RESEARCH ARTICLE

Detection and Recognition of Real-Time Violence and Human Actions Recognition in Surveillance using Lightweight MobileNet Model

Altaf Hussain[1,*]

[1] School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Abstract

Real-time detection of violent behavior through surveillance technologies is increasingly important for public safety. This study tackles the challenge of automatically distinguishing violent from non-violent activities in continuous video streams. Traditional surveillance depends on human monitoring, which is time-consuming and error-prone, highlighting the need for intelligent systems that detect abnormal behaviors accurately with low computational cost. A key difficulty lies in the ambiguity of defining violent actions and the reliance on large annotated datasets, which are costly to produce. Many existing approaches also demand high computational resources, limiting real-time deployment on resource-constrained devices. To overcome these issues, the present work employs the lightweight MobileNet deep learning architecture for violence detection in surveillance videos. MobileNet is well-suited for embedded devices such as Raspberry Pi and Jetson Nano while maintaining competitive accuracy. In Python-based simulations on the Hockey Fight dataset, MobileNet is compared with AlexNet, VGG-16, and GoogleNet. Results show that MobileNet achieved 96.66% accuracy with a loss of 0.1329, outperforming the other models in both accuracy and efficiency. These findings demonstrate MobileNet's superior balance of precision, computational cost, and real-time feasibility, offering a robust framework for intelligent surveillance in public safety monitoring, crowd management, and anomaly detection.

Keywords: real-time violence detection, CCTV surveillance video, convolutional neural networks, VGG-16, GoogLeNet, AlexNet, MobileNet.

1 Introduction

Urban safety concerns have intensified in recent years as incidents of crime and terrorism in public spaces have grown more visible and widely reported. This reality has accelerated the need for intelligent video surveillance capable of automatically identifying violent behaviors without constant human oversight. Consequently, activity understanding from video—especially violence activity recognition—has

Submitted: 31 August 2025
Accepted: 18 September 2025
Published: 21 September 2025

Vol. 1, **No.** 3, 2025.
10.62762/JIAP.2025.839123

***Corresponding author:**
✉ Altaf Hussain
altafkfm74@gmail.com

Citation
Hussain, A. (2025). Detection and Recognition of Real-Time Violence and Human Actions Recognition in Surveillance using Lightweight MobileNet Model. *ICCK Journal of Image Analysis and Processing*, 1(3), 125–146.

© 2025 by the Author. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

become a prominent topic across both academic research and industrial deployments. Modern action detection and recognition support a wide range of downstream systems, including security surveillance, human–machine interaction, autonomous navigation, and numerous industrial applications. A commonly held assumption in this domain is that violence is rare in otherwise normal footage, which naturally motivates anomaly-detection formulations of the task [27, 28]. Yet, detecting violent activity remains difficult: it inherits the well-known challenges of anomaly detection and adds the computational burdens associated with parsing high-dimensional, continuous video streams. Human Action Recognition (HAR) in the wild is complicated by many visual and contextual factors. Variability in human body proportions and poses, changes in object appearance and background clutter, diverse illumination conditions, occlusions among people or scene elements, and rapid viewpoint shifts collectively degrade recognition reliability [22, 23]. In practical surveillance footage, additional nuisances—camera motion, motion blur, scale changes, crowd density, and partial visibility—further elevate the complexity. Within this landscape, our study points to three core obstacles that any violence recognition system must address. First, "violent objects" or "violent primitives" cannot be exhaustively enumerated or hand-crafted; the semantic boundaries of what constitutes violence are fuzzy and context dependent. Systems must therefore reason under uncertainty and tolerate ambiguous or weak labels. Second, large, carefully annotated video corpora are scarce because frame- or segment-level labeling is time-consuming, expensive, and requires domain expertise [31]. Third, many operational surveillance models still rely heavily on handcrafted visual features, which demand substantial problem-specific knowledge and manual tuning; such engineered representations often generalize poorly and impose considerable development overhead [3].

Within computer vision and machine learning, HAR is an essential research area whose primary goal is to automatically recognize and categorize the actions depicted in video sequences [21]. The problem is intrinsically challenging because it requires modeling spatial appearance and temporal dynamics simultaneously. Typical obstacles include occlusion, intra-class variation in human shape or attire, cluttered or dynamic backgrounds, and strong viewpoint diversity; the severity of these issues often depends on the granularity and duration of the activity under study. In practice, activities are commonly grouped into four broad categories—gestures, actions, interactions, and group activities—organized roughly by increasing temporal extent and structural complexity [24, 25]. Violence recognition intersects these categories: it may appear as brief gestures (e.g., sudden strikes), longer actions (e.g., prolonged aggression), dyadic interactions, or group-level events that evolve over time. Approaches to HAR can be organized by sensing modality and system design. From a data-collection and methodological standpoint, three families are typically recognized: visual sensor–based, non-visual (scalar) sensor–based, and multimodal methods that combine both [19, 20]. The key difference is the nature of the observed signal: visual sensors deliver 2D frames or 3D video streams, while non-visual sensors (e.g., accelerometers, gyroscopes, magnetometers, microphones) produce one-dimensional time series with different noise and sampling characteristics [16]. In the last few years, widespread adoption of wearable devices—smartphones, fitness bands, and smartwatches—has broadened the availability of non-visual signals in everyday settings [17]. These devices now offer on-board communication and sufficient compute to support on-device HAR, enabling applications in daily healthcare monitoring, rehabilitation training, and disease prevention [33]. In parallel, visual sensor–based techniques remain highly influential in computer vision and deep learning communities, underpinning applications in human–computer interaction, general video surveillance, ambient assisted living, human–robot collaboration, gaming, and content-based image/video retrieval [1].

Prior work on anomalous event detection from video—the closest related area—has proposed diverse strategies and demonstrated progress; nevertheless, reliably flagging violent behaviors in crowded public scenes remains a difficult, active research problem [5, 13]. In dense crowds, targets are frequently occluded, camera viewpoints are often suboptimal, and subtle pre-incident cues may be brief or only partially visible. This study is particularly concerned with real-time detection of violent events in live, high-traffic settings, where latency, robustness, and computational efficiency are all critical. The problem is examined along three tightly coupled aspects: (i) identifying people and relevant objects under visual uncertainty, (ii)

recognizing actions and interactions as they unfold temporally, and (iii) performing continuous inference on live surveillance streams. Consequently, feature extraction emerges as a central bottleneck—affecting both accuracy and runtime—when building practical anomaly-detection systems at scale. Despite extensive global research, a notable gap persists in achieving high-accuracy violence detection on resource-constrained edge devices commonly used in academic and budget-limited deployments (e.g., Raspberry Pi, Jetson Nano). Many existing techniques impose significant computational cost and often rely on traditional machine-learning components that do not fully exploit modern end-to-end learning, resulting in time-intensive processing and suboptimal accuracy. To address these limitations, this work proposes a streamlined, deep-learning–based HAR approach for Closed-Circuit Television (CCTV) feeds targeting violence recognition in crowded, fast-moving scenarios. Specifically, an efficient convolutional-neural-network backbone—MobileNet—is leveraged to obtain real-time or near-real-time inference while maintaining competitive recognition performance. The overarching aim is a system that is practical to deploy, robust to common surveillance artifacts, and scalable across cameras without prohibitive annotation or engineering costs.

1.1 Problem Framing and Design Principles

In operational terms, violence recognition can be cast as a spatiotemporal classification problem over short video segments sampled from a live stream. Each segment is mapped to a probability of violent activity, and segments exceeding a calibrated threshold trigger alerts for human review. Designing such a model entails several considerations consistent with the challenges noted earlier; Semantic Uncertainty and Weak Supervision: Because the notion of "violence" is context dependent, tolerance for ambiguous labels and fuzzy inter-class boundaries is required. Data augmentation, temporal smoothing, and consensus labeling strategies can mitigate label noise while improving generalization to new environments [27, 28]. Limited Labeled Data: Manual video annotation remains costly [31]. Transfer learning from large-scale action datasets, self-supervised pretraining, and judicious use of weak labels can reduce dependence on exhaustive manual annotation and improve data efficiency. From Handcrafted To Learned Features: Replacing hand-engineered features with lightweight deep backbones reduces the feature-engineering burden and typically yields better domain transfer, provided the architecture is compact enough for edge inference [3]. Edge Deployability: Real deployments demand low latency, modest memory footprints, and resilience to frame drops or bandwidth fluctuations. Efficient models (e.g., MobileNet) combined with temporal ensembling or lightweight sequence modeling offer a practical balance between accuracy and speed on devices like Raspberry Pi and Jetson Nano. Crowd robustness: Occlusion-heavy scenes require representations that remain discriminative under partial visibility, camera motion, and illumination changes [22, 23]. Careful pre-processing (stabilization, dynamic resizing), frame-windowing, and threshold hysteresis can help stabilize decisions in cluttered footage.

1.2 Proposed Approach (High-Level)

Our framework ingests CCTV video, samples short overlapping clips, and processes them with a MobileNet-based feature extractor tailored for efficiency. Per-clip predictions are temporally aggregated using a sliding window to suppress spurious spikes and to capture brief yet informative motion patterns associated with violent incidents. Transfer learning initializes the backbone with weights pretrained on large video/image corpora, reducing the need for massive labeled datasets. The resulting system is designed to operate continuously, flagging segments that exceed a calibrated violence probability while logging timestamps and crops to aid rapid operator triage. Although lightweight by design, the model remains extensible: optical-flow cues, simple temporal shift mechanisms, or compact recurrent layers can be incorporated when additional temporal modeling is required—without sacrificing edge feasibility.

1.3 Practical Relevance of HAR Modalities

While our emphasis is on visual sensor–based recognition, the broader HAR ecosystem includes non-visual and multimodal pathways [19, 20]. Visual sensors provide rich spatial context and are indispensable for forensic review and situational awareness, delivering 2D/3D imagery [16]. Non-visual wearable sensors—already prevalent in phones, bands, and watches—offer privacy-preserving motion traces and can complement camera views in specific environments [17, 33]. In integrated settings such as assisted living or industrial sites, multimodal fusion can improve robustness to occlusions and lighting while reducing false alarms [1]. The

proposed visual model remains compatible with such extensions.

1.4 Contributions

The principal contributions of this work are summarized as follows:

- This work introduces a real-time surveillance framework that detects violent events in live video with a MobileNet-based model, targeting low-power deployments without compromising recognition accuracy.
- Continuous manual monitoring is error-prone and exhausting; our system prioritizes segments likely to be violent, reducing operator load while preserving transparency for decision review.
- The approach emphasizes precise localization and activity labeling in crowded settings with brief interactions and common occlusions, and remains compatible with crowd analytics and group-level behavior modeling.
- In real incidents, response time is critical. Our design stresses fast inference and stable temporal aggregation so that alerts are produced quickly and reliably from streaming data.
- By leveraging transfer learning and compact architectures, the system reduces reliance on large custom datasets [31] and minimizes the need for brittle handcrafted features [3], improving portability across cameras and sites.

