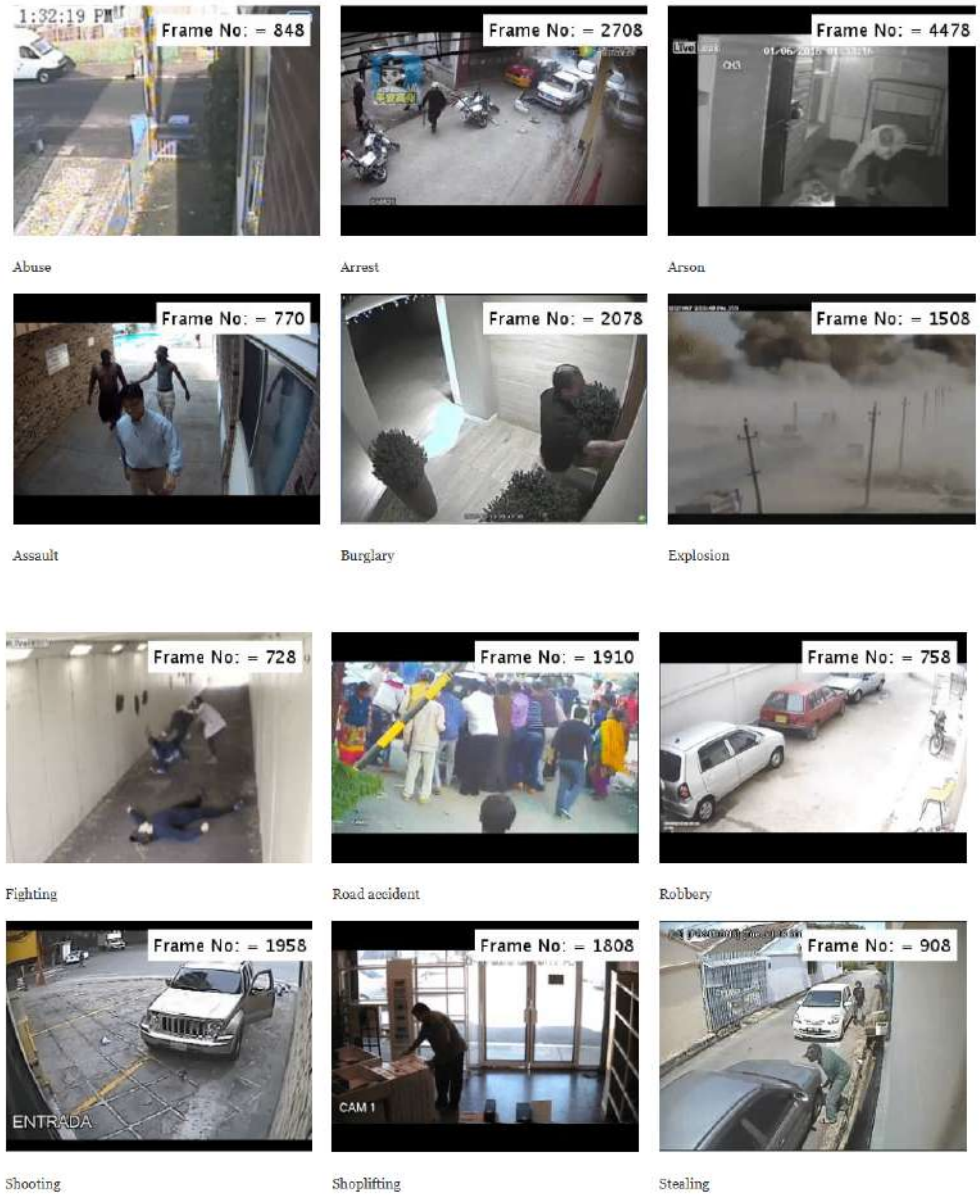
In their reference work, the authors also report baseline results on RWF-2000 using several deep models, including C3D, I3D, ConvLSTM, and their proposed Flow Gated Network. The latter combines RGB appearance and optical-flow motion streams within a 3D convolutional architecture and achieves an accuracy above 87% on the test set of RWF-2000. These baselines highlight both the difficulty of the dataset and its suitability for benchmarking modern architectures for video-based violence detection.

3.1.2 UCF-Crime

The UCF-Crime dataset [65] was introduced by Sultani *et al.* as part of the "Real-World Anomaly Detection in Surveillance Videos" benchmark, with the explicit goal of moving video anomaly detection away from small, staged datasets towards large-scale, realistic CCTV footage. UCF-Crime aggregates long, untrimmed videos captured by operational surveillance cameras in diverse environments, including streets, parking lots, shops, and other public or semi-public spaces. Figure 3.3 shows representative scenes from the UCF-Crime dataset, illustrating the diversity of environments and anomaly types captured by real-world surveillance cameras. The anomalies of interest span a broad spectrum of safety-critical events, such as interpersonal violence and property-related crimes, making the dataset particularly relevant to security and public-transportation monitoring scenarios.

Ten trained annotators search platforms such as YouTube and LiveLeak using anomaly-related keywords (e.g., "car crash", "road accident", "robbery", "street fight"), with systematic variations of the queries across multiple languages to increase coverage. The collected material is then manually filtered according to strict criteria: only footage recorded by fixed CCTV cameras is retained, while news footage, compilations, heavily edited videos, hand-held recordings, pranks, and clips where the anomalous event is visually unclear are discarded. This curation process yields 950 unedited surveillance videos containing clear anomalies and 950 normal videos without anomalous events, for a total of 1,900 real-world surveillance sequences.

UCF-Crime is designed as a large-scale, untrimmed dataset. The 1,900 videos jointly span approximately 128 hours of footage, with highly variable durations ranging from tens of seconds to several minutes [65]. Videos are captured at heterogeneous frame rates and resolutions typical of deployed CCTV systems; in the reference implementation, all frames are resized to 240×320 pixels and temporally resampled to 30 fps for feature extraction . As highlighted by the duration distribution in Figure 3.2, anomalous events typically occupy only a small fraction of each untrimmed video. The anomalies are organized into 13 semantically distinct categories: *Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion,*



Fighting, Robbery, Shooting, Stealing, Shoplifting, and *Vandalism*. These categories cover both person-to-person violence (e.g., *Assault*, *Fighting*, *Abuse*) and other high-impact safety threats such as road accidents, fires, and property crimes.

From an annotation standpoint, the dataset supports both weakly supervised anomaly detection and more fine-grained temporal evaluation. Each video is annotated at video level as either *anomalous* (containing at least one instance of any of the 13 anomaly classes) or *normal* (no anomaly present). For testing and quantitative evaluation, anomalous videos are additionally annotated with the temporal extent of the anomalous event: multiple annotators independently mark the start

Figure 3.3: *[13] Gallery of UCF-Crime dataset.*

Table 3.2: *Global characteristics of the UCF-Crime dataset.*

Property	**Value**
Total videos	1,900 untrimmed surveillance videos
Total duration	≈128–129 hours of footage
Video type	Long, untrimmed real-world CCTV recordings
Environments	Indoor and outdoor public / semi-public spaces
Frame rate (preprocessed)	320-60fps
Resolution (preprocessed)	240×320 pixels
Anomaly categories	13 realistic anomaly types
List of anomaly categories	Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, Vandalism, Road Accident
Normal videos	Videos without any annotated anomaly
Label granularity (training)	Video-level labels: Normal vs Anomalous
Label granularity (testing)	Frame-level temporal annotations for anomalous intervals
Typical tasks	Weakly supervised anomaly detection; anomaly localization; multi-class anomaly recognition

and end frames of the anomaly, and their annotations are averaged to obtain a consolidated ground truth. This provides frame-level labels for the anomalous intervals while preserving the original, untrimmed structure of the videos. The final benchmark split consists of 1,610 videos for training (800 normal and 810 anomalous) and 290 videos for testing (150 normal and 140 anomalous), with all 13 anomaly classes represented in both partitions. Training is typically framed as weakly supervised binary anomaly detection (normal vs. anomalous video), while testing evaluates frame-level anomaly localization via the anomaly scores produced along the temporal axis of each video. However, when reusing UCF-Crime in the context of this thesis, its richness must be carefully tailored to the specific goal

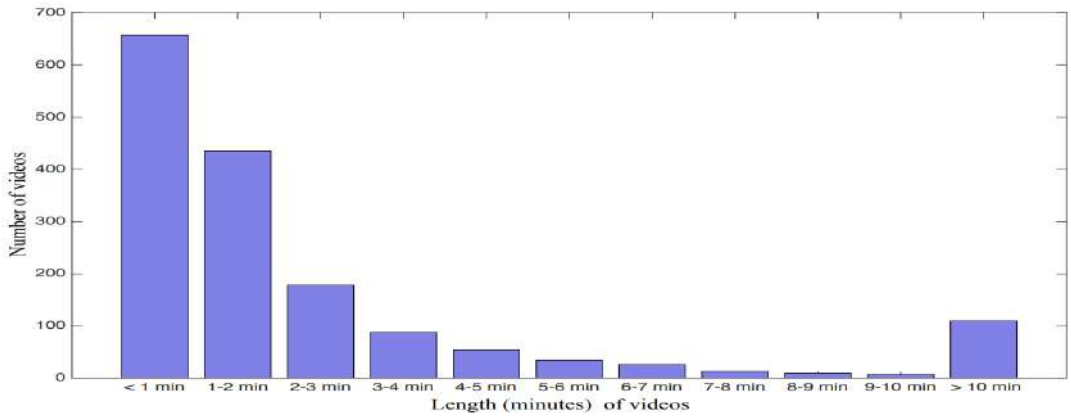


Figure 3.4: *[65] Distribution of videos according to length (minutes).*

of detecting aggressive interactions inside buses. First, the anomaly taxonomy is broader than interpersonal aggression: several classes, such as *Shooting*, *Vandalism*, *Arson*, *Explosion*, and *Road Accident*, seldom depict close-range physical fights in crowded interiors comparable to a bus cabin, and are therefore of limited use for training a violence detector targeted at public-transport vehicles. For this reason, a dedicated analysis was conducted to identify which anomaly classes and which video segments are actually informative for the recognition of assaults and fights, and to discard those that mainly involve property damage, outdoor incidents, or events driven by external hazards rather than human-to-human aggression.

Second, as highlighted by the duration distribution in Fig.3.4, UCF-Crime videos exhibit extremely heterogeneous lengths: while some clips last only a few tens of seconds, many sequences span several minutes. In most anomalous videos, the event corresponding to the annotated class occupies only a very small fraction of the temporal extent of the recording (often a few seconds), whereas the rest of the video contains normal activity or static scenes. This strong temporal sparsity is desirable for anomaly-detection research, but it is less aligned with the short, fixed-length clips (a few seconds) that are typical of real-time violence detection on-board buses. In this work, UCF-Crime is therefore not used as-is: instead, we exploit it as a source of realistic surveillance footage and manually or algorithmically focus on those temporal portions and anomaly types that best match the target scenario of aggressive behavior in a constrained, bus-like environment.

3.1.3 SmartCity CCTV Violence Detection Dataset

The Smart-City CCTV Violence Detection (SCVD) dataset [1] was introduced to explicitly address the problem of detecting both non-weaponized and weaponized violence in realistic smart-city surveillance scenarios. In contrast to earlier benchmarks that either focus only on generic fights or only on weapon appearance, SCVD

is constructed entirely from CCTV-like footage and is designed to model the presence or absence of weapons at the scene level. The overarching goal is to support deep models that can not only recognise violent behaviour, but also estimate the potential severity of an event by distinguishing between non-weaponized and weaponized aggression in urban environments.Representative examples from the Smart-City CCTV Violence Detection dataset are shown in Figure 3.5. A concise overview of the dataset and its per-class composition is summarised in Table 3.3.

From a data collection perspective, SCVD is built by crawling publicly available videos from YouTube using violence- and weapon-related keywords.

Figure 3.5: *gallery of SCVD dataset*

To reduce geographical and contextual bias, the search queries are augmented with different regional prompts so that the resulting corpus spans multiple countries and urban settings. Only footage that matches typical CCTV distributions is retained: static camera viewpoints, wide fields of view, and fixed installations in indoor or outdoor public and semi-public spaces. The version used in this thesis is the Kaggle release of SCVD [1], which contains 399 short clips, all preprocessed to a spatial resolution of 1280×720 pixels (720p) at 30 fps. Each clip is a trimmed segment, typically between 5 and 10 seconds in duration, extracted from a longer surveillance recording so as to concentrate on the most informative portion of the scene.

SCVD adopts a three-class, video-level annotation scheme with mutually exclusive labels: *Normal* (N), *Violence* (V), and *Weaponized-Violence* (WV). Normal clips depict everyday activities or benign human interactions without observable aggression. The Violence class contains non-weaponized physical altercations (e.g., pushing, punching, kicking, group fights) where the threat to safety is primarily due to direct human-to-human contact. Weaponized-Violence clips represent the most critical scenarios in the dataset: they comprise violent events in which one

or more objects are used as weapons, including both conventional weapons (e.g., guns, knives) and improvised or less conventional objects capable of causing harm. Rather than assigning fine-grained labels to individual weapon types, the dataset operates at the scene level by grouping all such situations under the WV class, encouraging deep models to learn generic visual cues associated with the presence of weapons.

SCVD Dataset Property	**Value**
Global video properties	
Total number of clips	399 CCTV video clips
Frame size	1280×720 pixels (720p)
Frame rate	30 fps (fixed for all clips)
Typical clip duration	Trimmed segments of approximately 5–10 s
Video format	RGB `.avi` surveillance videos
Camera type	Fixed-position CCTV surveillance cameras
Acquisition context	Indoor and outdoor smart-city public / semi-public areas
Label granularity	Video-level labels; no frame- or box-level annotations
Per-class composition	
Normal (N)	200 clips (50.1% of the dataset)
Violence, non-weaponized (V)	99 clips (24.8% of the dataset)
Weaponized-Violence (WV)	100 clips (25.1% of the dataset)
Total	399 clips (100%)

Table 3.3: *Technical characteristics and composition of the Smart-City CCTV Violence Detection (SCVD) dataset, using the Kaggle release based on the original SCVD proposal . All clips are real CCTV recordings from smart-city scenarios, stored as RGB `.avi` videos at 720p (*1280×720*) and 30 fps, with trimmed durations of about 5–10 s.*

In the Kaggle release [1], the 399 clips are distributed as follows: 200 Normal, 99 Violence, and 100 Weaponized-Violence samples (see Table 3.3). This yields a reasonably balanced dataset from the standpoint of binary violence detection (Normal vs. V+WV) while also providing a nearly balanced split between non-weaponized (V) and weaponized (WV) aggression. All clips share the same spatial and temporal preprocessing (720p at 30 fps) and are annotated only at video level; no bounding boxes or frame-level temporal boundaries are provided. This design is consistent with the original methodological proposal, which converts short video

segments into *salient super images* for efficient DCNN-based classification rather than relying on heavy 3D architectures.

In the context of this thesis, SCVD complements the other datasets by providing CCTV-based scenes in which both unarmed and armed aggression occur under surveillance-like viewing conditions. Its characteristics—real CCTV viewpoints, short trimmed clips, and an explicit separation between Normal, Violence, and Weaponized-Violence—make it particularly suitable for stress-testing models on high-risk situations and for analysing how the presence of weapons alters the visual and motion patterns associated with violent behaviour. In our experiments, we rely on the video-level labels and use SCVD primarily as an additional source of realistic, weapon-aware surveillance footage for training and cross-dataset evaluation of violence detection models.

3.1.4 Bus Violence Dataset

The Bus Violence dataset [7] was specifically introduced to fill the lack of public benchmarks for video violence detection in public transport environments. While most existing datasets focus on general urban surveillance scenarios with fixed outdoor cameras, Bus Violence is entirely recorded inside a moving city bus, where actors simulate both violent and non-violent behaviours under realistic operating conditions. The benchmark is designed to stress deep learning models with challenges that are typical of public transport, such as dynamic backgrounds due to vehicle motion, strong illumination changes, and frequent occlusions in a confined interior space.

From a data collection perspective, the dataset was acquired during a three-hour recording session in daytime conditions, while the bus continuously travelled and stopped within closed zones. During this session, approximately one hour was dedicated to simulated violent actions and two hours to non-violent situations. Ten actors participated in the recordings, repeatedly boarding and alighting the vehicle and changing their clothes over time to increase the diversity of appearances and interaction patterns.Figure 3.6 presents representative clips from the Bus Violence dataset, recorded inside a moving city bus under realistic operating conditions. The violent scenarios include fights between passengers, kicking or tearing pieces of onboard equipment, and robbery-like actions such as pulling or stealing an object from another person. Non-violent sequences depict passengers sitting, standing, walking along the aisle, or entering and exiting the bus in a normal fashion. Thanks to the controlled but realistic setup, the raw footage covers a range of illumination conditions (e.g., driving in direct sunlight, parking in heavily shaded areas) and background changes associated with vehicle motion.

The recording system was manually installed inside the bus and consisted of three interior surveillance cameras capturing the scene at 25 fps in H.264-encoded `.mp4` format. Two cameras were mounted in the front and rear corners of the cabin,

providing oblique views of the passenger compartment, while a fisheye camera was placed near the centre of the ceiling to obtain a wide-angle top-down view. In the public release, the video clips are provided at resolutions of 960×540 pixels for the two corner cameras and 1280×960 pixels for the fisheye camera, yielding three distinct viewpoints and effective resolution groups in the final dataset. This multi-view configuration allows models to be evaluated on substantially different perspectives and projection geometries within the same physical environment.

Figure 3.6: *gallery of Bus Violence dataset*

After acquisition, the continuous recordings were manually segmented and curated. The raw videos were split into short clips ranging from 16 to 48 frames, corresponding to temporal windows of approximately 0.64–1.92 s at 25 fps. Each clip is intended to capture a single, well-defined situation (either violent or non-violent), avoiding segments in which different behaviours overlap. The curation pipeline followed a two-stage manual annotation protocol. In the first stage, three annotators independently labelled each candidate clip as *violent* or *non-violent*. In the second stage, two additional experts reviewed the preliminary labels, discarded ambiguous or wrongly segmented samples, and consolidated the final annotations.