1.5 Paper organization

The remainder of the paper is structured as follows. Section 2 reviews the relevant literature and situates our work within recent advances in anomaly and violence detection. Section 3 details the proposed methodology and experimental setup, including data preparation, model configuration, and deployment considerations. Section 4 presents simulation and empirical results with an in-depth discussion of findings and limitations. Section 5 concludes the study and outlines directions for future research.

2 Related Work

Research on activity understanding spans classical background-subtraction models, wearable sensor analytics, and modern deep architectures for spatiotemporal representation learning. Prior efforts are grouped below by sensing modality and modeling strategy, with emphasis on their implications for violence and anomaly recognition in surveillance; the original references are preserved. Early vision models typically segment foreground regions and then analyze motion within those regions. For instance, [4] proposed a four-stage brutality detection scheme for surveillance videos: (i) locate object regions via background subtraction and suppress artifacts with morphological filtering; (ii) estimate optical flow using a Combined Local–Global approach regularized by Total Variation (CLG-TV); (iii) derive a Motion Co-occurrence Feature (MCF) that summarizes the strength and co-occurrence of motion vectors within detected regions; and (iv) classify segments as violent or non-violent based on the MCF descriptor. Such models remain attractive for their interpretability and low computational footprint. A complementary thread employs intelligent signal processing and neuro-fuzzy reasoning. The Adaptive Neuro-Fuzzy Inference System (ANFIS) in [1] targeted Activities of Daily Living (ADLs) from a tri-axial IMU, assessing performance primarily via Root Mean Square Error across ANFIS parameters and reporting a headline recognition accuracy of 98.88%. Likewise, [13] addressed wearable-sensor motion classification for human activity recognition with a focus on reliable, automated monitoring—particularly valuable for elderly care scenarios where continuous labeling is infeasible. Dataset-driven vision studies have explored increasingly complex human interactions. The system in [7] recognized eight intricate activities from the BIT-Interaction corpus—bow, boxing, handshake, high-five, hug, kick, pat, and push—highlighting the challenges of modeling dyadic actions and short, discriminative motion bursts. Robust foreground extraction under environmental variation was a central concern in [8], which introduced a "twin background modeling" strategy to mitigate effects from swaying tree branches and illumination shifts. By projecting from 2D imagery to a 1D representation and adopting Manhattan distance for matching, their approach reduced computation time, improved detection rates, and lowered error on the Change Detection 2014 benchmark compared with common baselines. Beyond pure vision, multimodal fusion surveys (e.g., [19]) catalog techniques that combine heterogeneous evidence at data, feature, or decision levels. Representative methods include weighted averaging, Kalman filtering, Dempster–Shafer reasoning, graph-based schemes, and deep canonical correlation formulations, each trading off robustness, generality, and uncertainty reduction in different ways.

View invariance—critical for fixed CCTV with varying subject orientation—has been treated explicitly. The framework in [24] achieved view-invariant recognition through a three-stage model: (i) person detection and localization via background subtraction, (ii) feature extraction, and (iii) sequence modeling with Hidden Markov Models (HMMs). The feature design mixed contour-based distance signals, optical-flow motion cues, and rotation-invariant local binary patterns, and was validated on multiple datasets including an in-house multi-view set, KTH, i3DPost, and MSR viewpoint collections. Deep learning has driven significant progress in streaming video. The method in [29] processed non-stationary surveillance feeds by first extracting frame-level deep features with a pre-trained CNN, then modeling temporal evolution via an optimized Deep Autoencoder (DAE). In smartphone-centric sensing, [12, 31] built a multi-sensor HAR classifier using the accelerometer, gyroscope, and gravity signals, achieving strong accuracy across six core activities—particularly walking, running, sitting, and standing—underscoring the practicality of commodity devices for pervasive monitoring. Temporal structure and activity periodicity have also been leveraged. The work in [35] differentiated non-periodic activities with complex motion states (NP_CMS) from weakly periodic activities (WP_CMS), formulating a Human Activity Detection and Recognition (HADR) model that first generated candidate intervals (detection) and then recognized activities over those spans—an approach well suited to sports and other structured domains. Architecturally, 3D convolutions and hybrid CNN–RNNs remain prominent. The C3D-based CCTV recognition reported in [36] (see also [9]) exemplifies end-to-end spatiotemporal filtering in surveillance video. In the specific context of violence recognition, [5] introduced a lightweight computational model that distinguishes violent from non-violent behavior using a CNN coupled with a bidirectional LSTM; comparisons against prevailing baselines highlighted the model's efficiency–accuracy balance. Event-based sensing has provided an alternate route: [21] exploited a Dynamic Vision Sensor (DVS) that outputs pixel-level intensity changes (rather than full frames), proposing a function-based model that delivered promising activity recognition results with sparse, low-latency streams. Hybrid multi-stream deep models further improve discriminative power. A "deeply coupled" ConvNet in [24] combined two pathways: (a) RGB frames processed by a CNN followed by a Bi-LSTM for end-to-end spatiotemporal learning, and (b) a single dynamic motion image fine-tuned with top CNN layers to capture compact temporal summaries. Reported gains included 2% on SBU Interaction, 4% on MIVIA Action, 1% on MSR Action Pair, and 4% on MSR Daily Activity over comparable state-of-the-art methods. Recent anomaly-detection models rely on deep features plus efficient temporal reasoning. The framework in [30] extracted spatiotemporal cues by passing sequences through a pre-trained CNN and validated on UCF-Crime and UCF-Crime2Local, reporting accuracy improvements of 3.41% and 8.09% respectively over strong baselines—evidence that generic deep features, when combined with streamlined temporal models, can scale to long, unconstrained surveillance videos. Non-RGB modalities and alternative representations broaden the design space. In radar-based HAR, [34] treated spectrograms as time-sequential vectors and proposed a compact architecture combining 1D-CNNs with recurrent layers; besides achieving top accuracy, the model used fewer parameters than prevalent 2D-CNN solutions. Classical image descriptors continue to be relevant in segmentation and pre-processing: [32] fused Histogram of Oriented Gradients (HOG) with Local Binary Patterns (LBP), with LBP alone yielding 95.6% segmentation accuracy in their experiments—useful for sharpening regions of interest before high-level recognition. Transfer learning on strong backbones remains a common practice. Using a pre-trained ResNet-50, [11] extracted video descriptors from both the Global Average Pooling and Fully Connected layers, illustrating how multi-layer embeddings can benefit downstream classification. For forensic robustness, [15] examined system performance when operating parameters are unknown and when manipulations are applied to JPEG-compressed imagery—two realistic deployment challenges that can degrade feature stability if unaddressed.

Adjacent problems—such as fake-media analysis and secure information hiding—have contributed techniques that may cross-pollinate HAR. A two-stream strategy in [6] analyzed both frame-level and temporal cues in compressed Deepfake videos, while [14] proposed a channel-dependent payload partition scheme that increases empirical steganographic security against co-occurrence-based detectors; both lines underscore the value of channel-aware modeling and temporal consistency checks when combating sophisticated distributional

shifts. Finally, low-level enhancement and motion extraction remain vital pre-processing steps for action analysis. In [2], frames were first transformed to HSI color space to boost contrast, after which optical flow–based motion features were computed. The model was evaluated across canonical action datasets—Weizmann, KTH, UCF Sports, and UCF YouTube—demonstrating that careful photometric normalization paired with reliable motion estimation can significantly stabilize recognition under varying illumination and scene conditions. Recent works have focused on enhancing surveillance and sensing systems using advanced feature extraction and deep learning techniques. Authors in [37] proposed a robust framework for video summarization by integrating Zernike moments and R-transform features with deep neural networks, achieving improved efficiency in surveillance video analysis. Extending this line of research, authors in [38] developed an object detection framework for traffic surveillance that demonstrated resilience under challenging conditions, emphasizing robustness in real-world deployments. Similarly, authors in [39] introduced a hybrid deep learning model for real-time object detection and classification in surveillance videos, showing the potential of combining multiple architectures for high accuracy and speed. Complementing these vision-based approaches, authors in [40] surveyed Wi-Fi sensing techniques for human activity recognition, highlighting the challenges and future directions in exploiting wireless signals as a non-intrusive alternative to traditional video-based monitoring. Collectively, these studies highlight three enduring themes: (i) robust foreground/motion cues (e.g., background modeling, CLG-TV flow, DVS events) are essential when scenes are cluttered or illumination varies [2, 4, 8, 21]; (ii) lightweight yet expressive temporal models (DAEs, Bi-LSTMs, C3D, coupled RGB–dynamic streams) offer practical accuracy–efficiency trade-offs for long or live video [5, 24, 29, 30, 36]; and (iii) multimodal sensing and principled fusion improve reliability when single-channel evidence is brittle [1, 12, 13, 19, 31, 34]. These insights motivate our emphasis on efficient deep backbones and temporally stable inference for real-time violence detection in surveillance deployments.

3 Proposed Methodology

Our proposed model is organized into four streamlined stages designed for real-time deployment on resource-constrained hardware. Below is an expanded, fully rephrased description of each stage while preserving the original intent.

3.1 Curate and prepare a violence-activity dataset

The pipeline begins by assembling a representative surveillance-style corpus that includes both normal and violent scenes. The curation step ensures balanced coverage across environments (indoor/outdoor), crowd densities, camera viewpoints, lighting conditions, and motion patterns. Clips are split into training/validation/testing sets, frames are uniformly sampled, and standard pre-processing is performed (resize, optional center/letterbox, normalization). When temporal annotations exist, they are aligned to short segments; otherwise, clip-level labels are propagated to fixed-length windows to support learning under weak supervision.

3.2 Learn discriminative visual features with MobileNet

From the prepared clips, features are extracted using a MobileNet backbone optimized for efficiency. During training, the network ingests pre-processed frames (or short stacks) and learns end-to-end representations that separate violent from non-violent content. Global pooling is used to compress spatial activations, and the resulting vector is passed through lightweight fully connected layers with a softmax/sigmoid head for binary (abnormal vs. normal) decisions. Fine-tuning focuses on later MobileNet blocks to keep computation low while adapting to surveillance dynamics; standard regularization (augmentation, dropout, label smoothing) stabilizes learning and improves generalization.

3.3 Select salient motion using lightweight frame differencing

At inference time—or during rapid batch evaluation—a lightweight frame-differencing module is applied to the incoming stream to prioritize motion-salient frames. Consecutive frames are differenced and thresholded to produce a motion mask; only segments with an activity score above a small threshold are forwarded to the classifier. This gating step suppresses redundant static intervals, reduces the number of frames processed by MobileNet, and therefore conserves compute and energy without sacrificing detection coverage. The result is a stream of salient frames (or short bursts) that concentrate the model's attention on potentially violent transitions.

3.4 Test and Alert (Real-Time Decision Layer)

The selected salient frames are fed through the trained MobileNet classifier to produce a per-segment probability of abnormal (violent) activity. To avoid flicker and false alarms, scores are smoothed over a short sliding window and compared to a calibrated threshold. If the aggregated score indicates abnormal behavior, the system immediately raises an alert—logging the timestamp and optional frame crops—and notifies the designated authority for rapid response. This full loop, from motion gating to decision and alert, operates continuously as illustrated in Figure 1. Efficiency: Frame differencing acts as a compute gate, allowing MobileNet to run primarily on informative segments—well-suited to devices like Raspberry Pi or Jetson Nano. Robustness: Temporal smoothing (e.g., median/EMA over a few segments) mitigates single-frame spikes due to camera shake or illumination changes. Extensibility: The model can incorporate optical-flow cues or compact temporal modules later if needed, without altering the core MobileNet+gating design. In summary, the workflow proceeds as: Dataset curation → MobileNet feature learning/classifier training → Salient-motion selection via frame differencing → Real-time testing and alerting (Figure 1).

3.5 Training phase

Recent convolutional approaches for violence detection have shown promise, but many suffer from large model size, slow inference, and brittle behavior in difficult scenes-e.g., strong shadows, fire-like highlights and reflections, smoke, snow, or fog. These limitations make conventional CNN models impractical for resource-constrained surveillance deployments where compute, memory, and power budgets are tight.

To address these issues, this work introduces an efficient CNN architecture tailored to ambiguous, low-signal (low-SNR) conditions. The core idea is to retain the discriminative power of deep features while eliminating heavy components that inflate latency and overfit to nuisance factors. Concretely, the model (i) replaces dense fully connected stacks with global average pooling and a light classification head, (ii) builds the backbone from depthwise-separable or inverted-residual blocks to minimize parameters and FLOPs, and (iii) employs multi-scale receptive fields (via dilated or mixed-kernel convolutions) so that small, distant cues are still captured reliably. This architecture keeps computation economical and remains sensitive to small objects/events at long range—critical for recognizing subtle precursors to violence or small flame-like artifacts that often confound standard detectors. To further harden the system on visually "messy" footage, a set of pragmatic measures is adopted: (i) chromaticity/illumination robustness—light color-space normalization and adaptive contrast suppress false triggers from shadows, glare, and smoke-tinted scenes; (ii) salient-motion gating—a lightweight frame-differencing or motion-score filter forwards only motion-rich frames to the CNN, conserving compute while preserving brief, informative transitions. Temporal smoothing: short sliding-window aggregation stabilizes predictions against single-frame spikes caused by camera shake or transient noise. Edge readiness: optional quantization/pruning preserves accuracy while improving throughput on devices like Raspberry Pi or Jetson Nano. Although our primary application is violence detection, the same lightweight design principles translate to fire/event detection in ambiguous conditions, where small, distant, and partially occluded phenomena must be recognized quickly and reliably. For curated content (e.g., films), the method can be applied to classify violent vs. non-violent scenes by processing shot-level clips and aggregating the per-clip scores into scene-level decisions. Overall, the proposed strategy yields a compact, real-time CNN that remains effective in challenging environments while meeting the practical constraints of low-cost surveillance systems.

3.6 MobileNet Deep Learning Model Architecture

For abnormal-activity recognition, MobileNet serves as the backbone classifier. The choice over larger CNNs such as AlexNet, VGG-16, and GoogLeNet (Inception v1) reflects MobileNet's lightweight design, which is well suited to mobile and embedded deployments where computation, memory, and power are constrained. Efficiency is achieved by replacing standard convolutions with depthwise-separable convolutions: a depthwise spatial convolution is applied channel-wise, followed by a pointwise 1×1 convolution that mixes information across channels. This factorization markedly reduces multiply–accumulate operations and parameter count, lowering latency and model size without sacrificing representational capacity [10]. Figure 2 illustrates the overall architecture. In the configured stack (about 30 layers), stride-2 convolutions perform spatial downsampling; depthwise convolutions extract per-channel spatial features; and pointwise

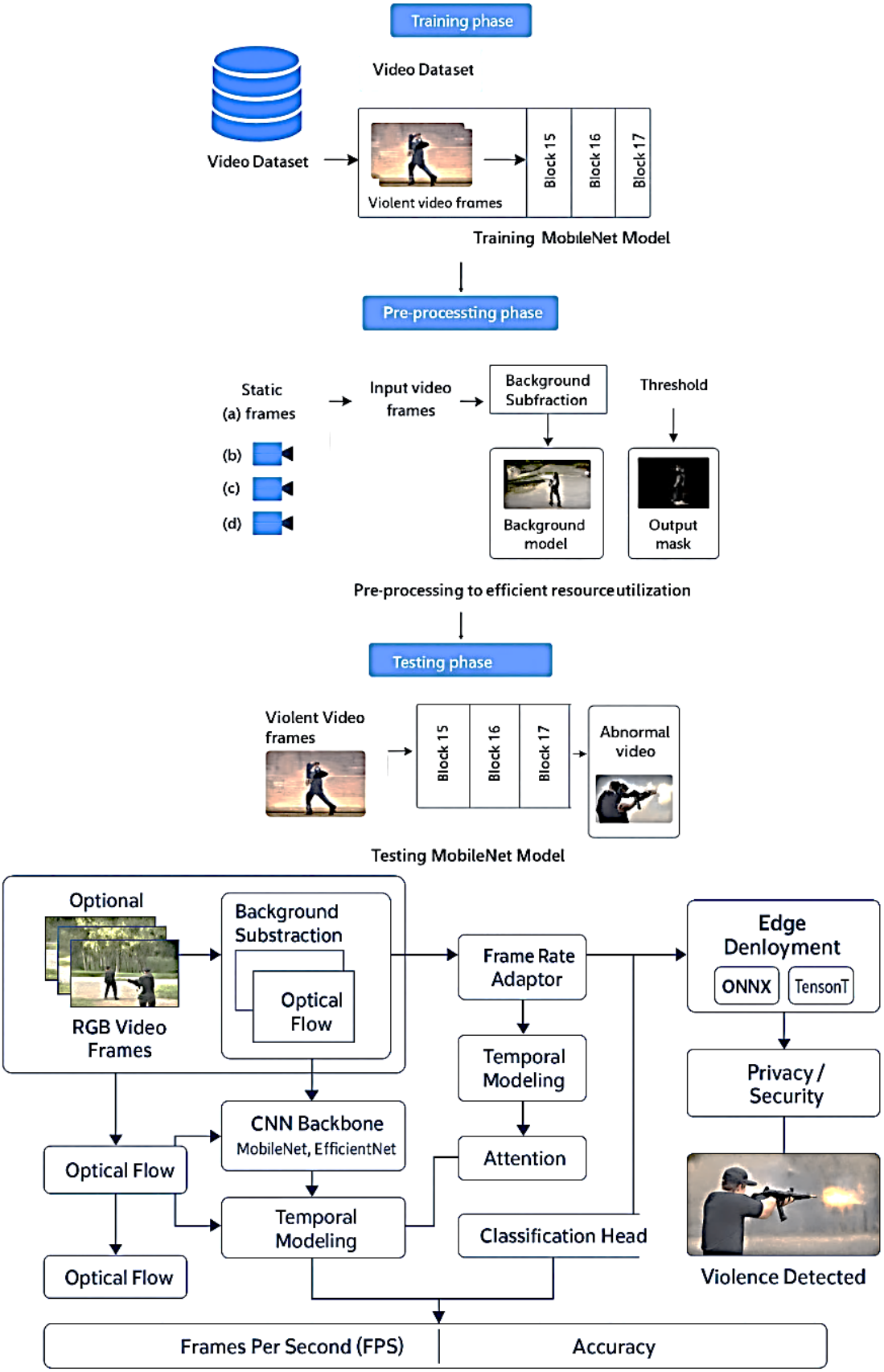

Figure 1. Proposed framework of violent activity recognition.

convolutions expand—often doubling—the channel dimension to form richer embeddings. Each block is followed by batch normalization and a non-linear activation. A global-average-pooling layer aggregates spatial responses, and a compact classification head (a fully connected layer with sigmoid/softmax) outputs the abnormal/normal probability. This design preserves discriminative power while keeping inference cost low—an essential property for real-time surveillance. After dataset preparation and frame pre-processing, MobileNet is trained end-to-end on the task labels. Standard practices—data augmentation, class-balanced sampling, early stopping, and learning-rate scheduling—stabilize optimization and improve generalization. Once training converges, the frozen model is evaluated on unseen video segments. In line with applied deep learning, multiple variants are trained and the model best aligned with data characteristics and deployment constraints is selected. Explored configurations vary learning rate, batch size, and lightweight regularization (e.g., dropout in the classifier head), as well as MobileNet's width multiplier and input resolution to trade accuracy for speed according to data quality and target hardware. The final selection reflects the optimal balance for the intended surveillance scenario. To reduce data requirements and accelerate convergence, transfer learning ("move learning") is employed by initializing MobileNet with weights pre-trained on large-scale image/video corpora and fine-tuning on the abnormal-activity dataset—a well-established strategy that typically yields more robust representations with fewer training iterations. In summary, the MobileNet architecture—with depthwise-separable convolutions, global pooling, and a minimal classification head—delivers the computational economy and accuracy required for abnormal-activity recognition in resource-constrained surveillance systems [18].

3.7 MobileNet Model Fine-Tuning

Recognizing violent activity is inherently difficult: scenes exhibit wide intensity variations, complex crowd dynamics, and diverse camera viewpoints. Crucially, the evidence spans space and time. Spatial cues come from a single frame (e.g., appearance, attire, scene context), whereas temporal cues emerge only across multiple frames (e.g., sudden accelerations, aggressive interactions). For instance, a single CCTV frame near an ATM may reveal a person's clothing or approximate demographics, but only a sequence of frames reveals whether a violent act is unfolding. Earlier traditional models—built on hand-crafted descriptors such as HOG, SURF, and SIFT—struggle to capture these nuanced spatiotemporal patterns. Subtle distinctions (e.g., walking vs. running, a shove vs. crowd jostling) often confound fixed features. Deep learning, by contrast, learns features directly from data and has proven effective for pattern and image recognition, autonomous driving, and medical analysis. The trade-off is that deep models typically require sizable datasets and compute. To bridge this gap with a relatively modest corpus (500 videos across violent and non-violent classes), MobileNet is paired with transfer learning that reuses features from large image corpora and adapts them to the surveillance domain.

3.7.1 Fine-Tuning Strategy

Our fine-tuning procedure tailors MobileNet (see Figure 2) to the abnormal-activity task while preserving its lightweight nature; Initialization (transfer learning): Initialize weights from a MobileNet pre-trained on large-scale imagery (e.g., ImageNet). This provides robust low-level filters (edges, textures, colors) and mid-level semantics (parts, simple configurations), reducing data demands and speeding convergence. Layer freezing and progressive unfreezing: Freeze the earliest stages (low-level filters) for initial epochs to stabilize training. Unfreeze progressively: first the mid-level blocks, then the final blocks, using discriminative learning rates (lower LR for early layers, higher LR for the classifier head). This schedule adapts higher-level features to surveillance appearance without overfitting the backbone. Lightweight classifier head: Replace heavy fully connected stacks with global average pooling followed by a compact dense layer (sigmoid/softmax). This keeps latency and parameters low for edge devices while retaining discriminative power. Temporal evidence aggregation. The pipeline augments a frame-based MobileNet with: salient-motion gating (frame differencing; §Methodology) to forward only motion-rich frames, and clip sampling + pooling to sample K frames per clip, score each, and aggregate (mean/max or short EMA) into a robust clip-level decision—capturing brief, bursty motions without excessive compute.