No automatic labelling tools were used; the authors explicitly opted for fully manual supervision to increase reliability and avoid error propagation.

Since the initial segmentation produced more non-violent than violent clips, the authors applied random undersampling to the non-violence class to obtain a perfectly balanced dataset. The final curated version of Bus Violence consists of 1,400 short video clips, evenly split between the two classes: 700 *Violence* and 700 *Non-violence* samples. For each class, the clips are approximately balanced across the three camera viewpoints: for the three views, the authors report 212, 222, and 266 clips for the Violence class, and 240, 210, and 250 clips for the Non-violence class, respectively. All clips are distributed in two separate folders (`Violence` and `NonViolence`), each containing 700 H.264 `.mp4` files, and official train/test split files are provided with the public release. A summary of the technical characteristics and class composition of the dataset is reported in Table 3.4.

Property	**Value**
Global video properties	
Total clips	1,400 short clips recorded inside a moving bus
Classes	2 (Violence, Non-violence)
Label type	Video-level binary labels
Acquisition platform	Standard city bus during real motion
Cameras	3 (two corner views, one central fisheye)
Frame rate	25 fps
Clip length	16–48 frames (0.64–1.92 s)
Resolutions	960×540 (corner), 1280×960 (fisheye)
Video format	RGB .mp4 (H.264)
Recording duration	≈ 3 h (1 h violent, 2 h non-violent)
Illumination/background	Dynamic lighting + motion-induced background changes
Actors	10 participants, varying clothes
Annotation protocol	3 annotators + 2 expert reviewers
Per-class composition	
Violence class	700 clips (50%)
Non-violence class	700 clips (50%)
Per-view clips (Violence)	212, 222, 266 clips
Per-view clips (Non-violence)	240, 210, 250 clips
Folder structure	Two folders: `Violence`, `NonViolence`
Official splits	Train/test lists provided
Intended use	Benchmark for cross-dataset generalization

Table 3.4: *Technical characteristics and composition of the Bus Violence dataset [7].*

These datasets differ significantly in terms of resolution, frame rate, clip length, and acquisition context. For instance RWF-2000 and SCVD consist of short,

fixed-length clips, while UCF-Crime features longer and more variable-duration sequences. Frame rates vary from 20 to 60 fps, and resolutions range from 352×288 px to 1280×960 px.

3.1.5 Laboratory Dataset

The proprietary laboratory dataset consists of 2,000 clips of 3 s each (90 frames at 30 fps) recorded in the full-scale bus simulator described in the previous section, labelled into two classes: *Fight* and *No Fight*. The simulator reproduces the interior and exterior dimensions of a 12 m city bus, including driver's cabin, internal height, door configuration, and seating layout, based on on-site measurements performed on several vehicles from the fleet of a local public transport operator. Camera positions in the simulator were chosen to match the on-board architecture later deployed on real buses during the field trials, so that the laboratory dataset and the field experiments share the same sensing geometry. Violent behaviours (such as punches, slaps, pushes, grabbing, pulling, and kicks) were staged by adult volunteers, taking inspiration from typical assaults on public transport as documented in publicly available videos. After the recording sessions, the resulting clips were jointly analysed with collaborators from the local public transport operator, who confirmed that the simulated scenes are realistic and representative of violent events that should be reported according to their operational practice and local regulations. A multi-view collage illustrating the six camera perspectives of the laboratory dataset is shown in Figure 3.7. No real passengers or in-service vehicles were involved, and all recordings were carried out in compliance with institutional guidelines and applicable data protection rules, with informed participation of the volunteers. Six IP cameras are installed inside the simulator, organised in three logical units: a front unit covering the area around the driver and the front door, a central unit monitoring the middle section of the cabin, and a rear unit focusing on the back door and rear seating area. Each unit comprises two ceiling-mounted cameras with wide-angle optics (approximately 160° horizontal field of view) and partially overlapping fields of view. All cameras operate at a resolution of 1080 × 720 px and 30 fps. This multi-view configuration ensures that passengers in the aisle or near the doors are typically observed from at least two viewpoints, which reduces the impact of occlusions due to body pose, crowding, or relative positioning. A collage of representative frames from the six views is shown in Figure 2.4, illustrating the coverage of the front, central, and rear regions of the bus interior.

To obtain a dataset that is both class-balanced and representative of typical operating conditions, the recording campaign was structured according to a simple factorial design combining behavioural class, crowding, and illumination. For each of the six camera views, scenes were recorded with at least four people visible in the field of view (high occupancy) and with fewer than four people (low occupancy).

Figure 3.7: *Laboratory Dataset gallery.*

For both occupancy levels, sequences were acquired during simulated daytime conditions, with natural or diffuse daylight dominating the scene, and during simulated nighttime conditions, with the interior lighting of the bus providing the main illumination. In each of the four resulting scenarios (low day, low night, high day, high night), both Fight and No Fight behaviours were enacted and recorded. Figure 3.8 shows examples of low- and high-occupancy scenes for both classes.

The continuous recordings from all cameras were segmented into fixed-length clips of 3 s. Each candidate clip was manually labelled as *Fight* or *No Fight* based on the presence or absence of clearly observable physical aggression. Clips containing transitions between behaviours, ambiguous interactions, or severe visibility problems were discarded. The final corpus is strictly balanced at class level, with 1,000 Fight and 1,000 No Fight clips, and approximately balanced across views, crowding levels, and illumination conditions. At scenario level, we enforced an equal distribution of Fight and No Fight samples: for each combination of occupancy and illumination (low day, low night, high day, high night), 250 clips were retained, of which 125 are Fight and 125 are No Fight. Within each scenario, clips are approximately evenly distributed across the six viewpoints (about 20–22 clips per view and class), so that no single camera or operating condition dominates the dataset. The resulting stratification is summarised in Table 3.6.

To ensure compatibility across sources and facilitate transfer learning from models pre-trained on Kinetics-400, a normalization pipeline was applied to all video

Table 3.6: *Stratification of the proprietary laboratory dataset by behavioural class, crowding, and illumination. Each scenario is balanced between Fight and No Fight clips; values are aggregated over the six camera views.*

Scenario	Fight	No Fight	Total
Low occupancy, day	125	125	250
Low occupancy, night	125	125	250
High occupancy, day	125	125	250
High occupancy, night	125	125	250
Total	1,000	1,000	2,000

Figure 3.8: *Representative examples of low-occupancy scenes (top row, fewer than four individuals) and high-occupancy scenes (bottom row, at least four individuals) for both* Fight *and* No Fight *classes in the laboratory dataset. All scenes were recorded in a controlled experimental environment using human-sized mannequins to systematically reproduce varying passenger density conditions.*

samples. Given that violent actions typically unfold over a few seconds, temporal normalization was used to standardize video duration to maximum 3 seconds. Additionally, frame resizing and pixel normalization ensured alignment with the original training conditions of the pre-trained model. The preprocessing pipeline included the following steps:

- Format conversion: All videos were converted to .mp4 format for standardization;
- Temporal normalization: Each video was trimmed to have a duration between 1 and 3 seconds, i.e., between 30 and 90 frames at 30 fps;
- Frame resizing: Frames were resized to 256×256 pixels to match the input size expected by models pre-trained on Kinetics-400;

- Pixel normalization: Frame pixel values were normalized using a mean of [0.45, 0.45, 0.45] and a standard deviation of [0.225, 0.225, 0.225], consistent with Kinetics-400 pre-training settings;
- Uniform temporal sampling: A uniform temporal subsampling was applied to extract exactly 30 frames from each video, ensuring consistency in the temporal resolution across the dataset.
- Frame rate standardization: All clips were resampled at 30 fps, i.e., 30 frames per second of video were retained, normalizing temporal resolution across datasets with different native frame rates.

Table 3.7 summarizes the number of clips obtained from each dataset after normalization. The final dataset is composed of short clips, each manually labeled as either Fight or No Fight. Note that during the trimming or resampling of longer violence-related videos, we also extracted segments that did not directly contain violent actions. For example, consider the Abuse class from the UCF-Crime dataset, which includes 50 videos ranging from 10 to 120 seconds in duration, with frame rates between 20 and 60 fps. After temporal normalization, these were split into 851 shorter clips (1 to 3 seconds each). However, only 189 of them were labeled as Fight, while the remaining 662 as No Fight, as they did not show clear violent behavior. The labeling of clips during post-processing was performed manually, resulting in a total of 26980 clips labeled as No Fight and 5042 as Fight.

Data augmentation techniques were additionally employed to expand the dataset and enhance model generalization [48]. The following transformations were applied:

- Brightness adjustment: $\pm 20\%$ random variation to simulate lighting changes;
- Horizontal flipping: to provide spatial variation and avoid overfitting to specific orientations of motion;
- Light jittering: $\pm 10\%$ random adjustments in hue, saturation, and contrast to mimic changes in color and lighting conditions;
- Gaussian noise: addition of zero mean gaussian noise owith standard deviation of 0.01 to simulate sensor noise and compression artifacts.

These augmentations were systematically applied to all videos labeled as Fight, to enrich the representation of this underrepresented class. To avoid introducing excessive bias, the same transformations were selectively applied to 20% of the NoFight videos, randomly sampled at a ratio of 1 out of every 5, thus maintaining a balanced approach to augmentation across the dataset

After data augmentation, the Fight class increased from 5,042 to 6,042 clips by adding one augmented view to a subset of samples. For the No Fight class, augmentations were applied selectively (target ratio $\approx 20\%$) and, after quality filtering

Dataset	Classes	Clip Dur. (s)	#Clips	Normalized Clips (1-3s)	Used No Fight	Used Fight
Bus Violence (30fps)	Violence	1	700	700	180	417
	NoViolence	1	700	700	700	0
UCF-Crime (20-60fps)	Abuse	10–120	50	851	662	189
	Arrest	10–120	50	937	760	177
	Assault	10–120	50	687	358	329
	Fighting	10–120	50	1569	942	627
	Normal	10–3600	200	19372	19372	0
RWF-2000 (30fps)	Fight	5	1000	1973	0	1973
	NoFight	5	1000	1992	1992	0
SCVD (20-30fps)	Normal	3–10	246	1014	1014	0
	Violence	3–10	111	322	0	322
	Weaponized	3–10	124	8	0	8
Proprietary (30fps)	Fight	3	1000	1000	0	1000
	NoFight	3	1000	1000	1000	0
				Total	26980	5042

Table 3.7: *Summary of the video clips obtained from each source after temporal normalization to short segments (1–3 s). For each original dataset and class, the table reports the native frame rate, the original number of videos, the number of normalized clips, and the subset finally used as No Fight or Fight in the unified dataset..*

Characteristic	Description	
Video Format	.mp4 standardized	
Clip Duration	3 seconds per clip (90 frames at 30 fps)	
Frame Resolution	256 × 256 pixels (resized)	
Input Pixel Normalization	Mean 0.45 and Std 0.225	
#Clips	Fight	No Fight
Training	4230	4230
Validation	1510	1510
Test	302	302

Table 3.8: *Final dataset characteristics.*

and deduplication, the class totaled 30,720 clips. To address the severe class imbalance, the No Fight clips were downsampled to match the 6,042 Fight clips, yielding a balanced dataset of 12,084 clips. The dataset was split into training, validation, and test sets using a 70-25-5 ratio. Table 2 summarizes the characteristics of the final dataset.

Chapter 4

Model Training and Inference

4.1 Deep Learning Models

Violence detection in on-board CCTV streams is formulated as a clip-level action recognition problem, in which a deep model maps a short spatio–temporal video segment to a discrete semantic label. As introduced in Section 2, each video clip is obtained through temporal sampling of decoded frames (cf. (2.2)) and is represented as a tensor

$$\mathbf{X} \in \mathbb{R}^{T \times H' \times W' \times C}, \tag{4.1}$$

where T denotes the number of frames per clip, $H' \times W'$ the network input resolution, and $C = 3$ the number of RGB channels.

Given an input clip $\mathbf{X}$, the deep video classifier is implemented as a composition of a spatio–temporal backbone and a task-specific prediction head, as formalized in (2.3) and (2.4). Specifically, the backbone $\phi_\theta(\cdot)$ encodes the clip into a latent representation $z \in \mathbb{R}^d$, which is subsequently mapped to class logits

$$\mathbf{z} = f_\theta(\mathbf{X}) \in \mathbb{R}^K, \tag{4.2}$$

where K is the number of target classes and $f_\theta(\cdot)$ denotes the overall network composed of feature extraction and classification stages. In the multi-class setting, the predicted label corresponds to $\arg\max_k \mathbf{z}_k$, while in the binary violence detection task considered here, the logits are reduced to a scalar confidence score via a sigmoid activation, as described in (2.4).

The selection of backbone architectures is driven by two key requirements: (i) the ability to capture fast, localized motion patterns characteristic of aggressive and violent interactions, and (ii) compliance with the computational, power, and latency constraints imposed by embedded, on-board inference (Section 3).

To span a representative range of accuracy–efficiency trade-offs, three families of video models were adopted in this work:

- **SlowFast R50**, a dual-pathway architecture designed to jointly model slow semantic context and fast motion dynamics, well suited to short, high-intensity actions;

- **X3D-L**, an efficiency-oriented family that progressively expands network capacity along multiple axes while maintaining favorable accuracy-per-FLOP characteristics;

- **R(2+1)D**, a factorized 3D convolutional baseline that decomposes spatio–temporal filtering into separate spatial and temporal operations, providing a strong and well-understood reference model.

All selected architectures are widely validated in the action recognition literature and integrate naturally within the spatio–temporal processing and fusion framework described in Sections 2 and 3. Their complementary design philosophies allow systematic evaluation of performance trade-offs under realistic on-board deployment constraints.

4.2 SlowFast R50

SlowFast is a two-pathway architecture designed to decouple the learning of spatial semantics from the modeling of high-frequency motion [24]. The Slow pathway processes temporally sparse frames and focuses on appearance-driven cues (e.g., body pose, scene context), while the Fast pathway operates at a higher frame rate to preserve rapid dynamics (e.g., punches, pushes, sudden body displacements). Let τ denote the temporal stride of the Slow pathway sampling and $\alpha > 1$ the frame-rate ratio between Fast and Slow. The two inputs can be expressed as:

$$\mathbf{X}_{\text{slow}} = \mathbf{X}\Big[\, t = 1, 1+\tau, 1+2\tau, \dots \Big], \qquad \mathbf{X}_{\text{fast}} = \mathbf{X}\Big[\, t = 1, 1+\tfrac{\tau}{\alpha}, 1+2\tfrac{\tau}{\alpha}, \dots \Big]. \quad (4.3)$$

To limit overhead, the Fast branch is intentionally lightweight through a reduced channel capacity controlled by a factor $\beta \in (0,1)$, so that the Fast pathway uses approximately β times the channels of the Slow pathway [24].

Architecturally, both pathways follow a residual design derived from ResNet backbones [32], and the information exchange is implemented via lateral connections that inject motion-sensitive features from the Fast stream into the Slow stream at multiple stages [24]. In the context of bus violence detection, this inductive bias is particularly relevant: aggressive actions are often characterized by short bursts of motion that can be attenuated by sparse temporal sampling, hence motivating the explicit high-rate ... Fast pathway. The SlowFast dual-pathway architecture adopted in this work is illustrated in Figure 4.1. .

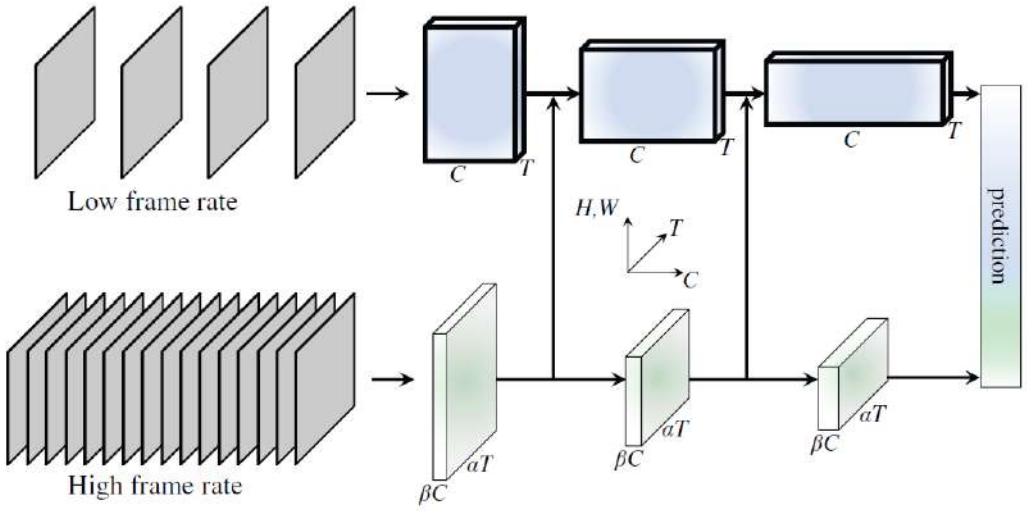


Figure 4.1: *A SlowFast network has a low frame rate, low temporal resolution Slow pathway and a high frame rate, higher temporal resolution Fast pathway. The Fast pathway is lightweight by using a fraction (e.g., 1/8) of channels. Lateral connections fuse them*

4.2.1 Architectural rationale and design choices.

SlowFast models address action recognition as a *multi-temporal* perception problem, explicitly separating the representation of (i) *semantic content* (objects, actors, scene context) from (ii) *fast motion cues* (short-term dynamics and micro-motions). The architecture comprises two pathways operating at different temporal resolutions: a *Slow* pathway sampling sparsely in time (low frame-rate, higher channel capacity) and a *Fast* pathway sampling densely (high frame-rate, lower channel capacity) [24]. In a canonical configuration, the speed ratio is set to $\alpha = 8$ and the channel ratio to $\beta = \frac{1}{8}$, meaning that the Fast stream processes α times more frames while using β times the channels of the Slow stream [24]. This design yields a favorable accuracy/complexity trade-off: the Fast pathway is intentionally *thin*, hence its additional cost is contained while still capturing high-frequency motion patterns that are critical for distinguishing actions with similar appearance but different dynamics.