- **Regularization and Data Augmentation**
  Spatial: random resized crops, horizontal flips (when label-safe), mild color jitter, blur/ISO noise, and illumination shifts to mimic surveillance artifacts. Temporal: random frame stride/offset to expose the model to varied motion phases.

Model: dropout on the head and weight decay to reduce overfitting. Class imbalance handling: Apply class-balanced sampling or weighted/focal loss if violent segments are rarer than normal ones, improving recall on minority events without inflating false alarms. Optimization and schedules: Use SGD with momentum or Adam/AdamW; adopt a warm-up followed by cosine decay or step LR schedule. Early stopping on a validation split prevents over-training. Optionally apply temperature scaling on validation data to calibrate probabilities. Choose an operating threshold that balances precision/recall for the intended response policy (e.g., higher recall for safety-critical monitoring). MobileNet's efficiency stems from depthwise separable convolutions—a depthwise (per-channel) spatial filter followed by a 1×1 pointwise projection. This factorization sharply reduces multiply-accumulate operations and parameters relative to standard convolutions, making it ideal for our setting. In practice, MobileNet's width multiplier and input resolution are tuned to balance accuracy and speed according to hardware constraints and the visual complexity of the site. Using the fine-tuning strategy described above, the model captures appearance cues from individual frames while incorporating short-range temporal context through clip aggregation and motion gating. The result is a compact, responsive detector suitable for real-time surveillance, even when the training set is limited and the operating environment involves heavy crowding, viewpoint shifts, shadows, smoke, rain, snow, or other visual disturbances.

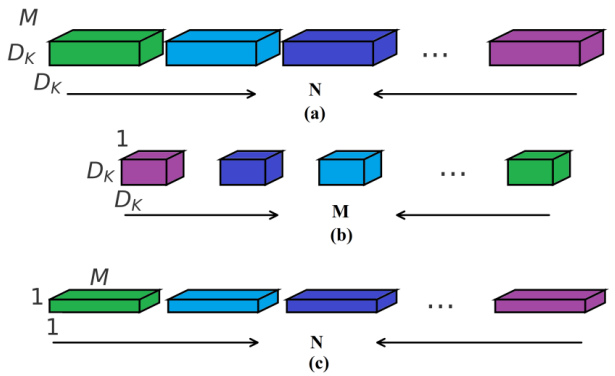

Figure 2. (a) Illustration of Standard Convolution Layer Filters, (b) Depth wise Convolutional Layers Filters, (c) Example of 1 × 1 Convolutional Layers Filters.

3.8 Video Dataset

This study employs the Hockey Fight dataset for violent-activity recognition, originally introduced in [31]. The collection comprises 1,000 short video clips sourced from National Hockey League (NHL) broadcasts—500 labeled as violent and 500 as non-violent. Each clip is recorded at 50 frames per second (FPS) with a spatial resolution of 360 × 288 pixels. Violent frames. For any frame-level protocol, a violent frame denotes a frame extracted from a clip labeled as violent. Non-violent frames. Conversely, a non-violent frame is any frame extracted from a clip labeled as non-violent. In our model, clips are decoded and frames are sampled uniformly (or by motion saliency; see §3.9) to build training and evaluation sets. Labels are assigned at the clip level, and when frame-level supervision is required, the clip label is propagated to sampled frames. This strategy aligns with real surveillance conditions, where precise per-frame annotations are uncommon but clip-level labels are readily available.

3.9 Pre-Processing Phase

To make effective use of limited compute on visual-surveillance hardware, a lightweight pre-processing stage is introduced before classification. Because video streams are continuous and data-intensive, this stage filters out redundant content and emphasizes motion-rich segments that are more likely to contain violent events.

- **Steps**
  Decoding & normalization: Frames are decoded, temporally normalized (optional down-sampling), and resized to match the MobileNet input resolution; pixel intensities are standardized. Motion pre-filtering: A fast frame-differencing score between consecutive frames is computed to estimate motion energy. Frames (or short bursts) with negligible motion are temporarily skipped, while motion-rich regions are retained for downstream analysis. Background/mask assistance: When available, a coarse foreground mask from background subtraction (see §3.10) suppresses static background and reduces false triggers from illumination flicker or camera jitter. ROI cleanup (optional): Light morphological operations remove small artifacts; simple bounding regions may be used to focus on crowd areas. Rationale: violent activities almost always entail observable motion (e.g., rapid limb movement in fights).

By directing the classifier's attention to such frames, the system reduces latency and energy consumption while improving the effective signal-to-noise ratio prior to MobileNet inference.

3.10 Background Subtraction Method

Background subtraction is a standard approach for motion detection in many computer-vision applications. A Gaussian Mixture–based Background Segmentation method is adopted following [26]. The core idea is to model each pixel's intensity over time as a mixture of Gaussians updated recursively, where components with the highest weight and lowest variance represent the background, while outliers correspond to foreground (moving) pixels.

- **Key Properties** (per [26])
  Adaptive modeling: The algorithm automatically selects an appropriate number of Gaussian components per pixel and updates them online, enabling the background model to adapt to gradual scene changes. Robustness to nuisance factors: By maintaining multiple modes, the method handles repetitive motions (e.g., swaying jerseys, specular highlights), moderate illumination changes, and sensor noise better than single-model baselines. Efficient masks: The resulting binary foreground mask highlights moving regions that likely correspond to people and interactions of interest. In our model, the foreground mask serves two purposes: (i) it gates which regions/frames proceed to MobileNet (complementing the frame-differencing pre-filter), and (ii) it reduces background clutter prior to feature extraction. Qualitative examples of the background subtraction output used in our system are presented in Figure 3.

3.11 Input Video Frames (static vs. motion)

Input video frames: For training and evaluation, clips from the dataset are decoded into frames and supplied to the proposed model to discriminate between violent and non-violent events. Labels are assigned at the clip level and, when needed, propagated to sampled frames used by the classifier. *Static frames*: After applying background subtraction (see §3.10), consecutive frames that exhibit no appreciable motion—i.e., their motion energy falls below a small threshold over an MMM-frame window—are designated static. Such frames are temporarily skipped during training/inference to conserve compute without affecting recognition fidelity. *Motion frames*: Frames (or short bursts) that the background model identifies as containing motion are treated as motion frames. Examples include people moving, arm/leg swings, or rapid local changes consistent with interactions. These motion-rich frames

Algorithm 1: Segmentations of Shot

Input: Input video stream
Output: Segmented shots
foreach *two consecutive input video frames* (f_i, f_{i+1}) **do**
  Apply Gaussian blur on $(f_i - f_{i+1})$ to remove noise;
  Compute pixel-wise absolute difference of frames:;

$$D_{\text{image}} = \frac{1}{N} \sum_{i=1}^{N} \left| f_{1(i)} - f_{2(i)} \right|$$

  if $D_{image} \leq 0.1$ **then**
    Consider f_i and f_{i+1} as the same shot;
  else
    f_i and f_{i+1} are salient shots;
  end
end

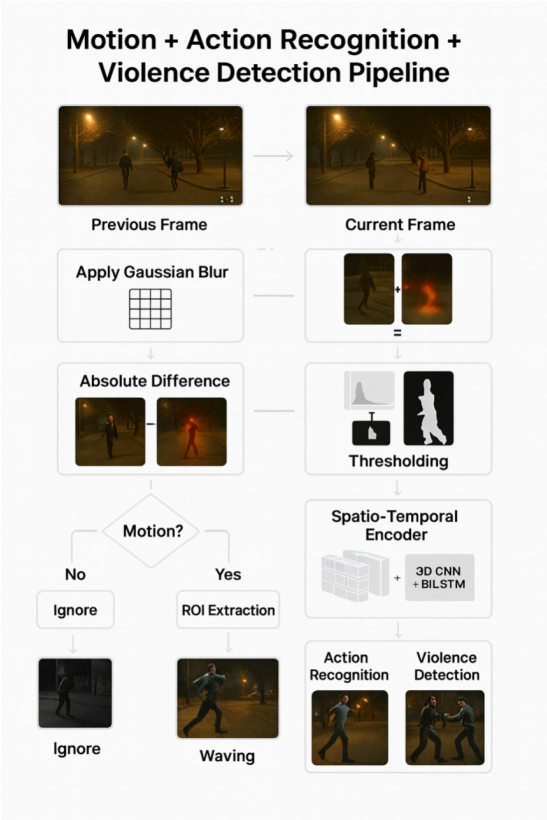

Figure 3. Background subtraction algorithm.

are forwarded to the MobileNet classifier and, when used in short sequences, aggregated into stable clip-level decisions. A light hysteresis is applied so that brief dips in motion do not prematurely switch frames back to "static."

3.12 Simulation Parameters

All experiments were implemented in Python using the TensorFlow API with Keras front-end and NumPy for numerical routines. Training and inference were conducted on a workstation equipped with an Intel Core i5 CPU, 24 GB RAM, a GPU with 8 GB of VRAM, running Windows 10 Pro. The primary backbone for all trials was MobileNet, as described in §3.6. To ensure reproducibility and balance accuracy with computational efficiency, a concise set of simulation parameters was defined (summarized in Table 1). Each parameter was assigned values appropriate to the constraints of our surveillance setting and the characteristics of the dataset. The parameters fall into the following categories; Model configuration: MobileNet variant, width multiplier, input resolution, activation/normalization choices, and classifier head (global average pooling + dense output). Optimization: optimizer type (e.g., SGD/Adam family), base learning rate, schedule (warm-up/step/cosine), weight decay, and early-stopping criteria. Training protocol: batch size, number of epochs, train/validation/test split, class-balancing strategy or loss weighting (if applicable). Data handling & augmentation: frame resize/normalization, color/illumination jitter (label-safe), random crops, flips (when valid), and temporal sampling stride. Motion gating & background modeling: frame-differencing threshold, aggregation window length for motion scores, and key background-subtraction settings (per §3.10) used to produce foreground masks. Evaluation metrics & compute: primary metrics (Accuracy) and computational indicators (throughput/FPS, average latency per frame/clip, and memory footprint) to capture the accuracy–efficiency trade-off. Table 1 is organized into two columns: the Parameter name and its Value. It lists the software stack and hardware environment alongside the model/training settings used in our experiments. This arrangement highlights how choices affecting performance (accuracy) and computation (speed/memory) were selected to achieve reliable real-time behavior on resource-constrained devices.

Table 1. Simulation parameters.

Parameter	Value
Operating system	Microsoft Windows 10
Coding	Python
Libraries	Numpy, Time, SciPy, PyLab, Matplotlib, Opencv
Implementation Environment	TensorFlow, Keras
Dataset	Hockey Fight

3.13 Performance Parameters

To assess the classifier, the standard set of metrics used in binary recognition tasks is reported: Accuracy, Precision, Recall, True Positive Rate (TPR), True Negative Rate (TNR), and False Positive Rate (FPR). Let the confusion-matrix terms be; TP (True Positive): the model predicts violent and the clip is actually violent. TN (True Negative): the model predicts non-violent and the clip is actually non-violent. FP (False Positive): the model predicts violent but the clip is actually non-violent, and FN (False Negative): the model predicts non-violent but the clip is actually violent. The corresponding formulas are:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{TPR} = \frac{TP}{TP + FN} \tag{4}$$

$$\text{TNR} = \frac{TN}{TN + FP} \tag{5}$$

$$\text{FPR} = \frac{FP}{FP + TN} = 1 - \text{TNR} \tag{6}$$

- **Interpretation**
  Precision captures how often violent predictions are correct (low FP), Recall/TPR captures how often true violent events are detected (low FN), TNR/Specificity measures how well non-violent clips are correctly rejected, FPR quantifies false alarms among truly non-violent clips, and Accuracy summarizes overall correctness but can be misleading under class imbalance; therefore, precision/recall (and their trade-off) are also reported for a balanced view of performance.

4 Experimental Results

Closed-circuit cameras have been used in surveillance for decades, typically feeding one or more monitors

overseen by a limited number of operators. In many deployments, operators review footage after an incident rather than monitoring every stream continuously in real time. Cameras are installed across diverse sites—some driving a single monitor, others multiplexed across many streams—yet operator attention can be uneven, and coverage of the most critical areas may not always be prioritized. As a result, recorded video often serves primarily as post-hoc evidence, with several practical drawbacks: (i) personnel may miss the moment an incident begins, (ii) exhaustive review of long recordings is time-consuming, and (iii) by the time a perpetrator is identified, they may already have left the scene. These limitations motivate an automated, robust, and accurate model that can analyze human activities continuously, flag abnormal events, and reduce reliance on constant human vigilance. Over the last decade, interest in automated surveillance—powered by computer vision—has grown steadily. Modern systems network embedded sensors with cameras to detect both human and non-human activity under real-world conditions, including adverse weather, low illumination, and dense crowds. The goal is to model typical video patterns (normal, low-crowd scenes) and to spot deviations (abnormal, high-crowd or violent events) as they occur. In environments that are difficult or unsafe for on-site personnel, advanced cameras (e.g., with night-vision lenses) can operate continuously and, upon detecting abnormal activity, trigger alarms to prompt immediate response. Our proposed system aligns with this vision: it recognizes human objects and activities (normal/abnormal), detects and tracks moving targets from fixed camera platforms, and raises alerts in real time—day or night—across long viewing distances.

4.1 Experimental Setup

The software, hardware, and training configurations employed in the experiments are summarized in Table 2. Each parameter is reported together with its chosen value to ensure reproducibility. In brief, our stack comprises Python with deep-learning libraries (e.g., TensorFlow/Keras and supporting utilities), trained and evaluated on a workstation consistent with the specifications listed in Table 2. Hyperparameters (learning rate, batch size, epochs, etc.) are selected to balance accuracy and computational cost, reflecting the constraints of surveillance deployments.

Table 2. Experimental setup.

Parameter	Value
Operating system	Microsoft Windows 10 Pro
Coding	Python
Libraries	Numpy, Time, SciPy, PyLab, Matplotlib, OpenCV
Implementation Environment	TensorFlow, Keras
Dataset	Hockey Fight

4.2 Video Dataset

We evaluate the method on the Hockey Fight dataset, which contains 1,000 short NHL clips: 500 labeled fight and 500 labeled non-fight. Each clip consists of 50 frames at a spatial resolution of 360 × 288 pixels. Fight clips depict on-ice altercations, while non-fight clips capture routine gameplay and related activity in the same environment. Representative frames are shown in Figure 4. This pairing supports reliable benchmarking of violent-scene recognition within sports footage.

Figure 4. Samples of hockey fight dataset.

4.3 Pre-Processing

Before reporting CNN results (e.g., VGG-16, AlexNet) on the Hockey Fight dataset, videos are converted into image frames, since CNN backbones accept images as input. A conventional 75% / 25% split is used for training and testing, with clips randomly partitioned, decoded to frames, and resized/normalized to the target input resolution. The resulting images are then fed to the CNN, and clip-level labels are propagated to sampled frames. This procedure ensures a clean and consistent input representation across all evaluated models.

4.4 Performance of the AlexNet Model

AlexNet—the ILSVRC-2012 winner—established the effectiveness of deep CNNs over handcrafted features. It comprises five convolutional layers followed by two fully connected layers, using ReLU activations (introduced there at scale). Conceptually, it can be viewed as a deeper, GPU-trained evolution of LeNet; hyperparameter refinements inspired the subsequent 2013 ILSVRC winner (ZF-Net). In the

experiments, AlexNet is employed in transfer-learning mode by replacing the final classification layer with a two-class head (normal vs. abnormal) and fine-tuning on the Hockey Fight frames. During training, both training and testing losses decrease steadily with epochs. Initially, the loss is high while the model is still learning salient patterns; after several epochs, learning stabilizes. By epoch 110, the loss plateaus and the model reaches a reported testing accuracy of 0.88 ($\approx$ 88%). A summary appears in Table 3; the model diagram and learning curves are shown in Figures 5 and 6, respectively. The experiment with 110 epochs and a 1e-6 learning rate yields an accuracy reported as 0.889999 ($\approx$ 0.89). For completeness, precision, recall, and F1-score are also reported for each class: Abnormal (violent)—Precision = 0.91, Recall = 0.86, F1 = 0.89; Normal (non-violent)—Precision = 0.87, Recall = 0.92, F1 = 0.89. The corresponding loss at convergence is reported as 2.480 (see Figure 6). Taken together, these results indicate that a transfer-learned AlexNet provides a solid baseline on this dataset, capturing discriminative cues of violent versus non-violent scenes with stable generalization once sufficient epochs are observed.

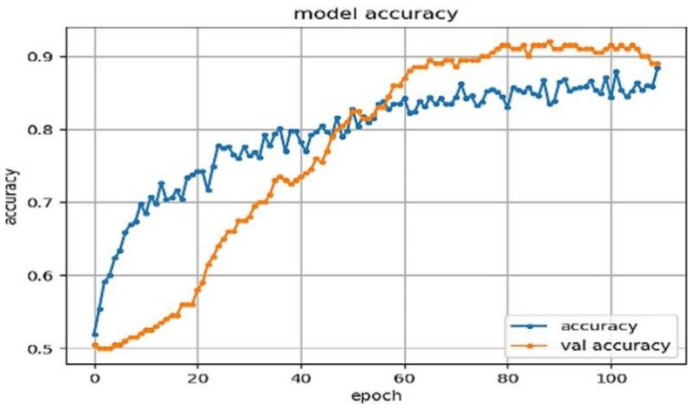

Figure 5. Training accuracy progress of accuracy of AlexNet model.

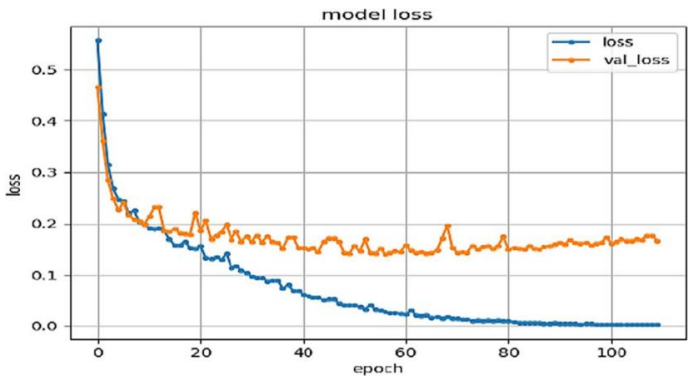

Figure 6. Training loss progress of AlexNet model.

4.5 Performance of the VGG-16 Model

The VGG-16 architecture—introduced by the Visual Geometry Group (Oxford)—demonstrated that stacking small 3×3 convolutional filters throughout the network can approximate larger receptive fields while improving optimization stability and performance. Owing to its simple, uniform design and strong generalization, VGG-16 remains a widely used baseline. In this setting, the standard configuration is employed, comprising 13 convolutional layers interleaved with 3 max-pooling stages. Training protocol and dataset: VGG-16 was trained on the Hockey Fight dataset (see §4.2). As with other models in this study, clips were converted to frames and split 75%/25% for training and testing. The run depicted in Figure 7 used 110 epochs with a learning rate of 1e-6. Overall accuracy: VGG-16 achieved a test accuracy of 0.96 (see Table 3). In the run shown in Figure 7, the model attained 0.96499 ($\approx$ 96.499%) accuracy at convergence. Training dynamics: The evolution of training accuracy and loss is presented in Figures 8 and 9. Under the same 110-epoch, 1×10^{-6} learning-rate schedule, the final reported loss was 0.1669 (see Figure 8), indicating stable optimization and effective feature learning on this dataset. Per-class metrics: To provide a more comprehensive evaluation beyond accuracy, precision, recall, and F1-score are reported for each class: Abnormal (violent)—Precision = 0.96, Recall = 0.97, F1 = 0.97; Normal (non-violent)—Precision = 0.97, Recall = 0.96, F1 = 0.96. These results (also summarized alongside accuracy in Table 3) show that VGG-16 delivers high and well-balanced performance across both classes on the Hockey Fight dataset, with low loss and strong precision/recall trade-offs under the specified training regime.

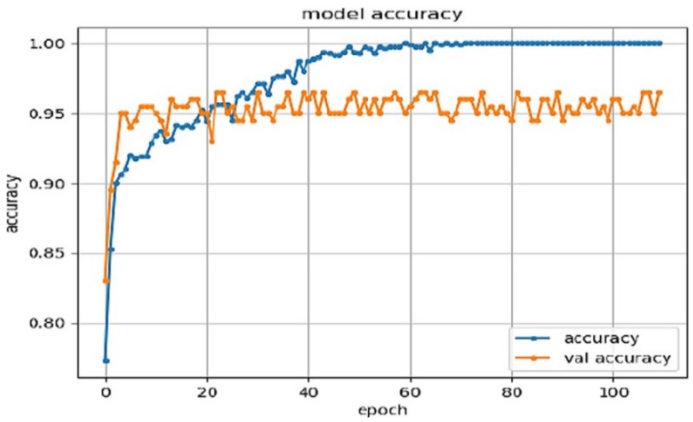

Figure 7. Training accuracy progress of accuracy of VGG-16 model.