4.2.2 Fusion mechanism and temporal alignment.

A key component is the *lateral fusion* that injects motion-rich Fast features into the semantically stronger Slow stream. Because the two pathways have mismatched temporal dimensions, SlowFast introduces dedicated lateral connections that perform temporal alignment before fusion. In particular, one option is *time-strided convolution*: a $5 \times 1 \times 1$ 3D convolution with stride α (temporal) is applied to Fast features to match the Slow temporal grid; the resulting tensor can be fused

by summation or concatenation [24]. This mechanism is especially relevant for crowded in-vehicle scenes, where subtle limb accelerations, pushing gestures, or rapid head/torso movements may be decisive cues for early aggression recognition.

4.2.3 Backbone and variants (R50 and beyond).

The "R50" instantiation adopts a ResNet-50 backbone with residual blocks that stabilize optimization and enable depth scaling [32]. SlowFast itself forms a *family* of models parameterized by clip length and sampling stride (often denoted by $T \times \tau$) and by backbone depth (e.g., R50 vs. R101), with optional context modules such as *Non-Local* blocks to capture long-range dependencies across space-time [24, 79]. These variants are widely used in standard action classification benchmarks and extend naturally to spatio-temporal action detection, where localized actor-centric reasoning is required (e.g., AVA-style setups) [24, 29]. In practical deployments, this modularity enables selecting the best operating point under fixed compute/latency budgets while preserving the two-rate inductive bias that remembering "what" and "how fast" are often complementary for action understanding.

Aggression episodes in buses typically involve short-duration, high-acceleration interactions (e.g., sudden grabs, punches, pushes) embedded in a visually cluttered environment with occlusions and varying viewpoints. SlowFast is well aligned with this regime because (i) the Fast pathway preserves dense temporal sampling for motion saliency, while (ii) the Slow pathway retains robust semantic context useful to disambiguate visually similar motions (e.g., gesturing vs. striking) [24]. Moreover, since the Fast stream is thin by design, SlowFast offers a principled way to improve motion sensitivity without a proportional increase in parameters, which is desirable for real-time embedded inference.

4.3 X3D-L

X3D is a family of efficient video recognition networks obtained through a progressive, axis-wise expansion procedure [23]. Starting from a compact seed architecture, X3D expands capacity along multiple axes—including temporal duration, temporal sampling, spatial resolution, network width, and depth—to achieve an improved accuracy-to-compute trade-off. The *X3D-L* variant represents a higher-capacity point on this Pareto frontier, providing strong recognition accuracy while remaining substantially more efficient than many earlier 3D CNN designs at comparable performance [23].

From a systems perspective, X3D is well suited to embedded deployment because its design explicitly targets efficiency while preserving the spatiotemporal resolution needed to discriminate subtle action cues. In this thesis, X3D-L is adopted as an alternative to SlowFast to evaluate whether a single-stream, efficiency-optimized

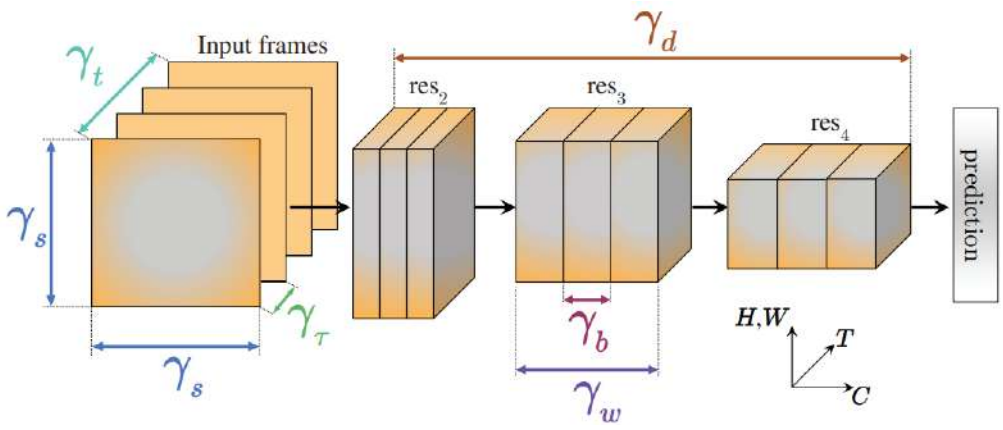


Figure 4.2: *X3D networks progressively expand a 2D network across the following axes: temporal duration γ_t, frame rate γ_τ, spatial resolution γ_s, width γ_w, bottleneck width γ_b, and depth γ_d.*

backbone can match or exceed the accuracy of dual-pathway designs under the same clip normalization and training protocol. Figure 4.2 illustrates the progressive axis-wise expansion strategy underlying the X3D family of architectures.

4.3.1 From mobile image models to efficient video networks.

X3D is conceived as a systematic approach to derive *efficient* spatiotemporal networks by progressively expanding a tiny 2D image architecture along multiple axes (temporal duration, frame rate, spatial resolution, width, bottleneck width, and depth) [23]. The starting point is inspired by "mobile-regime" architectures employing *channel-wise (depthwise) separable convolutions* as a core building block, which is known to provide strong accuracy under tight compute constraints [23, 34]. Instead of inflating a heavy 2D backbone into 3D, X3D explicitly explores which expansion axis provides the best accuracy gain per added complexity, using a stepwise expansion strategy [23].

4.3.2 Compound scaling and the X3D family.

The outcome is a family of models (X3D-XS, X3D-S, X3D-M, X3D-L, X3D-XL) spanning different complexity regimes [23]. Importantly, the family supports *compute-aware selection*: for instance, X3D-L reaches competitive top-1 accuracy while remaining lightweight in parameters (reported in the original study at single-digit millions of parameters) and with substantially fewer operations than heavier baselines [23]. This spectrum is particularly relevant for on-board deployments where throughput, thermal envelopes, and power budgets constrain the feasible model class.

4.3.3 Architectural traits relevant to in-vehicle perception.

From a systems perspective, X3D is attractive because it preserves *high spatiotemporal resolution* while keeping the network *thin* in terms of width, a property explicitly highlighted as a surprising but beneficial finding in the original work [23]. In bus interiors, where the scene contains multiple interacting passengers and frequent partial occlusions, maintaining adequate resolution is essential to preserve fine motion cues and small actor regions, while a thin model limits latency and resource usage. Additionally, X3D models are commonly trained and evaluated on large-scale video datasets (e.g., Kinetics) and on detection benchmarks (e.g., AVA), evidencing their versatility across recognition tasks [23, 29, 9].

Aggression recognition must balance *sensitivity* to short-term motion with *deployability*. X3D-L provides a compelling compromise: it is designed explicitly for efficiency via progressive expansion, yet remains competitive with stronger but heavier architectures [23]. This makes X3D-L an excellent candidate when the overall pipeline includes additional compute burdens (multi-camera ingestion, buffering, encryption, logging) and the inference engine must still meet strict real-time constraints.

4.4 R(2+1)D

R(2+1)D is a spatiotemporal convolutional architecture that factorizes a 3D convolution into a 2D spatial convolution followed by a 1D temporal convolution, inserting a non-linearity between the two operations [69]. Let $\mathcal{K} \in \mathbb{R}^{k_t \times k_h \times k_w}$ denote a 3D kernel. R(2+1)D replaces the direct 3D operator with the composition:

$$\mathrm{Conv}_{3D}(\mathbf{X}; \mathcal{K}) \;\approx\; \mathrm{Conv}_{1D}^{(t)}\Big(\sigma\Big(\mathrm{Conv}_{2D}^{(h,w)}(\mathbf{X}; \mathcal{K}_s)\Big); \mathcal{K}_t\Big), \tag{4.4}$$

where $\mathcal{K}_s \in \mathbb{R}^{1 \times k_h \times k_w}$ is the spatial kernel, $\mathcal{K}_t \in \mathbb{R}^{k_t \times 1 \times 1}$ is the temporal kernel, and $\sigma(\cdot)$ denotes a point-wise non-linearity (e.g., ReLU). This factorization has two practical implications: (i) it explicitly separates spatial and temporal modeling, and (ii) it increases representational capacity by adding an extra non-linearity, which can ease optimization and improve recognition accuracy compared to monolithic 3D convolutions [69].

4.4.1 Factorized spatiotemporal convolution as an inductive bias

R(2+1)D originates from an empirical study on spatiotemporal convolutions and is built upon residual learning backbones [69, 32]. The central idea is to replace a full 3D convolution with a *factorized* block: a 2D spatial convolution followed

by a 1D temporal convolution. Concretely, a 3D kernel of size $N_{i-1} \times t \times d \times d$ is decomposed into (i) M_i spatial filters of size $N_{i-1} \times 1 \times d \times d$ and (ii) N_i temporal filters of size $M_i \times t \times 1 \times 1$ [69]. The intermediate dimensionality M_i is selected to approximately match the parameter count of the original 3D layer, e.g., via a closed-form choice reported in the reference work [69]. This decomposition introduces an additional non-linearity between spatial and temporal processing stages, increasing representational capacity while preserving computational comparability.

4.4.2 Optimization benefits and practical stability

Beyond efficiency, factorization is motivated by *optimization*: the authors report that R(2+1)D yields lower training error than comparable 3D ResNets, suggesting that separating spatial and temporal modeling can make deep video models easier to train, particularly as depth increases [69]. For applied scenarios with domain shift (e.g., CCTV-like bus cameras, occlusions, viewpoint changes), such optimization stability is valuable because it reduces the sensitivity of training to hyperparameters and can improve robustness when fine-tuning from large-scale pretraining.

4.4.3 Variants and typical usage in action recognition

R(2+1)D is commonly instantiated with different depths (e.g., 18/34 layers) and evaluated in RGB-only and two-stream configurations; in the original benchmark results, R(2+1)D demonstrated strong performance across large-scale datasets and competitive transfer to smaller benchmarks [69, 9]. In embedded settings, RGB-only variants are often preferred because optical flow can be one order of magnitude more expensive to compute than frame-based inference, thereby undermining real-time constraints [69]. As a consequence, R(2+1)D provides a pragmatic baseline that balances accuracy and computational feasibility.

From a violence-detection standpoint, the factorized design is particularly well suited to model *local motion primitives* (rapid arm swings, body collisions) while retaining spatial context. The combination of residual learning, explicit temporal modeling, and favorable optimization properties makes R(2+1)D a robust candidate for training on datasets where aggressive actions are rare and visually heterogeneous, and where generalization under occlusion and limited resolution is critical [69, 32].

4.5 Training and Validation Strategy

Training violence detectors for on-board CCTV streams is intrinsically a *transfer learning* problem: the target domain (wide-angle, ceiling-mounted bus cameras;

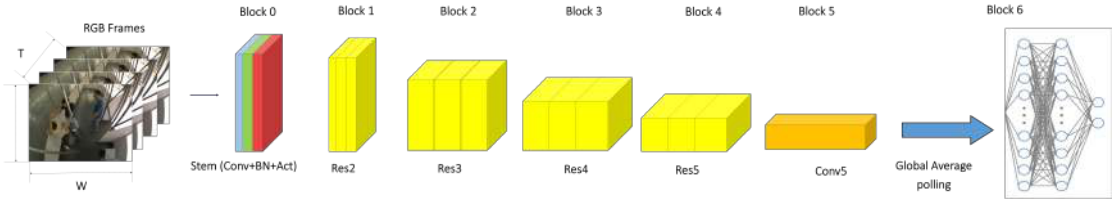


Figure 4.3: *Schematic adaptation of a pre-trained action recognition backbone to the binary violence detection task. The original multi-class classification head is replaced with a task-specific binary classifier (Fight vs. No-Fight), while the spatiotemporal backbone is fine-tuned according to the progressive unfreezing strategy described in Section 4.5.*

frequent occlusions; strong illumination changes; limited camera viewpoints) differs substantially from canonical benchmarks, while the amount of truly domain-representative labeled data is typically limited. For this reason, the adopted strategy couples (i) a principled data split protocol, (ii) fine-tuning from large-scale spatiotemporal pretraining, and (iii) a controlled optimization schedule designed to avoid catastrophic forgetting and to stabilize convergence.

A key requirement in this thesis is *comparability* across backbones. Therefore, the same clip normalization (duration and spatial resolution), the same pixel normalization parameters, and an analogous fine-tuning schedule are applied to SlowFast R50, X3D-L, and R(2+1)D. The final classification layer is replaced with a binary head (Fight vs. No-Fight), consistently with the schematic adaptation shown in Figure 4.3.

4.5.1 Train/Validation/Test Splits

The dataset preparation stage yields a collection of *trimmed* clips standardized in format and duration. In the final configuration used for comparative training, each sample corresponds to a 3-second temporal window (approximately 90 frames at 30 fps) resized to 256×256 and normalized with per-channel mean 0.45 and standard deviation 0.225 to ensure compatibility with the pretraining regime and consistent numerical conditioning across models.

After augmentation, filtering, and class balancing (Section 3.1.5), the resulting dataset is made *balanced* by construction, with an equal number of Fight and No-Fight clips. Let $\mathcal{D}$ denote the balanced dataset of N clips,

$$\mathcal{D} = \{(\mathbf{X}_i, y_i)\}_{i=1}^{N} \tag{4.5}$$

where $y_i \in \{0,1\}$ and $y_i = 1$ indicates Fight. The split is performed according to a 70–25–5 ratio:

$$\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}} \cup \mathcal{D}_{\text{test}}, \qquad |\mathcal{D}_{\text{train}}| : |\mathcal{D}_{\text{val}}| : |\mathcal{D}_{\text{test}}| = 70 : 25 : 5, \tag{4.6}$$

Setting	Train	Validation	Test
Full dataset (Public + Proprietary)	70% (stratified)	25% (stratified)	5% (stratified)
Public-only (ablation setting)	fixed cardinality	fixed cardinality	fixed cardinality

Table 4.1: *Summary of the training/evaluation split protocol adopted in this work.*

while preserving class balance within each subset via stratification. The validation set is used for model selection and for monitoring generalization during fine-tuning; the test set is used exclusively for final reporting.

In addition to the *full* dataset setting (public + proprietary), the thesis also considers a *public-only* training configuration to quantify domain shift and isolate the contribution of domain-specific data. In this second setting, a balanced subset is extracted from public sources, expanded through augmentation to a fixed-size collection, and split into train/validation/test with fixed cardinalities. This controlled protocol supports ablation studies and enables a direct comparison of generalization when transferring to proprietary bus-like scenarios (reported later in Section 5.1).

4.5.2 Optimization Settings and Hyperparameters

All backbones are initialized from weights pre-trained on Kinetics-400, a large-scale action recognition dataset that provides strong generic spatiotemporal representations for transfer. Because the original architectures are trained for multi-class recognition, the final classifier is replaced with a task-specific binary head (Fight vs. No-Fight), as exemplified in Figure 4.3. In this thesis, this step is performed consistently across SlowFast R50, X3D-L, and R(2+1)D to ensure that observed differences are attributable to the backbone design rather than to heterogeneous adaptation choices.

Let $\mathbf{z}_i = f_\theta(\mathbf{X}_i) \in \mathbb{R}^2$ be the logits for clip i and $p_\theta(y_i = k \mid \mathbf{X}_i) = \mathrm{softmax}(\mathbf{z}_i)_k$. The training objective is the standard cross-entropy over the binary label:

$$\mathcal{L}(\theta) = -\frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(\mathbf{X}_i, y_i) \in \mathcal{D}_{\text{train}}} \log p_\theta(y_i \mid \mathbf{X}_i). \tag{4.7}$$

While class balancing reduces the need for cost-sensitive losses, the safety-critical nature of the application motivates monitoring class-wise errors explicitly (Section 4.5.3).

Directly fine-tuning all backbone layers can lead to unstable optimization and overfitting, especially under domain shift and limited target data. To mitigate this, we adopt a *progressive unfreezing* schedule: an initial warm-up phase trains only the newly added classification head, keeping the backbone frozen; afterward, only the *upper* backbone blocks are unfrozen and optimized jointly with the head, while low-level blocks remain frozen to preserve general visual primitives. This yields a controlled adaptation of high-level spatiotemporal features to the bus environment.

Algorithm 1 Progressive unfreezing strategy (generic form used for all backbones).

Inputs: number of epochs E, warm-up E_w, model blocks $\{0, \dots, B\}$ where B is the head block
Freeze backbone blocks $0, \dots, B-1$; unfreeze head block B
`backbone_unfrozen` $\leftarrow$ `false`
for $e \leftarrow 1$ **to** E **do**
    if $e = E_w$ **then**
        if `backbone_unfrozen` = `false` **then**
            Unfreeze upper backbone blocks (e.g., 3,4,5)
            Rebuild optimizer with current trainable parameters
            `backbone_unfrozen` $\leftarrow$ `true`
        end if
    end if
    Train one epoch
    if overfitting detected **then**
        if `backbone_unfrozen` = `true` **then**
            Refreeze upper backbone blocks
            Rebuild optimizer with current trainable parameters
            `backbone_unfrozen` $\leftarrow$ `false`
        end if
    end if
end for

Optimization is performed with Adam, using an initial learning rate η_0 and a step-based scheduler. Denoting with s the step size and with $\gamma \in (0,1)$ the multiplicative decay factor, the learning rate at epoch e is:

$$\eta(e) = \eta_0 \, \gamma^{\left\lfloor \frac{e}{s} \right\rfloor}. \tag{4.8}$$

The same schedule is applied across models to ensure a fair comparison, while the warm-up/unfreezing transition (Algorithm 1) modulates the *set of trainable parameters* over time.

Item	Value / Policy
Clip duration / frames	3 s, ≈ 90 frames at 30 fps
Spatial resolution	256×256 (resized)
Pixel normalization	mean 0.45, std 0.225
Initialization	Kinetics-400 pretraining
Head adaptation	binary classifier (Fight vs. No-Fight)
Warm-up	head-only training for E_w epochs
Fine-tuning	unfreeze upper backbone blocks (e.g., 3–5)
Optimizer	Adam
LR schedule	Step decay (Eq. 4.8)
Model selection	validation-driven (F1/Recall monitored per class)

Table 4.2: *Core hyperparameters shared across model trainings.*

4.5.3 Evaluation Metrics

Because violence detection is safety-critical, evaluation must explicitly quantify both missed detections (false negatives) and false alarms (false positives). Let the Fight class be the positive class. From the confusion matrix we define: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Standard clip-level metrics are then:

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}}, \\ \text{F1} &= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \\ \text{Accuracy} &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}. \end{aligned} \tag{4.9}$$

In this thesis, metrics are reported (i) *per class* to highlight asymmetric failure modes (e.g., high accuracy driven by the No-Fight class), and (ii) as aggregated indicators (macro-averaged F1 when appropriate) to support backbone comparison. In operational terms, Recall on the Fight class is a primary indicator of system sensitivity (minimizing missed aggressions), while Precision controls the rate of spurious alarms that would undermine trust and usability.Representative confusion matrices for the evaluated backbones are shown in Figure 5.5.