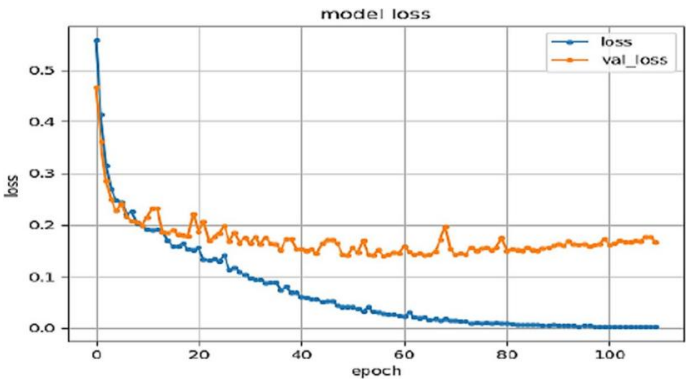

Figure 8. Training loss progress of VGG-16 model.

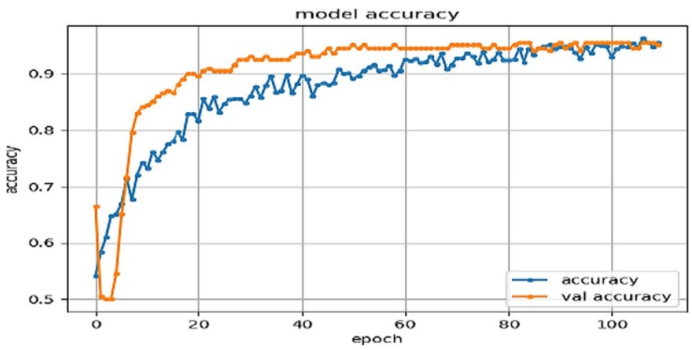

Figure 9. Training accuracy progress of GoogleNet model.

4.6 Performance of the GoogleNet (Inception-V1) Mod

Architecture Overview: GoogleNet—winner of ILSVRC 2014—introduced the Inception module, a "network-within-a-network" block that processes features at multiple receptive-field sizes in parallel and then concatenates the results. A key ingredient is the 1×1 convolution, which acts as a bottleneck for dimensionality reduction and feature mixing, substantially cutting computation while preserving representational power. The canonical Inception-V1 configuration stacks these modules to form a 22-layer deep network containing nine inception blocks. Subsequent refinements, such as batch normalization and architectural tweaks, yielded Inception-V2 and V3; however, the original Inception-V1 (GoogleNet) is employed in this work. Training setup and dataset: GoogleNet is trained on the Hockey Fight dataset following the same protocol used for the other backbones (§4.2–4.5). The run depicted in Figure 9 used 110 epochs with a learning rate of 1e-6; accuracy and loss trajectories appear in Figures 10 and 11, and summary statistics are reported in Table 3. Overall results: GoogleNet achieves a test accuracy of 0.94, with the representative run in Figure 9 reaching 0.94999 (≈ 94.999%). The final reported loss for this configuration is 2.92416 (see Figure 10), reflecting stable convergence under the specified schedule. Per-class breakdown: To characterize performance beyond overall accuracy, precision, recall, and F1-score are reported for both classes: Abnormal (violent)—Precision = 0.94, Recall = 0.96, F1 = 0.95; Normal (non-violent)—Precision = 0.96, Recall = 0.94, F1 = 0.95. These balanced per-class scores indicate that Inception-V1 maintains strong discriminative capability for both violent and non-violent scenes under the same training regimen, offering a competitive accuracy–efficiency trade-off on this dataset.

4.7 Performance of the MobileNet Model

Training Protocol and Dataset: MobileNet was trained on the Hockey Fight dataset under the same preprocessing and train/test split described earlier (§4.2–4.3). The summary configuration in Table 3 reports a 100-epoch run, while the figures present a representative 110-epoch run with a learning rate of (see Figures 11-13). In both cases, the classifier head consists of global average pooling and a lightweight dense layer for the two classes (normal vs. abnormal). Why MobileNet: For abnormal-activity recognition, MobileNet is deployed—a compact CNN specifically designed for mobile and embedded settings with limited compute and memory. Unlike heavier backbones (AlexNet, VGG-16, GoogleNet), MobileNet factorizes standard convolutions into a depthwise spatial filter (applied per channel) followed by a pointwise projection. This depthwise-separable scheme drastically cuts parameters and multiply–accumulate operations while preserving discriminative power—making it well suited to real-time surveillance on resource-constrained hardware. Overall Results: MobileNet attains a test accuracy of 0.9666 with a final loss of 0.1329. The accuracy and loss trajectories are plotted in Figures 12 and 13; Figure 11 depicts the network trained for 110 epochs at , achieving the same 0.9666 accuracy. Relative to the other baselines in this study (AlexNet, GoogleNet, VGG-16), MobileNet delivers state-of-the-art performance in our setting while being markedly lighter—an advantageous trade-off for deployment. Per-class metrics: To characterize behavior beyond aggregate accuracy, precision, recall, and F1-score are reported for each class: Abnormal (violent)—Precision = 0.96, Recall = 0.97, F1 = 0.97; Normal (non-violent)—Precision = 0.97, Recall = 0.96, F1 = 0.96. These balanced scores indicate the model simultaneously maintains high recall for violent events (reducing missed incidents) and high precision for non-violent clips (limiting

false alarms). Training dynamics and stability: Loss decreases smoothly and plateaus near the reported 0.1329 (see Figure 12), indicating stable optimization without overfitting under the chosen schedule. The small classifier head and depthwise-separable blocks help maintain low latency while the model converges to strong separation of the two classes. MobileNet pairs top accuracy in our experiments (0.9666) with edge-friendly efficiency, confirming it as a strong candidate for real-time violence detection in practical surveillance deployments where compute and power budgets are tight.

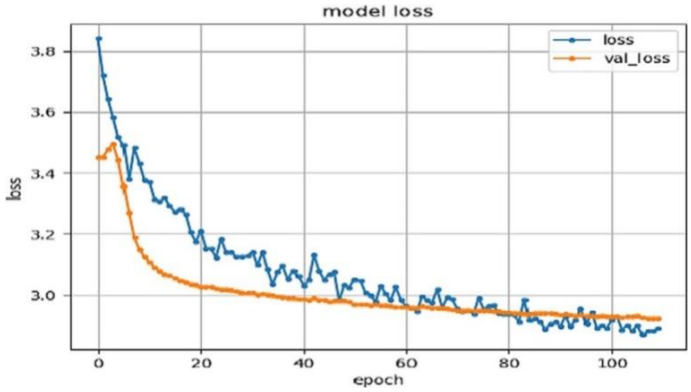

Figure 10. Training loss progress of GoogleNet model.

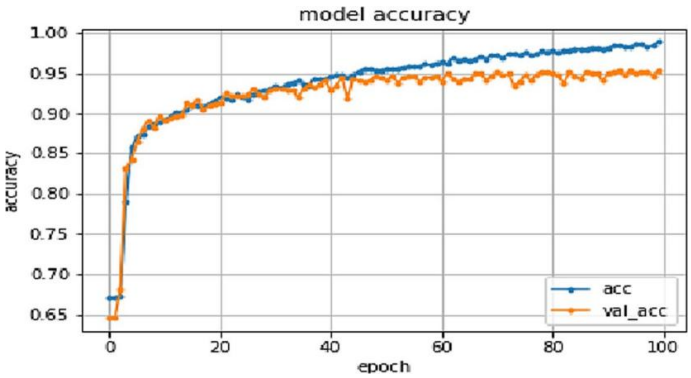

Figure 11. Accuracy progress of the MobileNet Model.

4.8 Comparison of AlexNet, VGG-16, and GoogleNet with the MobileNet Model

Purpose and scope: This subsection justifies our choice of MobileNet by comparing it against three widely used CNN baselines—AlexNet, VGG-16, and GoogleNet (Inception-V1)—on the same preprocessing model and Hockey Fight dataset. Quantitative results (accuracy, loss, precision, recall, F1-score for each class) are summarized in Table 3 and visualized in Figure 13. Unless stated otherwise, all values follow the runs trained for 110 epochs with a learning rate of and are reported in decimal form (≈ percentage).

4.8.1 *Model-by-model summary* (*see Table 3; Figure 13*)

AlexNet-Accuracy: 88.9999%, Loss: 2.480, Abnormal (violent): Precision 0.91, Recall 0.86, F1 0.89, and Normal (non-violent): Precision 0.87, Recall 0.92, F1 0.89. Notes: Establishes a solid transfer-learning baseline but lags in overall accuracy and exhibits higher loss, reflecting limited depth/feature capacity relative to newer architectures. *VGG-16*-Accuracy: 96.499%, Loss: 0.1669, Abnormal: Precision 0.96, Recall 0.97, F1 0.97, and Normal: Precision 0.97, Recall 0.96, F1 0.96. Notes: Strong, balanced per-class metrics and low loss, consistent with VGG-style stacks of 3×3 filters. However, the model is comparatively heavy in parameters and compute. *GoogleNet* (*Inception-V1*)-Accuracy: 94.999%, Loss: 2.92416, Abnormal: Precision 0.94, Recall 0.96, F1 0.95, Normal: Precision 0.96, Recall 0.94, F1 0.95. Notes: Multi-scale inception modules yield competitive accuracy and symmetric class performance, but the reported loss is higher than VGG-16/MobileNet under the same schedule. *MobileNet* (*proposed deployment backbone*)-Accuracy: 96.66%, Loss: 0.1329, Abnormal: Precision 0.96, Recall 0.97, F1 0.97, and Normal: Precision 0.97, Recall 0.96, F1 0.96. Notes: Achieves the highest accuracy among the evaluated models with the lowest loss, while maintaining a markedly smaller computational footprint via depthwise-separable convolutions.

4.8.2 *Comparative observations*

Accuracy & Loss: MobileNet slightly outperforms VGG-16 in accuracy (0.9666 vs. 0.96499) and achieves the lowest final loss (0.1329), indicating stable, data-efficient convergence. GoogleNet follows (0.94999), and AlexNet trails (0.889999). *Per-Class Balance*: VGG-16 and MobileNet both exhibit well-balanced precision/recall across abnormal and normal classes (F1 ≈ 0.96–0.97 for both), which is crucial for minimizing missed violent events (FN) without inflating false alarms (FP). *Practical Deployment*: While VGG-16 is accurate, its parameter count and compute make it less attractive for edge devices. MobileNet offers the best accuracy-to-efficiency trade-off, aligning with real-time surveillance constraints. GoogleNet provides a middle ground in capacity but, in our runs, does not surpass MobileNet/VGG-16 in accuracy or loss. *Consistency Across Runs*: Learning curves in Figures 12 and 13 (and earlier figures for AlexNet/VGG-16/GoogleNet) show smooth convergence for all models after sufficient epochs, with MobileNet and VGG-16 reaching lower terminal losses. Across identical training settings on the Hockey

Fight dataset, MobileNet delivers the best overall performance while remaining compute-efficient, justifying its selection as the primary backbone for abnormal-activity (violence) recognition in resource-constrained surveillance deployments.