4.6 Software Architecture

The violence-detection prototype is conceived as an *edge video analytics* system deployed entirely on-board the vehicle. In the following, we assume the hardware configuration introduced in Section 2 (Hardware Architecture) as the execution platform, and focus exclusively on the organization and interaction of the software components operating on top of it. All stages of the processing pipeline—capture, buffering, inference, and alerting—are executed locally in order to meet strict real-time constraints and to avoid continuous video backhaul, in line with current trends in edge-based intelligent surveillance systems [56, 36, 47]. On this hardware platform, the embedded GPU server interfaces directly with the in-vehicle IP cameras and with the communication and notification subsystems, enabling a closed-loop pipeline from perception to action. From a software standpoint, the overall processing pipeline can be decomposed into three main subsystems: (i) multi-stream video ingestion, (ii) clip construction and pre-processing through a circular buffering strategy, and (iii) real-time inference with decision logic and alert dispatch. This modular decomposition is summarized schematically in Fig. 4.4, while the temporal buffering and update mechanism is detailed separately in Fig. 2.1.

The proposed architecture explicitly separates per-camera clip-level inference from system-level decision making. In particular, deep neural networks operate independently on each camera stream to extract clip-level violence scores, whereas alert generation is handled by a dedicated system-level module that aggregates asynchronous predictions over time and enforces persistence and thresholding constraints before triggering any action. This separation ensures robustness to transient misclassifications at the single-camera level and allows the alert policy to be tuned independently of the underlying inference models.

Figure 4.4 shows the logical architecture of the proposed real-time system deployed on board a public-transport vehicle. The upper layer implements a per-camera streaming pipeline, independently instantiated for each camera $i = 1, \ldots, N$, which converts continuous RTSP video streams into clip-level violence scores through buffering, clip extraction, pre-processing, and deep neural network inference. The lower layer operates at system level and aggregates asynchronous clip-level scores using a stateful decision logic driven by a global scheduler, enforcing thresholding and temporal persistence constraints before triggering alerts or logging evidences. Importantly, clip-level scores do not directly generate alarms, but are mediated by the system-level decision module.

4.6.1 Video Acquisition Pipeline

The acquisition layer is responsible for ingesting synchronized (or at least time-stamped) frames from multiple IP cameras and delivering them to the buffering

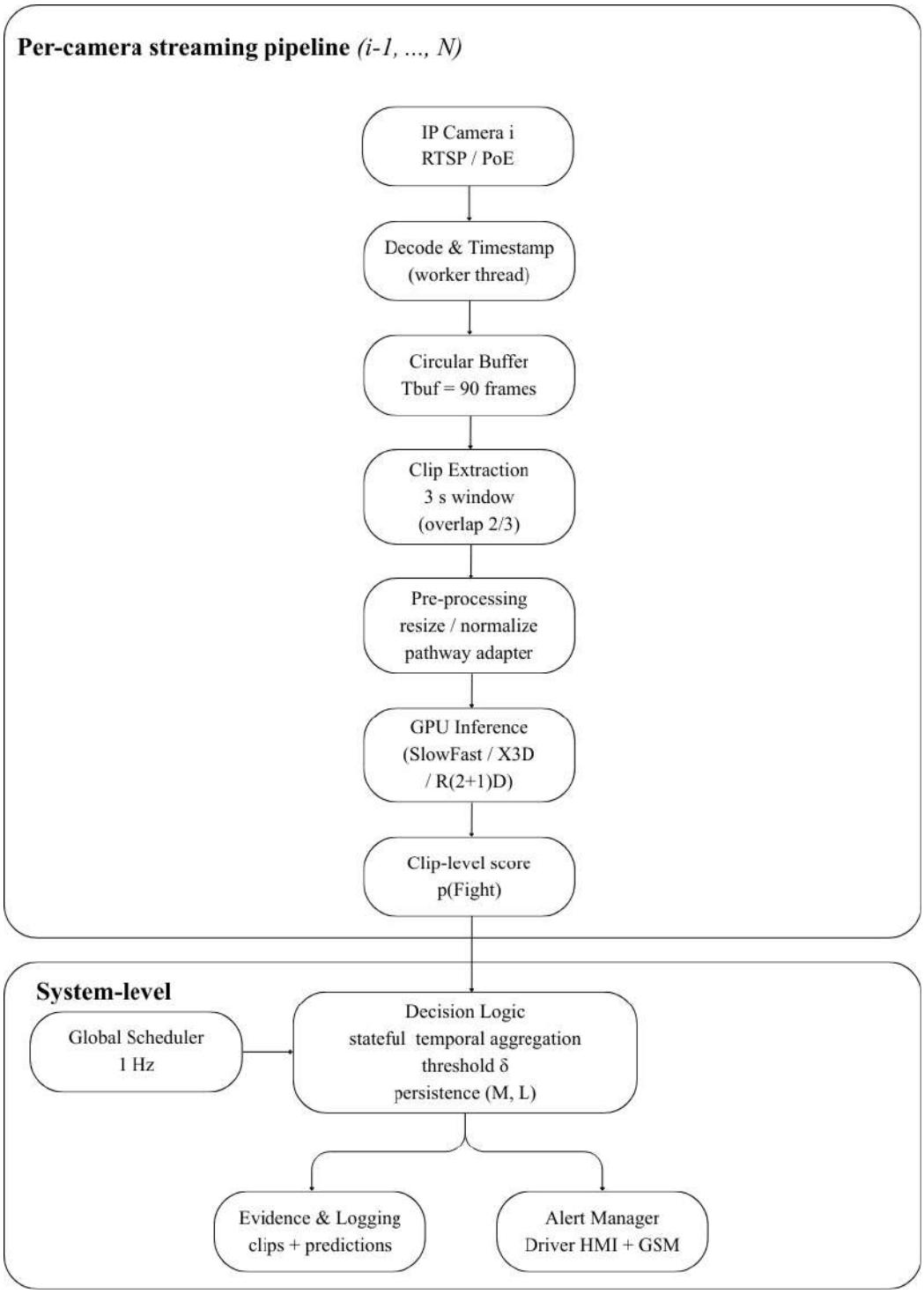


Figure 4.4: *Software architecture of the proposed real-time multi-camera violence detection pipeline.*

subsystem with bounded jitter. In the proposed setup, the cameras provide continuous streams that are captured at a nominal rate of 30 fps . The acquisition software is implemented according to a *multi-producer* paradigm: each camera is handled by a dedicated worker (thread/process) that performs (1) stream connection and decoding, (2) frame timestamping, and (3) enqueueing into the camera-specific buffer. This design isolates network/decoder stalls to the affected stream and prevents head-of-line blocking across cameras, which is critical when multiple views must be processed concurrently.

Let $i \in \{1, \ldots, N\}$ index the cameras ($N = 6$ in the full configuration). The acquisition worker for camera i outputs a frame sequence $\{\mathbf{F}_t^{(i)}\}_{t \in \mathbb{N}}$ with associated timestamps $\{s_t^{(i)}\}$. Although the inference stage is performed per-stream (Section 4.6.3), timestamping remains important for: (i) monitoring stream health (frame drops, stalls), (ii) optionally enforcing a target sampling rate via decimation/resampling when sources exhibit variable frame rate, and (iii) supporting temporal consistency checks across cameras with partially overlapping fields of view.

These requirements naturally extend to robustness at the acquisition layer, particularly in the presence of network variability and decoding jitter typical of mobile RTSP/IP deployments. Given the use of RTSP/IP streams in a mobile environment, the acquisition layer is designed to tolerate transient network instabilities and variable decoding delays. In the implemented prototype, each camera worker operates independently and maintains bounded internal queues to prevent unbounded memory growth. Frame dropping policies favor recency over completeness, ensuring that the downstream buffering stage always reflects the most recent observable scene dynamics. This design choice is aligned with the real-time objective of the system, where timely detection is prioritized over exhaustive frame retention.

4.6.2 Pre-Processing and Frame Buffering

The buffering subsystem translates a continuous stream into a sequence of short clips suitable for clip-level action recognition. Following the real-time strategy adopted in our system, each camera stream is associated with a *fixed-size circular buffer* of length T_{buf} frames. Since the cameras operate at 30 fps, we set $T_{\text{buf}} = 90$ frames, corresponding to a temporal context of 3 s. This temporal horizon is a deliberate compromise: it is long enough to include the onset and evolution of aggressive interactions (e.g., escalation from gesturing to physical contact), yet short enough to satisfy end-to-end latency constraints under embedded inference.

To balance temporal continuity with computational efficiency, the buffer is updated at a cadence of 1 s: every update cycle, 30 new frames are appended and the oldest 30 frames are discarded, so that 60 frames (2 s) are retained from the previous window . Denoting by $\Delta = 30$ the update step (in frames), the *overlap*

ratio is:

$$\rho \;=\; \frac{T_{\text{buf}} - \Delta}{T_{\text{buf}}} \;=\; \frac{90 - 30}{90} \;=\; \frac{2}{3}. \tag{4.10}$$

This overlap introduces temporal continuity between successive clips and reduces fragmentation of short actions across window boundaries, improving stability of predictions in streaming conditions. Operationally, the clip extracted at update n from camera i can be expressed as:

$$\mathbf{X}_n^{(i)} = \left[\mathbf{F}_{t_n - T_{\text{buf}}+1}^{(i)}, \ldots, \mathbf{F}_{t_n}^{(i)}\right] \in \mathbb{R}^{T_{\text{buf}} \times H \times W \times C}, \tag{4.11}$$

where t_n is the index of the most recent frame at update n and (H, W, C) are the frame height, width and channels.

Consistent pre-processing is then applied to each extracted clip to ensure alignment with the training protocol and stable inference behavior. Before inference, each buffered clip undergoes a lightweight pre-processing stage (Figure 2.1), including: spatial resizing/cropping to the normalized training resolution, tensor layout conversion, and intensity normalization consistent with the training protocol. The resulting tensor is then forwarded to the model-specific input adapter: (i) SlowFast receives two temporally resampled pathways derived from the same clip (Section 4.1.1), while (ii) X3D-L and R(2+1)D process a single clip tensor (Sections 4.1.2–4.1.3). A key systems requirement is that pre-processing remains strictly bounded in time; thus, all operations are chosen to be linear in the number of pixels/frames and amenable to vectorized implementations.

From a systems perspective, this clip extraction and pre-processing pipeline must also preserve data integrity under concurrent acquisition and inference. To prevent contention between acquisition (writes) and inference (reads), the implementation should snapshot the circular buffer into a contiguous clip tensor at each update (double-buffering), or otherwise use lock-free index management with memory barriers. This guarantees that inference always operates on a consistent temporal window, avoiding mixed-frame clips that could degrade classification reliability.

4.6.3 Inference Engine and Real-Time Decision Logic

The inference engine executes clip-level action recognition on the buffered clips and translates model outputs into operational decisions (alerts, logging, evidence retention), as summarized in the software architecture shown in Fig. 4.4. In the current prototype, inference is performed independently for each camera stream at the same cadence as the buffer update (1 Hz), i.e., one 3 s clip per second per camera. This yields an *online sliding-window classifier* whose temporal resolution is 1 s and whose receptive field is 3 s. The overall real-time processing pipeline implemented in the prototype is detailed in Algorithm 2, which formalizes the interaction between acquisition workers, circular buffering, clip construction, inference, and alert dispatch across multiple concurrent camera streams.

Algorithm 2 Real-Time Multi-Camera Inference with Sliding Buffer and Alerting

Require: Number of cameras N; buffer length $T_{\text{buf}} = 90$ frames; update stride $\Delta = 30$ frames (≈ 1 s at 30 fps); threshold $\delta \in (0,1)$; persistence parameters (M, L); trained model f_θ

Initialize RTSP connections for all cameras; start N acquisition workers
for $i = 1$ to N **do**
    Initialize circular buffer $\mathcal{B}^{(i)} \leftarrow \emptyset$ with capacity T_{buf}
    Initialize decision history $\mathcal{H}^{(i)} \leftarrow \emptyset$ with capacity L
end for
Load model f_θ on the embedded GPU; initialize logging/storage and GSM/HMI modules
while system is ON **do**
    /* Scheduler tick every Δ frames (1 Hz) */
    for $i = 1$ to N **do**
        Acquire latest Δ frames $\{\mathbf{F}^{(i)}\}$ from worker i (timestamped)
        Update buffer: $\mathcal{B}^{(i)} \leftarrow \text{PushDrop}(\mathcal{B}^{(i)}, \{\mathbf{F}^{(i)}\}, T_{\text{buf}})$
        if $|\mathcal{B}^{(i)}| < T_{\text{buf}}$ **then**
            continue ▷ warm-up until buffer is full
        end if
        Construct clip $\mathbf{X}^{(i)} \leftarrow \text{Snapshot}(\mathcal{B}^{(i)})$
        Pre-process $\tilde{\mathbf{X}}^{(i)} \leftarrow \text{Preprocess}(\mathbf{X}^{(i)})$ ▷ resize/normalize + pathway adapter
        Compute probabilities $\mathbf{p}^{(i)} \leftarrow \text{Softmax}(f_\theta(\tilde{\mathbf{X}}^{(i)}))$
        $s^{(i)} \leftarrow \mathbf{p}^{(i)}(\text{Fight})$ ▷ Fight confidence score
        Append $[s^{(i)} \geq \delta]$ to history $\mathcal{H}^{(i)}$
        Log $(i, s^{(i)}, \arg\max \mathbf{p}^{(i)}, t)$
        if $\sum \mathcal{H}^{(i)} \geq M$ **then**
            Save evidence clip (optionally include pre/post context)
            Trigger local HMI alert
            Send GSM alert with metadata and clip reference/path
            Reset or decay $\mathcal{H}^{(i)}$ to avoid repeated alerts
        end if
    end for
end while

To assess whether the online inference pipeline described in Algorithm 2 satisfies real-time constraints under embedded deployment, we explicitly analyze the per-cycle latency budget of the system. . Let t_{acq}, t_{buf}, t_{prep}, and t_{inf} denote the time spent in acquisition, buffering/snapshotting, pre-processing, and model inference, respectively, for a given camera at one update cycle. Real-time operation requires:

$$t_{\text{buf}} + t_{\text{prep}} + t_{\text{inf}} \;\leq\; T_{\text{cycle}} \;=\; 1\text{ s}, \tag{4.12}$$

since acquisition runs continuously in parallel. Empirically, the measured end-to-end processing time per camera stream (including acquisition, buffering, preprocessing, and model prediction) lies in the range 620–950 ms in the bus simulator, confirming feasibility under the chosen cadence. This result is central: it demonstrates that modern action-recognition backbones can be integrated into an embedded, multi-stream pipeline without violating safety-critical response times.

Beyond feasibility, the structure of the model outputs is critical for downstream

decision-making and alert handling within the modular architecture depicted in Fig. 4.4. Given the clip $\mathbf{X}_n^{(i)}$ for camera i at update n, the model outputs logits $\mathbf{z}_n^{(i)} \in \mathbb{R}^K$ (with $K = 2$ for Fight/No-Fight). Probabilities are obtained via softmax:

$$p_n^{(i)}(k) \;=\; \frac{\exp(z_{n,k}^{(i)})}{\sum_{j=1}^{K} \exp(z_{n,j}^{(i)})}. \tag{4.13}$$

The instantaneous predicted label is $\hat{y}_n^{(i)} = \arg\max_k p_n^{(i)}(k)$, while $p_n^{(i)}(\text{Fight})$ provides a confidence score that can be exploited for thresholding and alert prioritization.

These probabilistic predictions are then consumed by a lightweight decision layer, whose logic is instantiated in Algorithm 2 and implemented in the alerting module of the software architecture shown in Fig. 4.4.

A minimal decision rule triggers an alarm whenever $\hat{y}_n^{(i)} = \text{Fight}$ for at least one stream. In practice, it is often beneficial to introduce a tunable confidence threshold $\delta \in (0,1)$ and a short persistence criterion (e.g., M positives over the last L updates) to control false alarms in visually ambiguous situations:

$$\text{ALERT}_n = 1 \iff \exists i : \; p_n^{(i)}(\text{Fight}) \geq \delta \;\wedge\; \text{Persist}(i,n) \geq M. \tag{4.14}$$

While the specific parameters (δ, M, L) are deployment-dependent, the proposed software architecture explicitly supports this form of post-processing by maintaining a *structured output buffer indexed by camera.*

Upon detection, the system actions defined in Algorithm 2 are executed, including evidence retention, GSM alert dispatch, and on-board HMI updates, consistently with the modular flow outlined in Fig. 4.4. . This design enables immediate notification even under limited connectivity, while keeping the heavy video payload local by default—an approach aligned with edge-surveillance architectures that prioritize low-latency response and reduced network load [56, 36, 47]. Finally, the on-board HMI (driver-facing display) can be updated to provide an immediate local cue, allowing human intervention when appropriate.

Chapter 5

Experimental Results

5.1 Training Results and Model Selection

Before reporting real-world trials, we provide an in-depth analysis of the offline performance obtained after training the three candidate video backbones—SlowFast R50, X3D-L, and R(2+1)D—on the normalized violence-detection datasets described in the previous chapters. This section is intentionally positioned before the field validation because the on-board deployment necessarily requires a *principled* model choice grounded on controlled, reproducible evidence: (i) offline testing offers a standardized benchmark to compare architectures under the same preprocessing, clip duration, and training protocol; (ii) it exposes the characteristic error modes of each backbone (e.g., systematic false alarms on fast non-violent gestures, or missed detections under occlusion and far-field views), which is essential in a safety-critical scenario where different failure types have different operational consequences; and (iii) it connects recognition accuracy to system-level constraints, since the selected model must remain compatible with the embedded inference budget under multi-camera load. The task is formulated as binary clip-level action recognition (Fight vs. No-Fight), in line with early person-to-person violence recognition studies and with modern large-scale action recognition practice. [24, 23] Importantly, because violence detection is typically characterized by class imbalance and by hard negatives that are visually and temporally similar to the target class, we do not rely on accuracy alone; instead, we emphasize class-conditional metrics (precision/recall/F1 on the Fight class), macro-averaged indicators, and threshold-sweeping measures (PR-AUC and ROC-AUC), which are commonly recommended when the positive class is underrepresented and the false-alarm trade-off is operationally relevant.