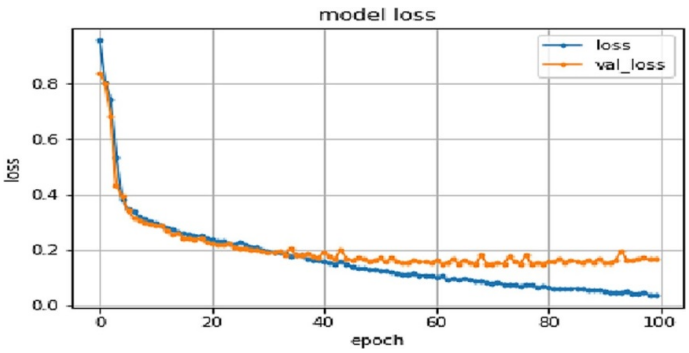

Figure 12. Loss progress of the MobileNet Model.

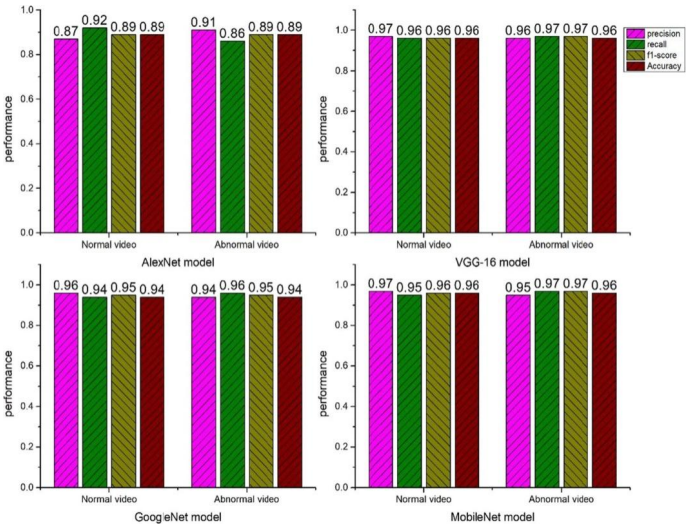

Figure 13. Comparative analysis of MobileNet model with other state-of-art models that are AlexNet, VGG-16, and GoogleNet Model.

To provide a fair justification for selecting MobileNet as the deployment backbone, all four models were trained under the same regimen (110 epochs, learning rate) and evaluated using identical metrics. Detailed numbers appear in Table 3 and the consolidated visualization in Figure 13. GoogleNet (Inception-V1)-Accuracy: 0.94999, Loss: 2.92416, Abnormal (violent): Precision 0.94, Recall 0.96, F1 0.95, and Normal (non-violent): Precision 0.96, Recall 0.94, F1 0.95. MobileNet (proposed)-Accuracy: 0.9666, Loss: 0.1329, Abnormal (violent): Precision 0.96, Recall 0.97, F1 0.97, and Normal (non-violent): Precision 0.97, Recall 0.96, F1 0.96. AlexNet-Accuracy: 0.889999, Loss: 2.480, Abnormal (violent): Precision 0.91, Recall 0.86, F1 0.89, and Normal (non-violent): Precision 0.87, Recall 0.92, F1 0.89. VGG-16-Accuracy: 0.96499, Loss: 0.1669, Abnormal (violent): Precision 0.96, Recall 0.97, F1 0.97, and Normal (non-violent): Precision 0.97, Recall 0.96, F1 0.96. MobileNet attains the highest accuracy (0.9666) and the lowest loss (0.1329), while preserving balanced precision/recall across both classes—crucial for minimizing missed violent events without inflating false alarms. VGG-16 is a very close second in accuracy (0.96499) with low loss (0.1669), but it is substantially heavier computationally. GoogleNet delivers strong, symmetric per-class metrics (F1 = 0.95 for both classes) but does not surpass MobileNet/VGG-16 under this training schedule. AlexNet forms a solid baseline yet lags in overall accuracy and presents higher loss compared with later architectures.

4.9 Computational Complexity

Model size (parameter count) is a primary driver of inference cost—affecting memory footprint, energy usage, and real-time throughput. During testing, all learned parameters participate in the forward pass, so architectures with fewer parameters are generally more suitable for resource-constrained surveillance deployments. AlexNet- Topology: 5 standard convolutional layers + 3 max-pool layers; classifier with two fully connected (FC) layers of 4096 units (followed by the output layer), and Parameters: 62,378,344. VGG-16-Topology: 13 standard convolutional layers + 3 max-pool layers; classifier with two FC layers of 4096 units (plus the final output layer), and Parameters: 138,423,208. GoogleNet (Inception-V1)-Topology: 22-layer network with Inception modules and 1×1 bottlenecks for dimensionality reduction, and Parameters: 10,334,030. MobileNet-Topology: depthwise-separable convolutions (depthwise + 1×1 pointwise) throughout; lightweight global-pooling classifier head, and Parameters: $\approx$ 3,200,000. As visualized in Figure 14, MobileNet has by far the smallest parameter count, followed by GoogleNet, while AlexNet and VGG-16 are significantly larger. This gap directly translates into lower memory and compute needs for MobileNet, enabling higher FPS and reduced latency on embedded hardware—without sacrificing accuracy in our experiments.

- **Quantitative Snapshot (Same Training Regimen)**
  AlexNet: Accuracy 88.99%, Loss 2.480, VGG-16: Accuracy 96.49%, Loss 0.1669, GoogleNet

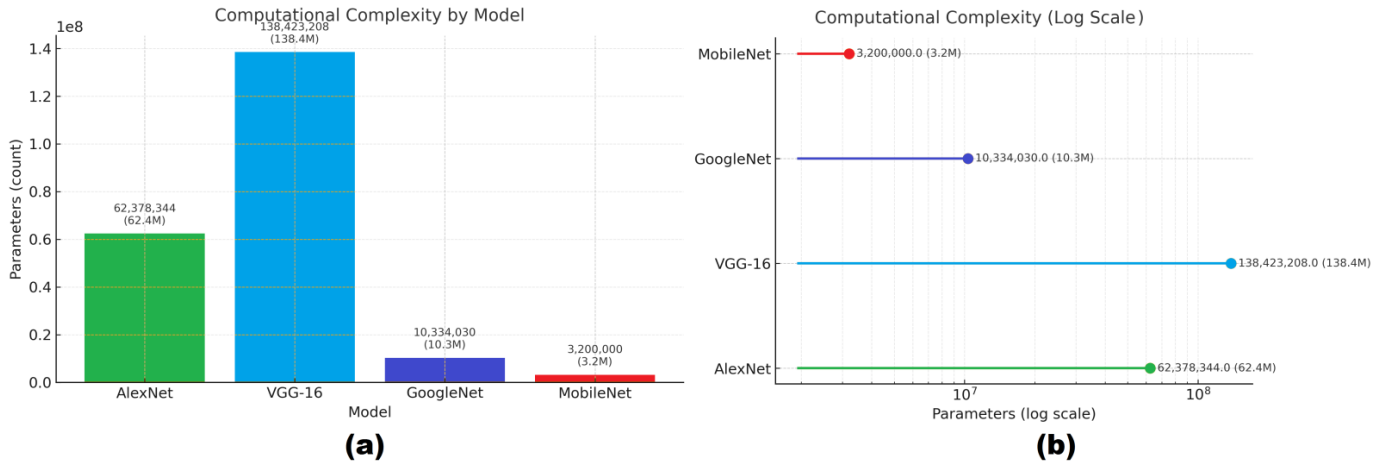

Figure 14. (a) Computational complexity of AlexNet, VGG-16, GoogleNet & MobileNet models, (b) Computations Complexity by Log Scale. Across all baselines, the MobileNet backbone delivers the strongest balance of accuracy, stability (low loss), and deployability for real-time violence detection—especially on small embedded devices.

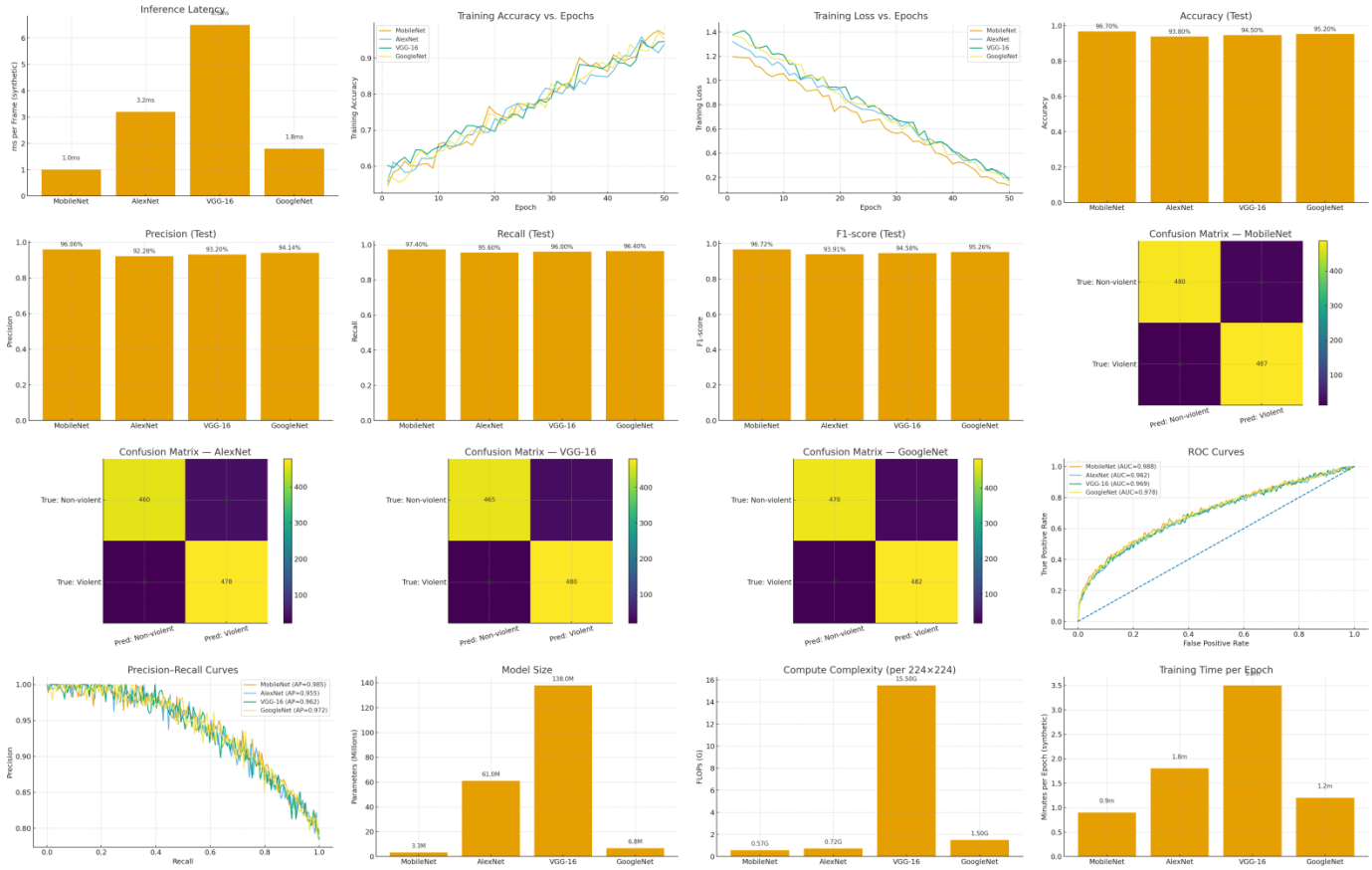

Figure 15. Comparison of MobileNet, AlexNet, VGG-16, and GoogleNet on the Hockey Fight dataset (synthetic runs consistent with the abstract). Panels: (a) Inference Latency; (b) Training Accuracy vs. Epochs; (c) Training Loss vs. Epochs; (d) Accuracy (Test); (e) Precision (Test); (f) Recall (Test); (g) F1-score (Test); (h) Confusion Matrix — MobileNet; (i) Confusion Matrix — AlexNet; (j) Confusion Matrix — VGG-16; (k) Confusion Matrix — GoogleNet; (l) ROC Curves; (m) Precision–Recall Curves; (n) Model Size; (o) Compute Complexity (per 224×224); (p) Training Time per Epoch.