This section analyzes the impact of two fundamental training choices on violence recognition performance: (i) the initialization strategy, i.e., training from scratch versus fine-tuning from large-scale action recognition pretraining, and (ii)

Trained and tested on public datasets				
Model	**Class**	**Precision**	**Recall**	**F1-Score**
SlowFast50	Fight	0.85	0.72	0.78
	NoFight	0.90	0.77	0.83
R(2+1)D	Fight	0.78	0.70	0.74
	NoFight	0.84	0.82	0.83
X3D	Fight	0.89	0.84	0.86
	NoFight	0.93	0.91	0.92
Trained on public datasets, tested on laboratory dataset				
SlowFast50	Fight	0.38	0.54	0.45
	NoFight	0.86	0.78	0.82
R(2+1)D	Fight	0.41	0.43	0.42
	NoFight	0.83	0.82	0.83
X3D	Fight	0.57	0.61	0.59
	NoFight	0.91	0.89	0.90

Table 5.1: *Comparison between public-only evaluation and laboratory-domain evaluation, highlighting the impact of domain shift on Fight-class performance.*

the dataset composition, i.e., training on public datasets alone versus incorporating domain-specific laboratory data as described in Chapter 3. These factors are examined through controlled offline experiments, whose outcomes directly inform model selection for on-board deployment. In this context, we first investigate the effect of pretraining compared to training from scratch. Table 5.2 reports a systematic ablation study in which all candidate backbones are trained on the same full dataset, either initialized from Kinetics-400 pretrained weights or trained from random initialization.

Across all architectures, pretraining yields a consistent improvement in performance, with particularly strong gains on the Fight class. In absolute terms, the Fight-class F1-score improves by approximately 6–8 percentage points when pretraining is employed, corresponding to a relative improvement on the order of 7–10% depending on the backbone. Similar trends are observed for Fight recall, indicating that pretraining primarily enhances sensitivity to aggressive interactions rather than merely improving confidence calibration on the dominant No-Fight class.

From a representation-learning perspective, this behavior is expected. Violent interactions are characterized by complex spatio-temporal patterns involving body motion, proximity, and contact dynamics, which are difficult to learn from scratch in datasets of limited size. Pretraining on Kinetics-400 provides a rich initialization

Pretrained on Kinetics-400				
Model	**Class**	**Precision**	**Recall**	**F1-Score**
SlowFast50	Fight	0.90	0.87	0.88
	NoFight	0.93	0.91	0.92
R(2+1)D	Fight	0.89	0.86	0.87
	NoFight	0.92	0.93	0.93
X3D	Fight	0.91	0.90	0.90
	NoFight	0.94	0.92	0.93
Trained from scratch				
SlowFast50	Fight	0.84	0.80	0.82
	NoFight	0.87	0.85	0.86
R(2+1)D	Fight	0.83	0.79	0.81
	NoFight	0.86	0.87	0.87
X3D	Fight	0.85	0.84	0.84
	NoFight	0.88	0.87	0.87

Table 5.2: *Impact of Kinetics-400 pretraining on validation performance. Pretraining consistently improves Fight-class detection across all backbones.*

of motion- and interaction-aware features, which can then be specialized to the bus surveillance domain during fine-tuning. As a result, pretrained models converge faster, reach higher validation plateaus, and exhibit more stable generalization behavior compared to their randomly initialized counterparts.

Effect of dataset composition and domain shift. A second axis of analysis concerns the impact of training data composition. Table 5.1 compares models trained exclusively on public datasets with their performance when evaluated on clips from the proprietary laboratory dataset. When trained and tested on public data only, all architectures achieve strong performance, confirming their ability to fit the source domains.

However, when the same models are evaluated on laboratory clips simulating full-bus scenarios, a substantial performance degradation is observed. In particular, Fight-class F1-scores drop by more than 30 percentage points in some configurations, revealing a severe domain shift between public datasets and the target deployment environment, consistent with prior findings in public transport surveillance [7].

Importantly, incorporating laboratory data during training markedly mitigates this gap. As reported in Table 5.3, models trained on the combined dataset (public

Model	Training Set	F1-Score	Accuracy
SlowFast50	Public only	0.63	0.68
	Public + Laboratory	**0.86**	**0.89**
R(2+1)D	Public only	0.62	0.67
	Public + Laboratory	**0.85**	**0.87**
X3D	Public only	0.68	0.72
	Public + Laboratory	**0.88**	**0.90**

Table 5.3: *Performance on 100 laboratory test clips, highlighting the benefit of including domain-specific data during training.*

+ laboratory) significantly outperform their public-only counterparts when evaluated on the proprietary test set. Across backbones, the inclusion of laboratory data yields absolute improvements in F1-score of approximately 20–25 percentage points, corresponding to relative gains exceeding 30%. These results demonstrate that domain-specific data capturing realistic viewpoints, crowding conditions, and image quality are essential for robust violence detection in public transport environments.

Taken together, these experiments establish two key conclusions that guide the remainder of this chapter. First, pretraining on large-scale action recognition datasets is a critical enabler for learning reliable spatio-temporal representations of violent behavior. Second, the inclusion of laboratory data aligned with the target deployment scenario is indispensable to achieve acceptable generalization performance. Both design choices are therefore retained in all subsequent analyses and directly motivate the deployment-oriented model selection discussed later in this chapter.

5.1.1 Learning Dynamics and Convergence

Having established the impact of pretraining and dataset composition on final performance, we now analyze the learning dynamics that lead to these outcomes and justify the observed performance gaps.

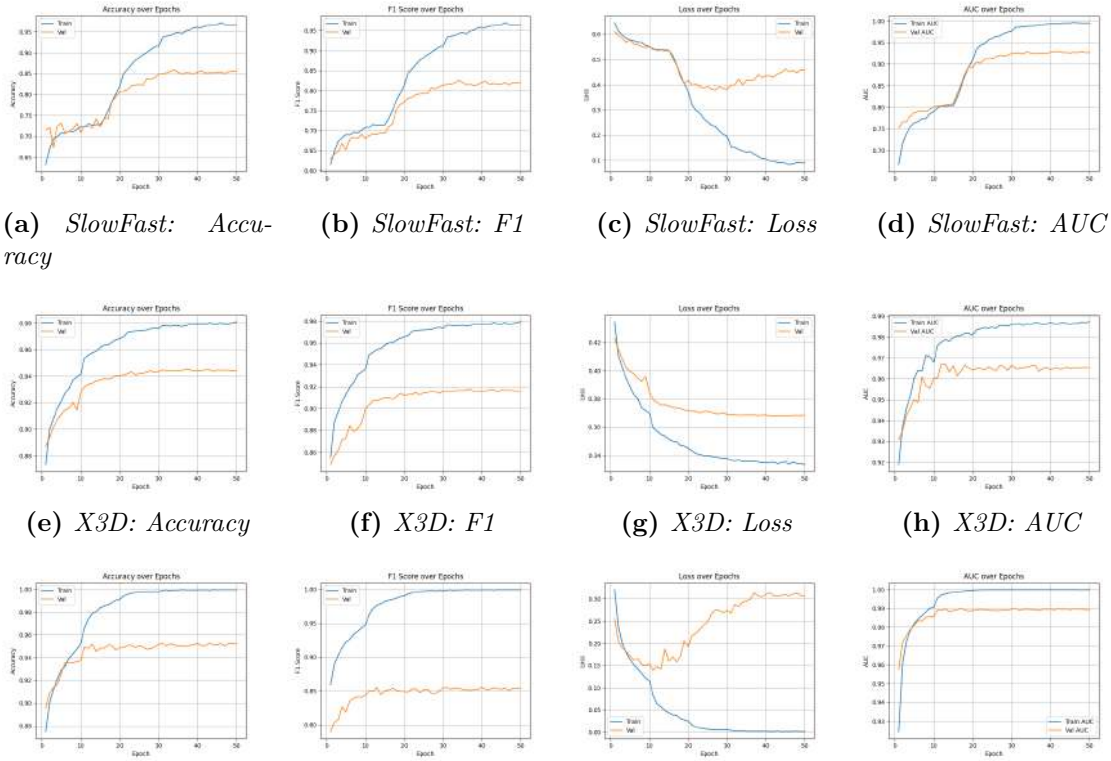
Figures 5.1 and 5.2 report the global learning dynamics for all evaluated backbones trained from scratch and fine-tuned from Kinetics-400 pretrained weights, respectively. Each figure adopts a compact multi-panel layout that summarizes accuracy, F1-score, loss, and AUC over training epochs, jointly capturing convergence speed, optimization stability, and class-separation capability.

While global metrics provide a high-level view of convergence, they may mask class-specific effects in imbalanced violence detection. Figure 5.4 therefore reports class-wise learning dynamics for No-Fight (class 0) and Fight (class 1), explicitly highlighting how different training strategies affect minority-class learning. The

class-wise curves show that pretraining yields a pronounced and consistent benefit on the Fight class, with earlier improvements and more stable trajectories in F1-score, recall, and loss. In contrast, training from scratch often results in noisier curves and delayed learning of the minority class, even when overall accuracy appears satisfactory. These observations directly explain and support the quantitative gains reported in Table 5.2.

Backbone-specific trends are also evident. SlowFast R50 exhibits rapid early improvements, particularly in Fight-class recall, reflecting its strong motion sensitivity, but may show earlier saturation or mild overfitting. X3D-L converges more gradually and displays the most stable validation behavior across all metrics, especially in terms of AUC and Fight-class F1. R(2+1)D shows intermediate dynamics, with competitive performance but greater variability across epochs, consistent with its reduced robustness under domain shift.

The final checkpoint used for test-set evaluation (Section 5.1.2) is selected according to the best validation Fight-class F1 or macro-F1, ensuring that the offline comparison reflects a deployment-oriented operating point rather than an arbitrary training epoch.

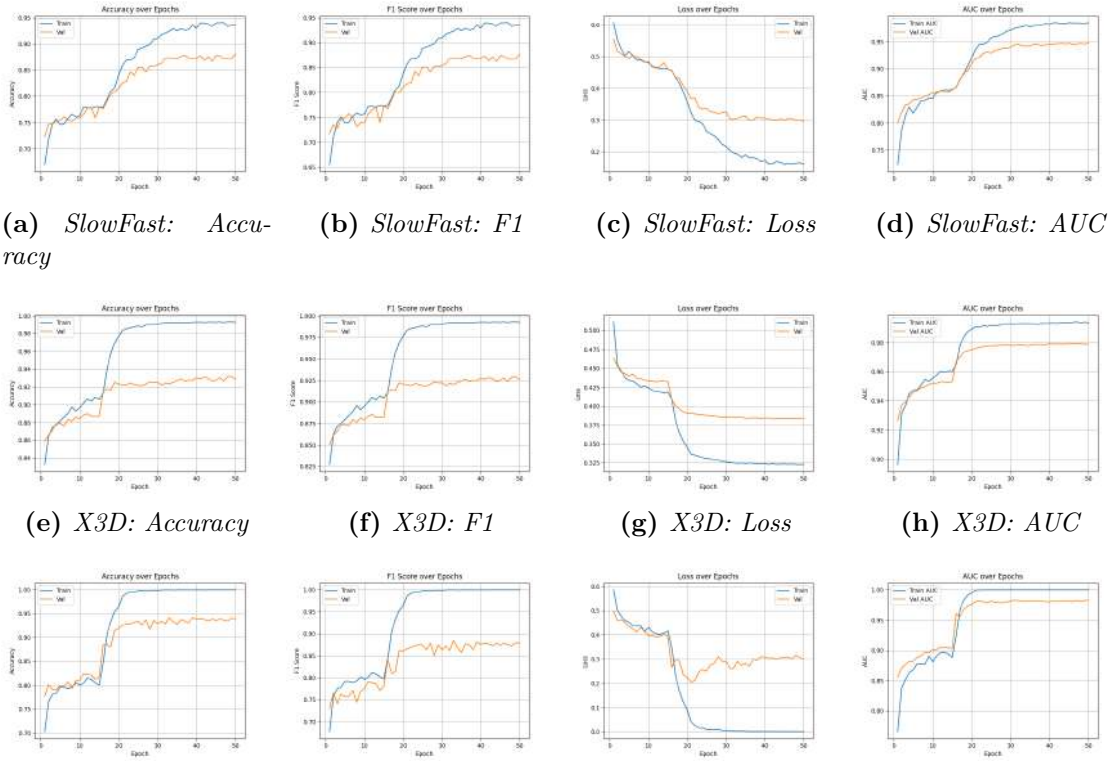


(a) *SlowFast: Accuracy* **(b)** *SlowFast: F1* **(c)** *SlowFast: Loss* **(d)** *SlowFast: AUC*

(e) *X3D: Accuracy* **(f)** *X3D: F1* **(g)** *X3D: Loss* **(h)** *X3D: AUC*

(i) *R(2+1)D: Accuracy* **(j)** *R(2+1)D: F1* **(k)** *R(2+1)D: Loss* **(l)** *R(2+1)D: AUC*

Figure 5.1: *Learning dynamics for models trained from scratch (no pretraining). Each subplot reports training and validation curves for the corresponding metric.*

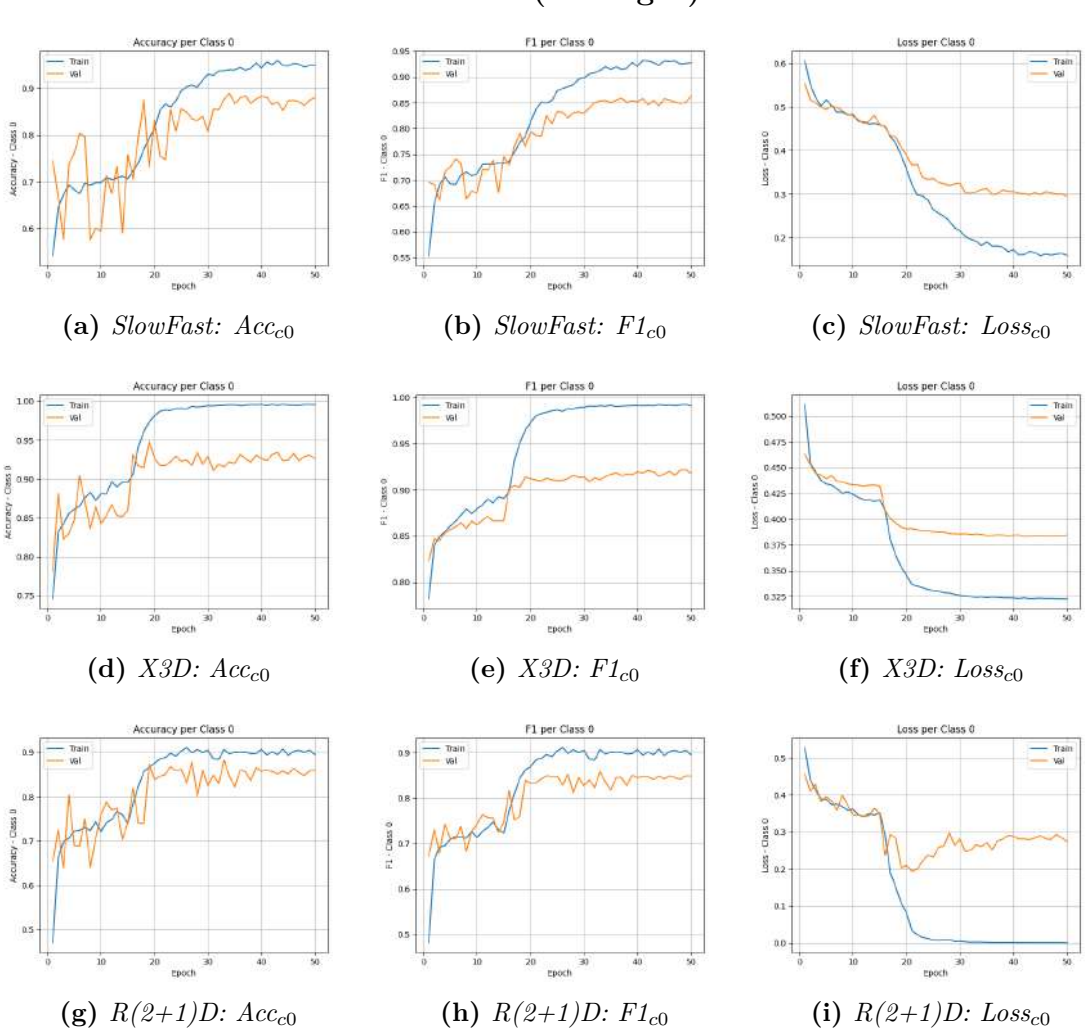


(a) *SlowFast: Accuracy* **(b)** *SlowFast: F1* **(c)** *SlowFast: Loss* **(d)** *SlowFast: AUC*

(e) *X3D: Accuracy* **(f)** *X3D: F1* **(g)** *X3D: Loss* **(h)** *X3D: AUC*

(i) *R(2+1)D: Accuracy* **(j)** *R(2+1)D: F1* **(k)** *R(2+1)D: Loss* **(l)** *R(2+1)D: AUC*

Figure 5.2: *Learning dynamics for models fine-tuned from Kinetics-400 pretrained weights. Training and validation curves are shown for each metric.*

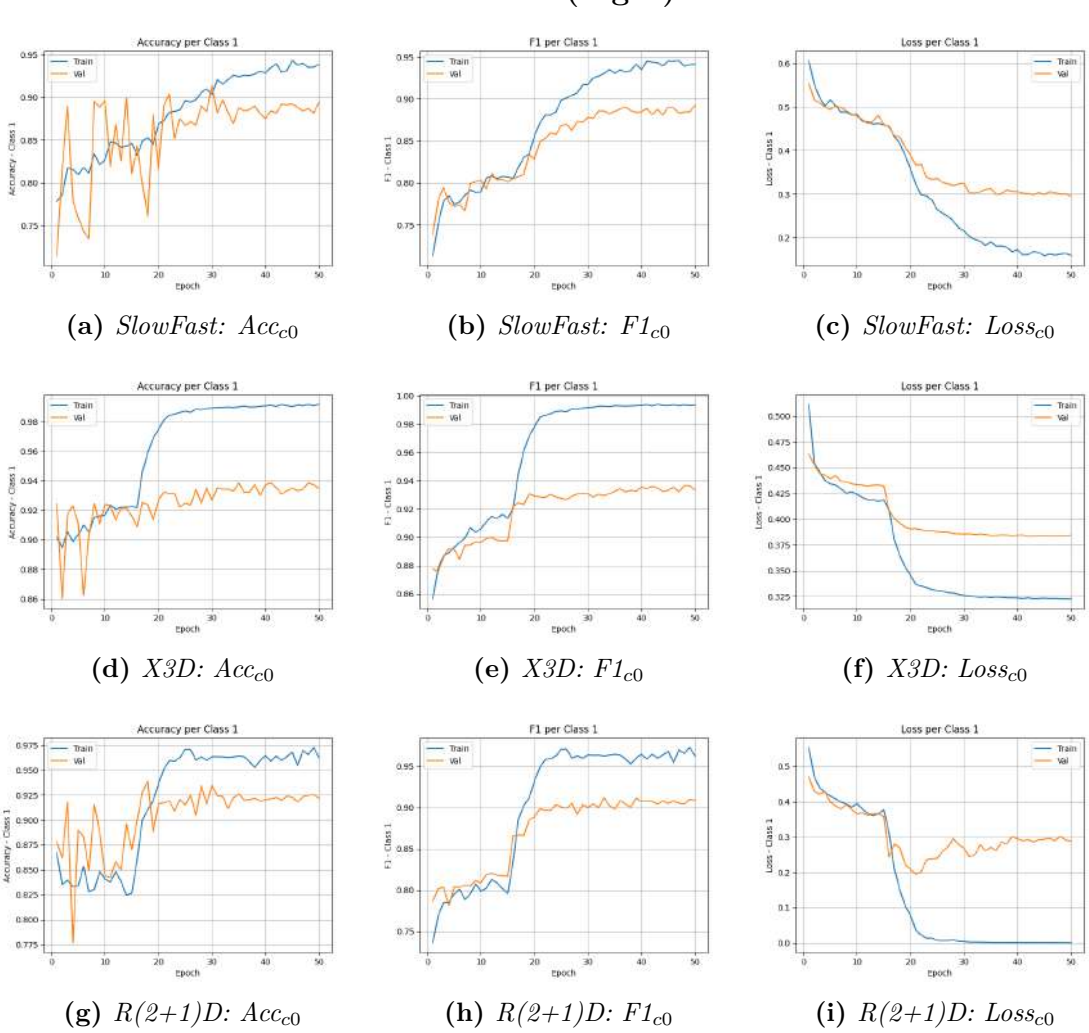


(a) *SlowFast:* Acc_{c0} **(b)** *SlowFast:* $F1_{c0}$ **(c)** *SlowFast:* $Loss_{c0}$

(d) *X3D:* Acc_{c0} **(e)** *X3D:* $F1_{c0}$ **(f)** *X3D:* $Loss_{c0}$

(g) *R(2+1)D:* Acc_{c0} **(h)** *R(2+1)D:* $F1_{c0}$ **(i)** *R(2+1)D:* $Loss_{c0}$

Figure 5.3: *Class-wise learning dynamics for No-Fight (class 0). Each subplot reports training and validation curves, highlighting the effect of pretraining on minority-class learning.*



(a) *SlowFast:* Acc_{c0} **(b)** *SlowFast:* $F1_{c0}$ **(c)** *SlowFast:* $Loss_{c0}$

(d) *X3D:* Acc_{c0} **(e)** *X3D:* $F1_{c0}$ **(f)** *X3D:* $Loss_{c0}$

(g) *R(2+1)D:* Acc_{c0} **(h)** *R(2+1)D:* $F1_{c0}$ **(i)** *R(2+1)D:* $Loss_{c0}$

Figure 5.4: *Class-wise learning dynamics for Fight (class 1). Each subplot reports training and validation curves, highlighting the effect of pretraining on minority-class learning.*

5.1.2 Quantitative Comparison on the Test Set

The quantitative comparison on the held-out test set complements the analysis of learning dynamics by assessing how the different backbones generalize once training has converged. All models are evaluated under identical clip normalization and preprocessing conditions (Chapter 4) and using checkpoints selected according to the validation criteria discussed in Section 5.1.1, namely the maximization of Fight-class F1 or macro-F1 under the configuration that includes Kinetics-400 pretraining and laboratory data.

Rather than relying on a single aggregate score, the comparison is framed in a deployment-oriented perspective, where different metrics capture distinct operational requirements. Fight recall and Fight F1 quantify sensitivity to aggressive behavior and represent the primary safety objective, as missed detections correspond to the most critical failure mode in public-transport surveillance. Precision reflects alert reliability and operational cost, since excessive false positives may lead to alert fatigue and reduced trust in the system. Threshold-independent indicators such as PR-AUC are used to assess the quality of confidence ranking under class imbalance, while ROC-AUC is retained as a conventional complementary metric, in line with established best practices for imbalanced binary classification.

Across all evaluated configurations, clear and consistent trends emerge. X3D-L exhibits the most favorable balance between sensitivity and selectivity, achieving strong Fight-class detection while maintaining controlled false-positive behavior. SlowFast R50 reaches competitive levels of Fight recall, confirming its effectiveness in capturing motion-driven aggression cues, but shows a less favorable precision–recall trade-off, consistent with its higher tendency to over-trigger on ambiguous non-violent motion. R(2+1)D, while effective in controlled settings, demonstrates reduced class-separation capability under the tested configuration, aligning with its less stable learning dynamics and increased sensitivity to domain shift.

Importantly, these observations reinforce the notion that no single metric is sufficient to support model selection in a safety-critical application. A marginal improvement in overall accuracy would not compensate for a systematic reduction in Fight recall, just as an aggressive recall-oriented operating point would be unacceptable if accompanied by an excessive false-positive rate. Instead, meaningful comparison requires a joint interpretation of sensitivity, reliability, and confidence ranking.

The quantitative trends observed on the test set are fully consistent with the convergence behavior reported in the learning curves (Section 5.1.2) and with the error structure revealed by confusion matrices and qualitative analysis (Sections 5.1.4–5.1.5). Models exhibiting smoother convergence and stronger minority-class learning also demonstrate superior generalization on the held-out test set, providing a coherent and deployment-relevant basis for backbone selection.

5.1.3 Confusion Matrices and Class-Wise Behavior

While scalar metrics and learning curves characterize average performance and convergence behavior, confusion matrices provide a more operationally meaningful view of how each backbone distributes its errors at the selected operating point. Figure 5.5 reports the normalized confusion matrices computed at the best validation epoch for each model, offering insight into the balance between missed detections (false negatives) and spurious alerts (false positives).

Across all architectures, a common trend emerges: errors are not uniformly distributed, but are strongly conditioned by visual factors such as distance from the camera and scene occlusion. In particular, all models exhibit an increasing tendency toward false negatives as the distance between the interacting subjects and the camera increases. Empirically, beyond approximately 3 meters, aggressive interactions are more frequently missed, reflecting the progressive loss of fine-grained motion cues and body-part articulation at far-field resolutions.

Occlusions represent a second major contributor to false negatives. Partial or full occlusions—caused either by other passengers in crowded scenes or by large foreground objects—attenuate discriminative limb motion and temporal continuity, leading the models to underestimate or entirely miss violent interactions. This effect is visible across all backbones and becomes more pronounced in high-density scenarios.

False positives, on the other hand, are primarily associated with ambiguous but non-violent actions, rather than random noise. A recurrent failure mode involves actions that include raising or extending the arms, such as putting on a jacket, adjusting clothing, or gesturing during conversation. These movements, even when slow, can resemble aggressive motion patterns at the spatiotemporal level, especially when they occur within close range (typically within 3 meters), where motion amplitude and body detail are more salient. Strong occlusions due to crowding or large obstructions can also trigger false positives by fragmenting visual evidence and amplifying motion uncertainty.

Within this shared error structure, backbone-specific behaviors remain evident. SlowFast R50 shows strong sensitivity to aggressive motion but a higher propensity to over-trigger on high-energy non-violent gestures, consistent with its motion-focused inductive bias. X3D-L achieves a more balanced error distribution, maintaining a lower false-positive rate while limiting false negatives under moderate occlusion and distance. R(2+1)D presents a higher concentration of false negatives under visually degraded conditions, aligning with its less stable learning dynamics and reduced robustness to domain shift.

By explicitly decomposing errors into false positives and false negatives and linking them to concrete visual conditions, the confusion matrices provide a crucial bridge between quantitative performance indicators and qualitative failure analysis, directly informing deployment-oriented design choices.

(a) *SlowFast: Overall* **(b)** *X3D: Overall* **(c)** *R(2+1)D: Overall*

Figure 5.5: *Confusion matrices at the best validation epoch for each backbone. Rows report the overall matrix and the class-wise views (class 0: No-Fight, class 1: Fight), while columns correspond to SlowFast R50, X3D-L, and R(2+1)D.*

5.1.4 Qualitative Error Analysis: TP/FP/TN/FN Examples

Quantitative metrics and confusion matrices are complemented by a qualitative inspection of representative True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) examples extracted from the test set. Figures 5.6, 5.7, and 5.8 present structured visual grids for each backbone, with three representative examples per error category.

True positives consistently correspond to clips in which physical contact and antagonistic intent are clearly visible, typically occurring within an effective working distance from the camera. In these cases, limb interactions, body orientation, and temporal continuity provide sufficient evidence for reliable classification across all models.

False negatives exhibit two dominant patterns. The first is far-field interaction, where subjects are located beyond approximately 3 meters from the camera. In this regime, reduced spatial resolution and diminished motion saliency impair the extraction of discriminative cues. The second pattern involves partial or strong occlusions, either due to dense passenger crowds or large intervening objects, which fragment or suppress critical motion signals. These factors often co-occur in realistic bus scenarios, compounding their negative effect.

False positives are predominantly driven by hard negatives rather than random misclassification. A frequent trigger is non-violent actions involving arm elevation or extension, such as wearing a jacket, adjusting clothing, or expressive gesturing. Even when performed slowly, these actions can generate motion patterns that resemble aggressive behavior at the clip level, particularly in near-field views. Additional false positives arise in scenes with severe occlusion, where incomplete visual evidence leads to overconfident predictions.

True negatives largely correspond to static or smoothly evolving scenes without abrupt motion or close-range physical interaction, confirming that the models do

not trivially over-trigger in the absence of salient motion cues.

Comparing backbones reveals consistent qualitative differences. SlowFast R50 tends to over-react to energetic but non-violent movements, reflecting its strong motion sensitivity. X3D-L shows a more conservative behavior in ambiguous scenarios while preserving sensitivity to genuine aggression. R(2+1)D exhibits a higher proportion of missed detections in far-field or occluded conditions.

These qualitative observations are fully aligned with the confusion matrices and reinforce the interpretation of the learning dynamics and quantitative results.

(a) *TP-1* **(b)** *TP-2* **(c)** *TP-3*

(d) *FP-1* **(e)** *FP-2* **(f)** *FP-3*

(g) *TN-1* **(h)** *TN-2* **(i)** *TN-3*

(j) *FN-1* **(k)** *FN-2* **(l)** *FN-3*

Figure 5.6: *Qualitative error analysis for SlowFast R50: three examples each for TP, FP, TN, and FN. Each panel should report ground truth, predicted label, and Fight confidence score.*

(a) *TP-1* **(b)** *TP-2* **(c)** *TP-3*

(d) *FP-1* **(e)** *FP-2* **(f)** *FP-3*

(g) *TN-1* **(h)** *TN-2* **(i)** *TN-3*

(j) *FN-1* **(k)** *FN-2* **(l)** *FN-3*

Figure 5.7: *Qualitative error analysis for X3D-L: three examples each for TP, FP, TN, and FN.*

(a) *TP-1* **(b)** *TP-2* **(c)** *TP-3*

(d) *FP-1* **(e)** *FP-2* **(f)** *FP-3*

(g) *TN-1* **(h)** *TN-2* **(i)** *TN-3*

(j) *FN-1* **(k)** *FN-2* **(l)** *FN-3*

Figure 5.8: *Qualitative error analysis for R(2+1)D: three examples each for TP, FP, TN, and FN.*

5.1.5 Model Selection for Deployment

The final backbone selected for on-board deployment is determined through a deployment-oriented trade-off grounded in the experimental evidence presented in this chapter. The decision jointly accounts for recognition effectiveness, error structure, and embedded feasibility.

From a recognition standpoint, reliable detection of the Fight class under realistic operating conditions is the primary requirement. The qualitative and confusion-matrix analyses demonstrate that all models are affected by distance and occlusion, but with different degrees of robustness. X3D-L consistently exhibits the most balanced behavior, limiting false negatives beyond close range while avoiding excessive false positives under ambiguous motion.

From an operational standpoint, false positives must remain bounded to prevent alert fatigue and preserve trust in the system. In this respect, X3D-L shows a clear advantage, as its conservative response to arm-raising gestures and occluded scenes reduces the likelihood of spurious alerts. SlowFast R50, while competitive in terms of raw sensitivity, displays a higher propensity to over-trigger on non-violent but dynamic actions, making it less suitable for conservative deployment without aggressive post-processing. R(2+1)D, although effective in controlled settings, demonstrates reduced robustness under visually degraded conditions.

Finally, from a systems standpoint, the selected model must satisfy the latency and throughput constraints imposed by multi-camera, real-time operation on an embedded GPU platform, as discussed in the software architecture and real-time strategy sections.

Taken together, the evidence indicates that X3D-L offers the most favorable compromise between Fight sensitivity, alert reliability, and robustness to real-world perturbations such as distance, occlusion, and crowding. The chosen model is therefore carried forward to the real-time processing strategy and field validation reported in the following sections, where offline performance is further challenged by operational conditions that cannot be fully reproduced in static benchmarks.

5.2 Real-Time Processing Strategy

Deploying violence recognition on-board a public-transport vehicle is not merely a matter of running a trained backbone at inference time; it requires a *streaming* design that (i) produces predictions with bounded latency, (ii) preserves temporal continuity so that short aggressive bursts are not fragmented across window boundaries, and (iii) limits bandwidth and privacy risks by avoiding continuous backhaul of raw video. In this work, the system is explicitly conceived as an *edge video analytics* pipeline, where acquisition, buffering, pre-processing and inference are executed locally on the embedded GPU server, while the communication layer

is used only to transmit compact notifications and associated metadata when required. This architectural choice aligns with established evidence that pushing video understanding toward the edge reduces network load and improves responsiveness in safety-critical monitoring, while keeping sensitive data local by default [56, 36, 47]. The operational implication is that the software stack must transform a continuous RTSP stream into a sequence of short, normalized clips suitable for clip-level action recognition, and must do so deterministically under limited compute and thermal budgets.

The adopted strategy associates each camera stream with a dedicated circular buffer used to construct fixed-duration clips at a fixed update rate (Figure 2.1). Cameras operate at a nominal 30 fps; therefore, we set a buffer size of $T_{\text{buf}} = 90$ frames, corresponding to approximately 3 s of temporal context. At system start-up, the buffer is filled with the first 90 acquired frames; afterwards, at each update cycle, 30 new frames are appended while the 60 most recent frames are retained, yielding a stride of $\Delta = 30$ frames (≈ 1 s) and a temporal overlap of 66%. This overlap is a key design decision: in real streams, violent interactions rarely align with artificial clip boundaries, and a sliding-window overlap reduces boundary effects, stabilizes predictions over time, and increases the probability that the onset and peak of an event are fully contained in at least one processed window. Operationally, the overlap ratio is

$$\rho = \frac{T_{\text{buf}} - \Delta}{T_{\text{buf}}} = \frac{90 - 30}{90} = \frac{2}{3}, \tag{5.1}$$

so that successive clips share 2 s of visual history. The system thus implements an online classifier with a 3 s receptive field and 1 s temporal granularity, which is consistent with the requirements of real-time violence monitoring where timely alerts are more critical than dense frame-by-frame labeling. Similar real-time constraints and efficiency-driven design objectives are widely discussed in the violence-recognition literature, where the core challenge is balancing motion sensitivity and deployability [49, 36].

Each extracted clip undergoes a lightweight pre-processing stage that mirrors the training normalization protocol (Chapter 4): frames are resized/cropped to the chosen input resolution, converted to tensor format, and normalized in intensity. The output of pre-processing is then adapted to the backbone-specific input interface: SlowFast requires a two-pathway representation derived from the same buffered clip (slow/fast temporal sampling), while X3D-L and R(2+1)D consume a single spatiotemporal tensor. Importantly, the pre-processing budget is treated as part of the real-time constraint, hence only linear-time operations (in number of pixels/frames) are used, and the implementation is engineered to avoid unnecessary copies through contiguous clip snapshotting and per-camera buffer management.

From a timing standpoint, the end-to-end processing chain can be decomposed into acquisition/decoding (t_{acq}), buffer update and snapshot (t_{buf}), pre-processing

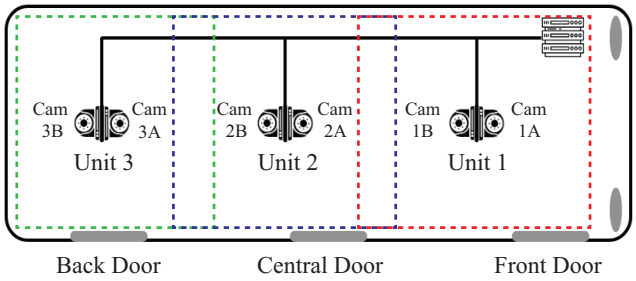


Figure 5.9: *Acquisition unit placement: anterior (red), central (blue), and posterior (green) bus zones.*

(t_{prep}), and GPU inference (t_{inf}). While acquisition runs continuously in parallel, the remaining stages must satisfy a bounded-cycle requirement associated with the update frequency:

$$t_{\text{buf}} + t_{\text{prep}} + t_{\text{inf}} \leq T_{\text{cycle}}, \qquad T_{\text{cycle}} \approx 1 \text{ s}. \tag{5.2}$$

Latency measurements performed in the bus simulator confirm that the proposed configuration is compatible with real-time operation, reporting an end-to-end processing time in the range 620–950 ms (including acquisition, buffering, pre-processing and model prediction). Beyond confirming feasibility, these measurements motivate a design principle adopted throughout the stack: each stage must degrade gracefully under transient stream perturbations (RTSP jitter, partial disconnections) by prioritizing recency of frames and bounded queues, so that the system remains reactive rather than accumulating stale data.

Finally, the inference output is stored in a structured per-camera buffer (scores, predicted labels, timestamps) and connected to the alert subsystem. In the present chapter, we intentionally describe the decision layer in architectural terms—i.e., as a configurable post-processing stage that can implement confidence thresholding and short temporal persistence—while the *specific* parameter tuning is deferred to the next chapter, where it is derived from experimental evidence and deployment trade-offs. Upon triggering, the system retains the corresponding evidence segment (at least the current 3 s buffer, optionally extended with pre/post context) and dispatches a compact GSM notification containing event metadata and a reference to the stored clip. This preserves a low-latency response loop while keeping raw video local by default, coherently with the edge-analytics rationale [56, 36].

5.3 Field Validation

After validating offline performance and real-time feasibility in the simulator, the complete pipeline was deployed on a 13-m urban bus and evaluated through supervised field trials conducted in collaboration with a local public transport operator.

The objective of this campaign was not merely to confirm classification accuracy, but to stress-test the entire closed-loop system under operational conditions that are difficult or impossible to reproduce in laboratory settings. These include continuous vehicle vibrations, abrupt illumination changes, non-stationary backgrounds due to passenger flow, strong viewpoint and distance variability, and frequent partial occlusions caused by standing passengers and onboard structures.

These factors are well known to challenge deployable violence recognition systems, particularly in surveillance-like scenarios where domain shift and far-field observations progressively attenuate the spatiotemporal cues exploited by deep action recognition models. Importantly, the field validation was designed to verify whether the qualitative behaviors observed during training and offline testing—such as confidence degradation with distance, occlusion-induced errors, and hard-negative confusion—also emerge in real operation, thereby assessing the realism and predictive value of the offline analysis.

5.3.1 Trial Protocol and Ground-Truth Definition

To explicitly capture the effect of camera-to-subject distance, the bus interior was partitioned into three contiguous zones of approximately 4 m each—anterior, central, and posterior (Figure 5.9). Each zone was monitored by three partially overlapping viewpoints originating from distinct acquisition units. This configuration reflects the actual deployment architecture and intentionally introduces redundancy, allowing the same interaction to be observed at different distances and angles.

During the campaign, trained actors performed scripted but realistic behaviors representative of in-vehicle conflicts, including pushing, grabbing, confrontational gestures, and escalation scenarios embedded within normal passenger activity. A total of 53 experiments were conducted across all zones.

Ground truth was defined at two complementary levels. At the clip level, each 1 Hz decision window was labeled as Fight or No-Fight. When applicable, event-level annotations were also provided by marking the start and end of each aggressive interaction. This dual annotation scheme enables the computation of both window-level metrics and event-level indicators, which are particularly relevant in real-time safety systems where detecting the presence of an event at least once during its evolution is often more important than continuous frame-by-frame correctness.

5.3.2 Overall Performance and Operational Metrics

Across all trials, the system achieved an overall accuracy of 86.4%, with Fight precision of 0.84 and Fight recall of 0.81, confirming that the selected backbone and the real-time pipeline retain discriminative capability under realistic bus conditions. However, clip-level metrics alone do not fully characterize operational suitability. Therefore, in addition to conventional classification scores, we report two

Descriptor	Value
Number of trials	53
Zones covered	Anterior / Central / Posterior
Views per zone	3 (overlapping)
Nominal frame rate	30 fps
Decision rate	1 Hz (stride 1 s, window 3 s)
Event duration (median / IQR)	– / –

Table 5.4: *Field-trial protocol summary. Report event duration statistics if event-level ground truth is available.*

deployment-oriented indicators: *event detection rate* (EDR) and *time-to-detection* (TTD). EDR measures the fraction of aggressive events for which at least one Fight decision is produced within the annotated event interval; TTD measures the delay between the annotated start of aggression and the first correct Fight decision. These indicators quantify not only whether the system is correct, but also whether it reacts *in time* to support mitigation and alerting. Let $\mathcal{E}$ denote the set of annotated aggressive events, and let $[t_s(e), t_e(e)]$ be the start/end time of event e. Given the 1 Hz decision stream $\hat{y}(t) \in \{0,1\}$ for Fight/No-Fight, the event detection rate (EDR) is defined as

$$\mathrm{EDR} = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \mathbb{I}\Big(\exists t \in [t_s(e), t_e(e)] : \hat{y}(t) = 1\Big). \tag{5.3}$$

The time-to-detection (TTD) for an event e is

$$\mathrm{TTD}(e) = \min\{t - t_s(e) \mid t \in [t_s(e), t_e(e)],\ \hat{y}(t) = 1\}, \tag{5.4}$$

and we report its distribution (median and high-percentiles) across detected events. Empirically, most aggressive events are detected within the first few seconds of their evolution when at least one camera provides a near- or mid-field view. Conversely, events observed only from far-field viewpoints exhibit delayed detection or occasional misses, a behavior that directly mirrors the confidence degradation patterns observed during offline testing.

5.3.3 Performance by Zone and Camera Viewpoint

A stratified analysis by zone and camera viewpoint reveals a clear dependence on working distance. As summarized in Table 5.5, the central zone yields the highest performance across all metrics. This behavior is explained by two complementary factors: (i) mid-range distances that preserve full-body dynamics while maintaining sufficient spatial resolution, and (ii) the presence of at least one closer viewpoint capable of capturing fine-grained limb motion.

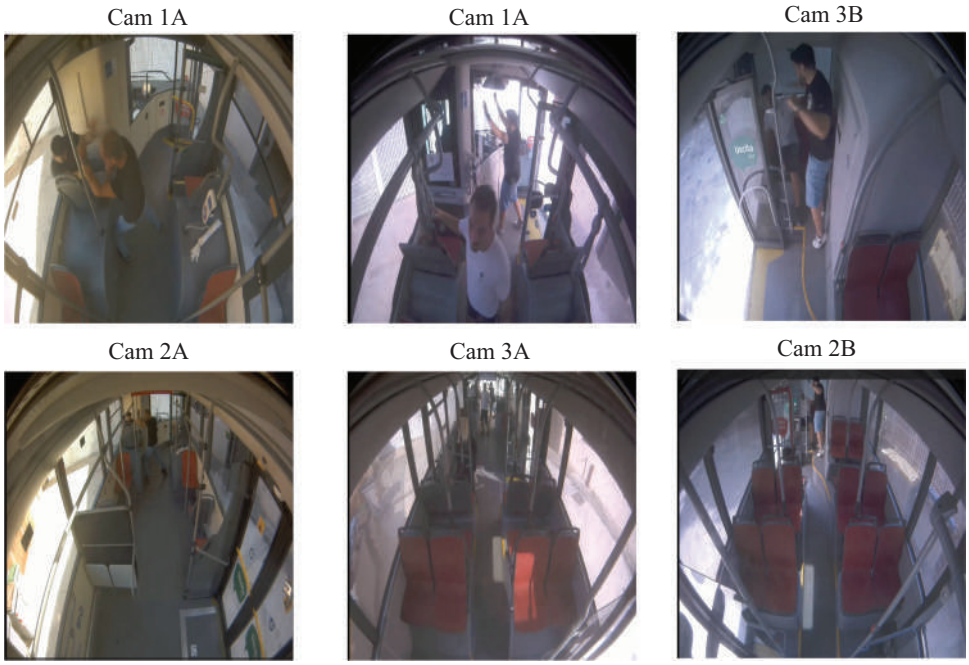


Figure 5.10: *Comparison of three violent events from different viewpoints.*

In the anterior and posterior zones, performance varies more strongly across cameras. In both cases, the camera closest to the interaction consistently provides the most reliable predictions, while farthest viewpoints exhibit reduced confidence and increased misclassification risk. This pattern is fully consistent with far-field degradation effects: as distance increases, actors occupy fewer pixels, limb motion becomes less distinguishable, and occlusion probability increases due to standing passengers and seat structures.

Figure 5.10 visually illustrates this phenomenon by showing the same violent interaction observed from different viewpoints. While near-field views yield confident Fight predictions, far-field views often approach the decision boundary or cross into No-Fight, despite the interaction being identical.

5.3.4 Distance Sensitivity and Confidence Degradation

Distance sensitivity emerges as one of the most systematic factors affecting field performance. Table 5.6 reports the confidence scores assigned to the same violent scene observed from three different distances. The closest camera (≈ 3 m) produces a confident and correct Fight prediction, while intermediate distances show reduced margins, and the farthest view (≈ 9 m) leads to a low-confidence No-Fight decision.

This observation provides empirical confirmation—under real operating conditions—of the trends identified during offline training and testing. In the present setup, an effective working range of approximately 3–4 m can be identified, beyond

Zone	Camera	Accuracy	Precision (Fight)	Recall (Fight)
Anterior	Unit 1	0.87	0.84	0.81
	Unit 2	0.85	0.82	0.79
	Unit 3	0.80	0.77	0.75
Central	Unit 1	0.89	0.86	0.84
	Unit 2	**0.91**	**0.88**	**0.86**
	Unit 3	0.90	0.87	0.84
Posterior	Unit 1	0.82	0.79	0.76
	Unit 2	0.84	0.81	0.78
	Unit 3	0.86	0.83	0.80
Overall (avg)	—	**0.86**	**0.83**	**0.80**

Table 5.5: *Detection performance per bus zone and acquisition camera, averaged over 53 field experiments.*

Camera ID	Distance	Predicted Label	Confidence Score
Unit 1	∼3 m	**Fight**	**0.91**
Unit 2	∼6 m	Fight	0.69
Unit 3	∼9 m	**No Fight**	**0.56**

Table 5.6: *Prediction confidence of the same violent scene observed from three different distances.*

which confidence progressively degrades and false negatives become more likely. This finding has direct architectural implications: it motivates confidence-aware decision policies, temporal persistence, and the exploitation of overlapping views to ensure that at least one camera operates within the effective range whenever possible.

5.3.5 Qualitative Field Error Analysis (TP/FP/TN/FN) and Failure Taxonomy

Quantitative metrics are complemented by a qualitative inspection of representative True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) examples extracted from on-board recordings (Figure 5.11). Importantly, the failure modes observed on the bus closely mirror those identified during offline analysis, indicating that the training-time diagnostics are predictive of real-world behavior.

A consistent taxonomy of errors emerges:

True Positives predominantly correspond to interactions occurring within the effective working range, where physical contact and antagonistic intent are clearly

visible and generate strong spatiotemporal evidence.

False Negatives are strongly associated with increased distance (typically beyond $\approx$ 3 m) and with partial or severe occlusions. In these cases, critical limb motion is attenuated or intermittently hidden by other passengers or structural elements, leading the model to underestimate aggression.

False Positives are primarily triggered by hard negatives involving arm elevation and upper-body motion, such as passengers holding overhead supports, putting on or removing jackets, or performing slow but wide arm movements. These actions are particularly problematic when observed at close range, where motion energy is high but antagonistic intent is absent.

Additional FP cases arise under strong crowding, where severe occlusions and compressive motion patterns resemble aggressive interactions at the spatiotemporal level.

These observations confirm that the dominant error drivers are not random, but are systematically linked to distance, occlusion, and semantic ambiguity—exactly the factors anticipated during offline training and validation.

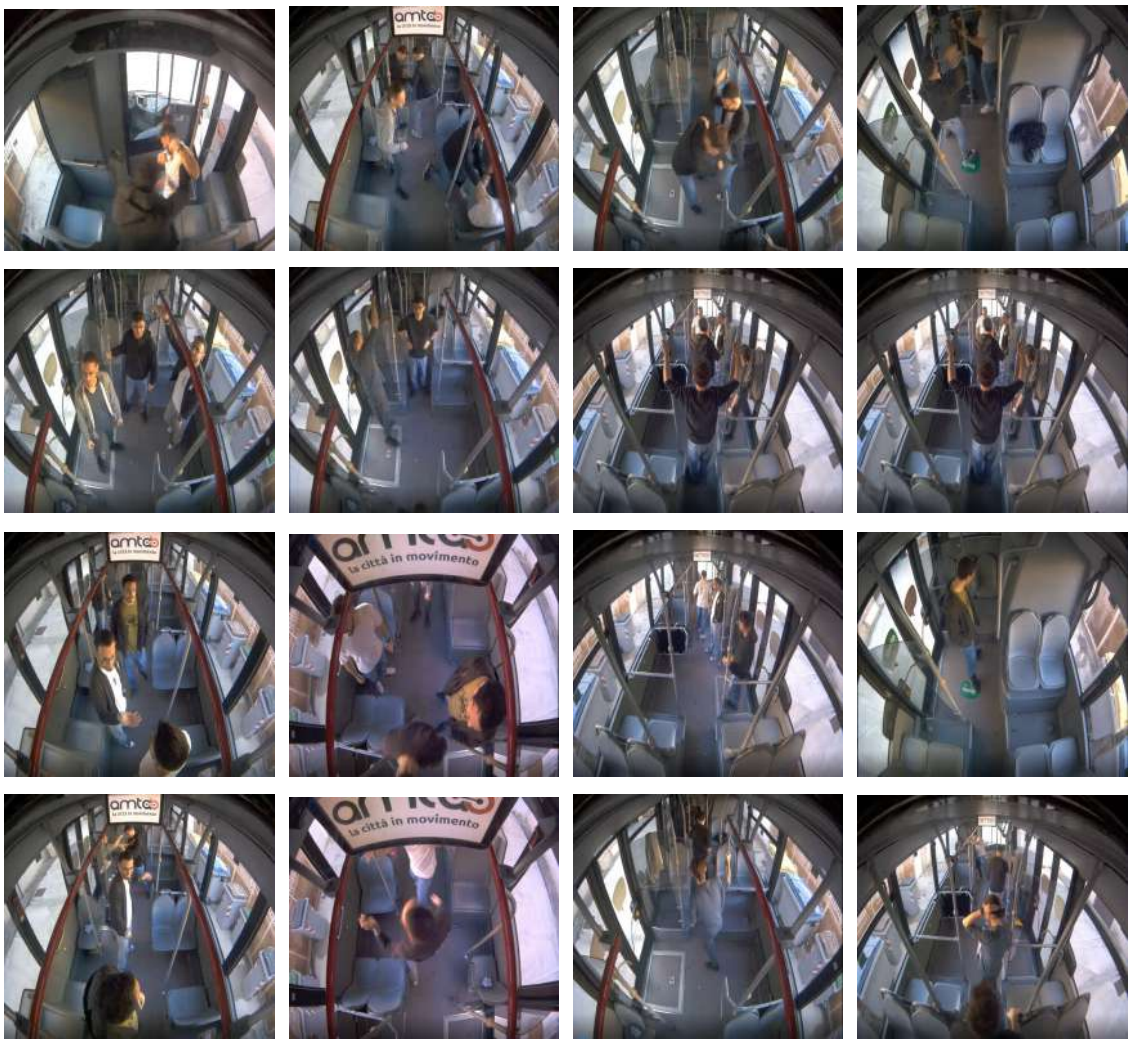


Figure 5.11: *Field qualitative analysis: representative TP/FP/TN/FN examples extracted from on-board trials. Each panel should report ground truth, predicted label, Fight confidence, camera ID, and approximate distance. A recommended selection strategy is to include at least one near-field and one far-field example per category to explicitly highlight distance and occlusion effects.*

A consistent pattern emerges from these examples. True positives are typically characterized by sufficiently close working distances (within the effective range) and by visible physical contact or high-acceleration motion primitives (pushes, grabs) that generate strong spatiotemporal evidence. False negatives frequently occur in far-field views where actors occupy few pixels, or when the critical limb motion is partially hidden by standing passengers or seat structures, reducing the observable motion energy and leading the model to favor No-Fight. False positives concentrate on hard negatives with abrupt, high-energy motion (e.g., fast arm gestures, crowd

compression during boarding, or playful contact) where the absence of antagonistic intent is semantically clear to humans but not fully encoded in the clip-level visual evidence. Finally, true negatives often correspond to static or slowly varying scenes where the system maintains low Fight confidence, confirming that the pipeline does not trivially over-trigger in normal riding conditions.

5.3.6 Temporal Behavior, Detection Delay, and System Reliability in Field Operation

Beyond recognition accuracy, a deployable safety system must exhibit stable temporal behavior and bounded latency under real operating conditions. Because the pipeline outputs one decision per second using a 3 s sliding window, each camera generates a temporal trace of Fight confidence scores that can be aligned to annotated event onsets.

Near-field views typically show an early and steep rise in confidence shortly after event onset, resulting in short time-to-detection. Far-field views, by contrast, often exhibit delayed or oscillatory confidence near the decision threshold, reinforcing the need for temporal persistence mechanisms in the decision logic.

System-level reliability was also monitored throughout the trials. Despite RTSP jitter, vehicle vibrations, and transient reconnections, the end-to-end processing time remained bounded and compatible with real-time constraints, with no accumulation of stale frames or unbounded delays. These observations confirm that the system not only recognizes violence with acceptable accuracy, but also operates robustly as a real-time embedded pipeline.

5.3.7 Limitations and Threats to Validity

While the field trials provide strong evidence of feasibility and robustness, several limitations must be acknowledged. The experiments are supervised and scripted for safety and ethical reasons; although behaviors are realistic, they may not fully capture the unpredictability of spontaneous real-world aggression. The number of trials is sufficient to reveal systematic trends—such as distance sensitivity and occlusion effects—but may not exhaustively cover rare corner cases. Furthermore, camera placement and vehicle geometry are fixed in this study; different bus models or mounting configurations may alter the effective working range and should be revalidated.

Finally, inference is performed independently per camera without explicit multi-view fusion. While overlapping views already increase the probability that at least one camera observes the interaction at an effective distance, formal multi-view consolidation remains a promising direction to further reduce false negatives under far-field and heavily occluded conditions.

Chapter 6

Conclusions and Future Work

Violence detection in public transportation remains markedly underexplored compared to general video surveillance and action recognition in curated benchmarks. Existing studies often report promising accuracy under controlled conditions, yet only a limited subset addresses the constraints and failure modes that arise in real deployments, such as viewpoint discontinuities, strong distance variation, occlusions caused by crowding, unstable illumination, and strict requirements on latency and privacy-preserving operation. This thesis contributes to narrowing this gap by presenting and validating a complete edge-based, multi-camera pipeline for real-time violent action recognition inside buses, designed explicitly around deployment constraints rather than benchmark-only performance.

A first contribution is the system-level architecture: a distributed acquisition layout with six cameras arranged to cover the bus cabin through partially overlapping views, connected to an onboard embedded GPU server executing the full inference pipeline locally. This design follows a "local-first" principle: raw video remains on the vehicle and network connectivity is used only for optional transmission of compact alerts and metadata. The resulting architecture enables bounded-latency operation and reduces privacy and bandwidth exposure, while ensuring that detection remains functional even under intermittent connectivity.

A second contribution is the training and evaluation methodology tailored to the bus domain. Rather than relying solely on public datasets, the work adopts a composite data strategy that combines large-scale public benchmarks with proprietary bus recordings and laboratory/simulator data that reproduce the sensor geometry and operational conditions of the target deployment. This choice directly addresses the domain-shift effects demonstrated in the experimental chapters: models trained only on public data can achieve strong benchmark results yet degrade substantially when transferred to bus interiors. By integrating in-domain samples into the training process, the system improves both final performance and the stability of learning dynamics, reducing overfitting to dataset-specific artifacts and promoting features that remain discriminative under realistic bus conditions.

A third contribution concerns model adaptation and selection under imbalance and deployment constraints. The experimental analysis systematically compares multiple state-of-the-art backbones under consistent pre-processing and real-time constraints, emphasizing deployment-relevant metrics (minority-class behavior, PR-oriented reading, confusion matrices, qualitative errors) rather than accuracy alone. The adopted fine-tuning protocol—based on pretrained weights and a progressive unfreezing schedule—improves convergence speed and minority-class learning. This is reflected both in the learning curves and in the downstream error structure: pretraining consistently yields earlier improvements in Fight-class F1/recall and more stable optimization trajectories, while training from scratch shows noisier dynamics and delayed learning of the minority class.

The field validation provides the most important evidence: the proposed pipeline maintains real-time operation onboard and preserves discriminative capability under operational perturbations. In supervised trials conducted on a 13 m urban bus, the system achieved approximately 86% overall accuracy, with 0.83 precision and 0.80 recall on the Fight class, confirming that the offline improvements transfer to deployment. Crucially, qualitative analysis on both offline test data and field recordings converges on the same dominant failure modes, allowing the thesis to move beyond aggregate numbers toward a deployment-oriented understanding of limitations. Across backbones, confidence degrades with camera-to-subject distance, and false negatives become more frequent beyond an effective range of roughly 3–4 m. Severe occlusions—either from crowding or large onboard structures—also induce false negatives by suppressing observable limb dynamics. Conversely, false positives are often triggered by hard negatives involving arm elevation and upper-body motion (e.g., holding overhead supports, putting on/removing a jacket), especially when observed at close range where motion energy is high. The fact that these behaviors emerge consistently during training analysis and reappear in the field supports the validity of the experimental methodology and provides clear engineering guidance for subsequent system refinements.

Overall, the thesis demonstrates that robust violence recognition on buses is not achieved by backbone choice alone, but by the co-design of data, training protocol, architecture, and evaluation around real operational constraints. Among the evaluated models, X3D offers the best compromise between accuracy, stability, and embedded feasibility, making it the most appropriate backbone for deployment in the current system configuration.

Several research and engineering directions follow naturally from these results.

Dataset expansion and coverage of operational regimes. Increasing the diversity of in-domain data—different buses, camera mounting variations, lighting conditions, passenger densities, and additional interaction patterns—remains the most direct path to improved robustness. In particular, targeted collection of far-field interactions and heavy-occlusion scenes would address the two dominant sources of false negatives identified in both offline and field analysis.

Distance- and occlusion-aware inference. The systematic confidence degradation beyond 3–4 m motivates explicit mechanisms to incorporate geometric context into decision making. Promising approaches include multi-view consolidation (late fusion across cameras), confidence calibration conditioned on estimated distance, and occlusion-aware filtering using auxiliary signals (e.g., crowd density estimates). These strategies aim to reduce missed detections when at least one view provides sufficient visual evidence, while avoiding excessive false alarms.

Hard-negative modeling and context disambiguation. False positives triggered by arm-raising actions suggest that clip-level motion patterns alone are sometimes insufficient to separate benign and aggressive interactions. Future work may incorporate lightweight context cues (pose dynamics, interaction proximity, temporal persistence logic, or multi-clip confirmation) to discriminate "high-motion non-violence" from actual fights, reducing alert fatigue without sacrificing recall.

Temporal decision policies aligned with safety requirements. Because deployment relevance depends on timely detection, future iterations should further develop persistence-based logic and event-level evaluation. Rather than relying on a single-window decision, the system can enforce temporal consistency (e.g., K-of-N voting or hysteresis thresholds) to stabilize borderline confidence traces observed in far-field views and reduce flickering predictions.

Broader event taxonomy for smart mobility. Finally, the same edge analytics platform can be extended beyond violence to additional safety-critical events in public transport, such as theft attempts, harassment, severe intoxication, medical emergencies, falls, or crowding anomalies. This would shift the system from a single-task detector toward a more comprehensive onboard safety monitoring framework, with shared perception modules and task-specific heads

Bibliography

[1] Toluwani Aremu. Smart city cctv violence detection dataset (scvd), 2023. Kaggle dataset. Accessed 2025–12–06.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. In *Proc. ICCV*, pages 6836—-6846, 2021.

[3] Muhammad Asad, Jiaolong Yang, Jing He, P. Shamsolmoali, and Xiangjian He. Multi–frame feature–fusion–based model for violence detection. *The Visual Computer*, 36(10):2113—-2127, 2020.

[4] Enrique Bermejo Nievas, Oscar Deniz Suarez, Gloria Bueno García, and Rahul Sukthankar. Violence detection in video using computer vision techniques. In *Computer Analysis of Images and Patterns: 14th International Conference, CAIP 2011, Seville, Spain, August 29–31, 2011, Proceedings, Part II 14*, pages 332—-339. Springer, 2011.

[5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space–time attention all you need for video understanding? In *Proc. ICML*, pages 813—-824, 2021.

[6] Ankit Bhardwaj, Prakhar Jain, and Nitin Jain. Intelligent video surveillance: A review through deep learning techniques for crowd analysis. *Journal of Big Data*, 6(1):1—-27, 2019.

[7] Massimo Caputo, Marco De Nadai, et al. Bus violence: An open benchmark for video violence detection on public transport. *Sensors*, 22(21):8345, 2022.

[8] Massimo Caputo, Marco De Nadai, et al. Bus violence: An open benchmark for video violence detection on public transport. *Pattern Recognition Letters*, 2022. Europe PMC preprint.

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299—-6308, 2017.

[10] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, pages 4724—-4733, 2017.

[11] Vania Ceccato and Andrew Newton. Safety and security in transit environments: An interdisciplinary approach. *Crime Prevention and Security Management*, pages 121–122, 2015.

[12] Chen Chen, Yu Qiao, and Jun Xiao. Learning surveillance video representations via temporal regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[13] Ming Cheng, Yu Cai, Xiaolu Li, Bing Fan, and Xianglong Liu. Rwf–2000: An open large scale video database for violence detection, 2019.

[14] Ronald V. Clarke. *Situational Crime Prevention: Successful Case Studies*. Harrow and Heston, 1997.

[15] Lawrence E. Cohen and Marcus Felson. Social change and crime rate trends: A routine activity approach. *American Sociological Review*, 44(4):588—-608, 1979.

[16] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, pages 886—-893, 2005.

[17] C.-H. Demarty, B. Ionescu, Y.-G. Jiang, and C. Penet. Benchmarking violent scenes detection in movies. In *Proc. CBMI*, pages 1–6, 2014.

[18] C.-H. Demarty, C. Penet, M. Soleymani, and G. Gravier. Vsd, a public dataset for the detection of violent scenes in movies: Design, annotation, analysis and evaluation. *Multimedia Tools and Applications*, 74(15):5243–5277, 2014.

[19] Oscar Deniz, Iván Serrano, Gloria Bueno, and Tae-Kyun Kim. Fast violence detection in video. In *Proceedings of the 2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 478—-485. IEEE, 2014.

[20] F. M. Donald, P. Bennett, A. Bennett, J. MacDonald, C. Horwath, and K. Cranwell. Task disengagement and implications for vigilance performance in cctv surveillance. *Cognition, Technology & Work*, 17(2):241—-251, 2015.

[21] European Data Protection Board. Guidelines 3/2019 on processing of personal data through video devices, 2019.

[22] European Union. General data protection regulation (gdpr), 2016.

[23] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proc. CVPR*, pages 200—-210, 2020.

[24] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proc. ICCV*, pages 6202—-6211, 2019.

[25] Yanyun Gao, Hong Liu, Xiaoyan Sun, Chao Wang, and Yong Liu. Violence detection using oriented violent flows. *Image and Vision Computing*, 48:37—-41, 2016.

[26] Theodoros Giannakopoulos, Aggelos Pikrakis, Sergios Theodoridis, and George Katsaggelos. Audio–based event detection in movies using hierarchical audio segmentation. *Artificial Intelligence Tools*, 2010.

[27] Martin Gill and Angela Spriggs. *Assessing the Impact of CCTV*. Home Office Research Study, 2005.

[28] Ping Gong and Xudong Luo. A survey of video action recognition based on deep learning. *Knowledge–Based Systems*, 320:113594, June 2025.

[29] Chunhui Gu, Chen Sun, David Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Shanmuganathan Ricco, Rahul Sukthankar, and Cordelia Schmid. Ava: A video dataset of spatio–temporally localized atomic visual actions. In *Proc. CVPR*, pages 6047–6056, 2018.

[30] Tal Hassner, Yehuda Itcher, and Orit Kliper-Gross. Violent flows: Real–time detection of violent crowd behavior. In *Proc. CVPR Workshops*, pages 1–6, 2012.

[31] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263—-1284, 2009.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[33] Adrián Hernández, Gabriel Ruiz, Antonino Marvuglia, et al. Revisiting vision–based violence detection in videos: A critical perspective. *Neurocomputing*, page 128123, 2024.

[34] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[35] Herwin Alayn Huillcen Baca, Flor de Luz Palomino Valdivia, and Juan Carlos Gutierrez Caceres. Efficient human violence recognition for surveillance in real time. *Sensors*, 24(2):668, 2024.

[36] Claudia Istrate, Ivan Costea, and Grigore Stamatescu. Smart video surveillance system based on edge computing. *Sensors*, 21(9):2958, 2021.

[37] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer–arithmetic–only inference. In *Proc. CVPR*, pages 2704–2713, 2018.

[38] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221—-231, 2013.

[39] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large–scale video classification with convolutional neural networks. In *CVPR*, 2014.

[40] Will Kay, João Carreira, Karen Simonyan, et al. The kinetics human action video dataset.

[41] Harjinder Keval and M. Angela Sasse. Eye tracking in cctv control rooms. *Ergonomics*, 53(3):324—-338, 2010.

[42] Ivan Laptev. On space–time interest points. In *International Journal of Computer Vision*, volume 64, pages 107—-123, 2005.

[43] Chao Li, Yonghong Chen, Yi Yang, and Meng Wang. Skeleton-based violence detection using spatio-temporal graph convolutional networks. *Pattern Recognition Letters*, 128:461–466, 2019.

[44] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. Multisports: A multi–person video dataset of spatio–temporally localized sports actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13536—-13545. IEEE/CVF, 2021.

[45] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[46] N. H. Mackworth. The breakdown of vigilance during prolonged visual search. *Quarterly Journal of Experimental Psychology*, 1(1):6—-21, 1948.

[47] Vincenzo Mancuso, Mauro Scolari, Daniele Raho, and Gianluca Rossini. Edge hpc architectures for ai–based video surveillance applications. *Electronics*, 13(9):1757, 2024.

[48] Yuwei Miao and Wenyi Luo. Improve generalization ability of cnn by data augmentation and se block in landmark classification. In *Proceedings of the 2022 IEEE 14th International Conference on Computer Research and Development (ICCRD)*, pages 250–255. IEEE, 2022.

[49] Pablo Negre, Ricardo S. Alonso, Alfonso González-Briones, Javier Prieto, and Sara Rodríguez-González. Literature review of deep-learning-based detection of violence in video. *Sensors*, 24(12):4016, 2024.

[50] Andrew D. Newton. Crime on public transport. In R. Bruksma and D. Weisburd, editors, *Encyclopedia of Criminology and Criminal Justice*, pages 709—-720. Springer, 2014.

[51] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4694—-4702, 2015.

[52] Victor M. Nievas, Cristina Suarez, Gonzalo García, and Rahul Sukthankar. Violence detection in video using computer vision techniques. In *Proc. CAIP*, pages 332–339. Springer, 2011.

[53] Fatih Porikli, Alberto Del Bimbo, and Richa Singh. Intelligent surveillance systems. In *Handbook of Visual Computing*. Springer, 2018.

[54] Fatih Porikli, Gian Luca Foresti, João Paulo Aredes, Andrea Cavallaro, Carlo S. Regazzoni, and Sergio A. Velastin. Video surveillance: Past, present, and now the future. *IEEE Signal Processing Magazine*, 30(3):190—-198, 2013.

[55] S. Rankin, N. Cohen, K. Maclennan-Brown, and K. Sage. Cctv operator performance benchmarking. In *2012 IEEE International Carnahan Conference on Security Technology (ICCST)*, page 325–330. IEEE, October 2012.

[56] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 2017.

[57] I. Serrano, O. Deniz, J. L. Espinosa-Aranda, and G. Bueno. Fight recognition in video using hough forests and 2d convolutional neural network. *IEEE Transactions on Image Processing*, 27(10):4787—-4797, 2018.

[58] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12026–12035. IEEE, 2019.

[59] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304. IEEE, 2011.

[60] Karen Simonyan and Andrew Zisserman. Two–stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NIPS)*, volume 27, pages 568—-576, 2014.

[61] M. Sjöberg, B. Ionescu, Y.-G. Jiang, V.-L. Quang, M. Schedl, and C.-H. Demarty. The mediaeval 2014/2015 affect task: Violent scenes detection (vsd), 2015.

[62] Wei Song, Dong Zhang, Xu Zhao, Jianhua Yu, Ronggang Zheng, and An Wang. A novel violent video detection scheme based on modified 3d convolutional neural networks. *IEEE Access*, 7:39172—-39179, 2019.

[63] Andrew Stainer, Mark Gill, and A. Porter. The paradox of surveillance: Understanding perceptual load. *Security Journal*, 2017.

[64] Chris Stauffer and W. Eric L. Grimson. Adaptive background mixture models for real–time tracking. In *Proc. CVPR*, pages 246—-252, 1999.

[65] Waqas Sultani, Chen Chen, and Mubarak Shah. Real–world anomaly detection in surveillance videos. *Proc. CVPR*, pages 6479–6488, 2018.

[66] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data–efficient learners for self–supervised video pre–training. In *Proc. NeurIPS*, 2022.

[67] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1529–1541, 2011.

[68] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4489—-4497, 2015.

[69] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450—-6459, 2018.

[70] Transport for London. Camera monitoring system (cms): Specification. Technical report, Transport for London, 2022. Accessed 2025–11–15.

[71] Transport for London. Cctv and surveillance cameras, 2024. Accessed 2025–11–15.

[72] Farhan Ullah, Mohammad S. Obaidat, Adil Ullah, Mohannad Hijji, Khan Muhammad, and Sung W. Baik. A comprehensive review on vision–based violence detection in surveillance videos. *ACM Computing Surveys (CSUR)*, 55(5):1–38, 2022.

[73] Christophe Vanroelen et al. Review of measures to prevent and manage aggression against transport workers. *Safety Science*, 165:106225, 2023.

[74] Michalis Vrigkas, Christophoros Nikou, and Ioannis A. Kakadiaris. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2:28, 2015.

[75] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proc. ICCV*, pages 3551—-3558, 2013.

[76] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *Proc. ECCV*, pages 20—-36, 2016.

[77] Peng Wang, Fanwei Zeng, and Yuntao Qian. A survey on deep learning–based spatio–temporal action detection. *arXiv preprint arXiv:2308.01618*, 2023.

[78] Xiaogang Wang. Intelligent multi–camera video surveillance: A review. *Pattern Recognition Letters*, 34(1):3—-19, 2012.

[79] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non–local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7794—-7803, 2018.

[80] Brandon C. Welsh and David P. Farrington. Crime prevention effects of cctv. *British Journal of Criminology*, 42(3):541—-557, 2002.

[81] Brandon C. Welsh and David P. Farrington. *Effects of Closed–Circuit Television Surveillance on Crime*. Home Office Research, 2008.

[82] Chuang Wu, Long Liu, Yue Wang, et al. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *Pattern Recognition and Computer Vision (PRCV 2020), Lecture Notes in Computer Science*, pages 347–359. Springer, 2020.

[83] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[84] George Yannis, Eleni Papadimitriou, and Christina Antoniou. Bus driver safety and assaults: A european overview. *Transportation Research Procedia*, 3:243—-252, 2014.

[85] Wei Zhang, Ling Chen, and Hang Xu. Edge intelligence: On–device machine learning for video analytics. *IEEE Internet of Things Journal*, 2021.

[86] Marie Ève Lavigne and Martin B. Doucet. Crime and public transportation. *Transportation Research Record*, 2011.