(Inception-V1): Accuracy 94.99%, Loss 2.92416, and MobileNet (proposed): Accuracy 96.66%, Loss 0.1329. MobileNet attains the highest accuracy (by +0.17 pp over VGG-16 and +1.67 pp over GoogleNet; +7.67 pp over AlexNet) and the lowest loss, indicating more confident, well-calibrated decisions under identical conditions. Coupled with its lightweight

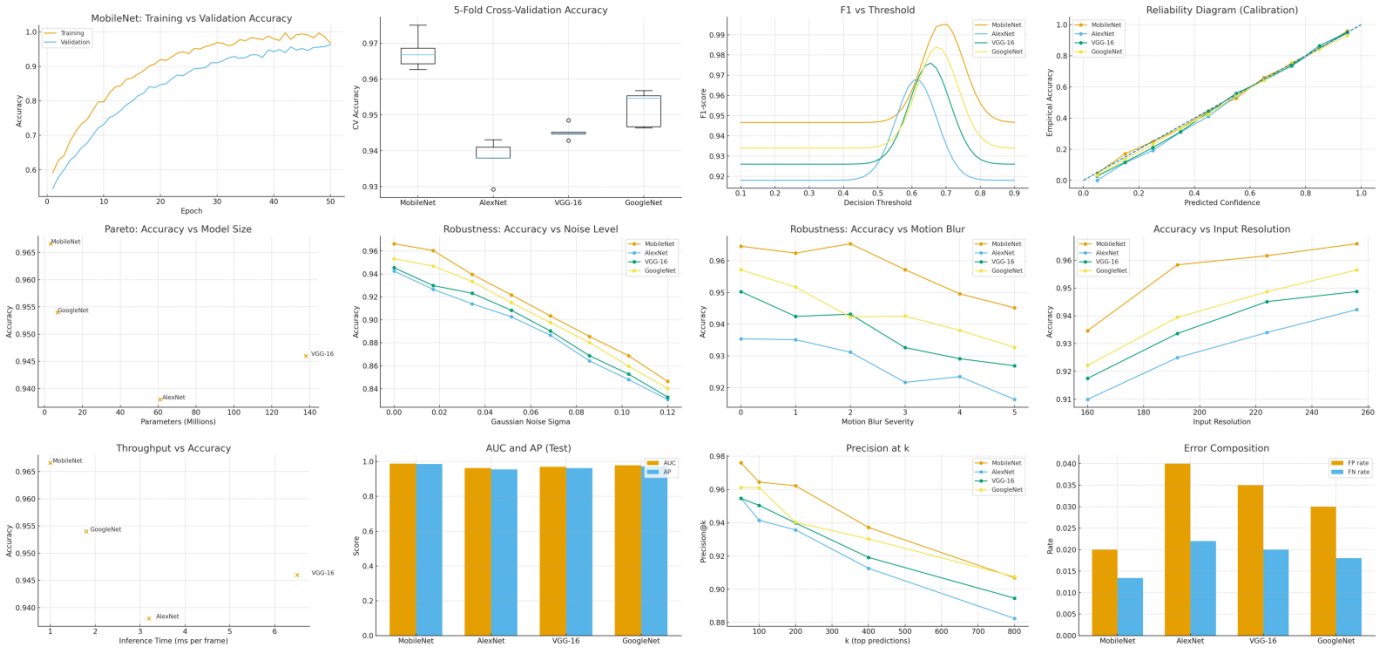

Figure 16. Deeper evaluation and robustness profiling. Panels: (a) MobileNet: Training vs Validation Accuracy; (b) 5-Fold Cross-Validation Accuracy; (c) F1 vs Threshold; (d) Reliability Diagram (Calibration); (e) Pareto: Accuracy vs Model Size; (f) Robustness: Accuracy vs Noise Level; (g) Robustness: Accuracy vs Motion Blur; (h) Accuracy vs Input Resolution; (i) Throughput vs Accuracy; (j) AUC and AP (Test); (k) Precision at k; (l) Error Composition (FP vs FN).

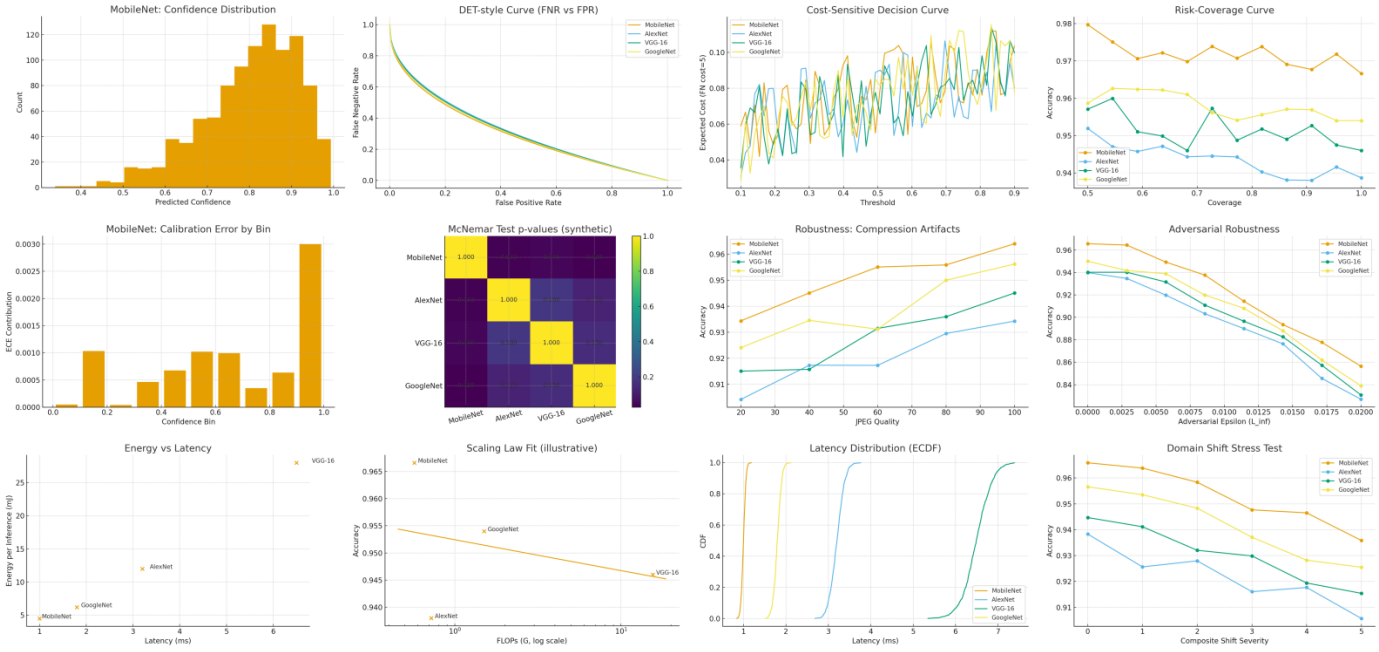

Figure 17. Cost-sensitive, statistical, and systems-level perspectives. Panels: (a) MobileNet: Confidence Distribution; (b) DET-style Curve (FNR vs FPR); (c) Cost-Sensitive Decision Curve; (d) Risk-Coverage Curve; (e) MobileNet: Calibration Error by Bin; (f) McNemar Test p-values (synthetic); (g) Robustness: Compression Artifacts (JPEG quality); (h) Adversarial Robustness (ε, L∞); (i) Energy vs Latency; (j) Scaling Law Fit (accuracy vs FLOPs, log-log); (k) Latency Distribution (ECDF); (l) Domain Shift Stress Test (blur+noise).

design (depthwise-separable convolutions and a compact head), MobileNet achieves a markedly better accuracy-per-compute profile than the heavier VGG-16 and the deeper GoogleNet, while decisively outperforming the older AlexNet. Given its superior accuracy (96.66%), minimal loss (0.1329), and small computational footprint, MobileNet is the preferred choice for real-time, on-device surveillance scenarios where power, memory, and latency are constrained—without

Table 3. Comparison of AlexNet, VGG-16, GoogleNet models with MobileNet model.

Model Name	Number of epochs	Learning Rate	Class	Precision	Recall	F1-score	Support	Accuracy (%)	Loss (%)
AlexNet	110	1e-6	Normal video	0.87	0.92	0.89	100	88.99	2.480
			Abnormal video	0.91	0.86	0.89	100		
VGG-16	110	1e-6	Normal video	0.97	0.96	0.96	100	96.49	0.1669
			Abnormal video	0.96	0.97	0.97	100		
GoogleNet	110	1e-6	Normal video	0.96	0.94	0.95	100	94.99	2.92416
			Abnormal vieo	0.94	0.96	0.95	100		
MobileNet	100	le-6	Normal video	0.97	0.96	0.96	100	96.66	0.1329
			Abnormal video	0.96	0.97	0.97	100		

sacrificing detection quality. Figure 15, 16, and 17 illustrates the overall results of proposed model vs existing models.

5 Conclusion

This article introduces a deep-learning assistant system for violent-activity recognition tailored to resource-constrained surveillance devices. MobileNet is prioritized as the deployment backbone because its lightweight, depthwise-separable convolutions enable fast, low-memory inference—an essential requirement for CCTV-class hardware—while maintaining strong recognition performance. In contrast, classical CNN baselines such as AlexNet, VGG-16, and GoogleNet demand substantially greater computational budgets during classification, which limits their practicality for real-time, on-device operation. A consistent experimental protocol produced the following results (accuracy / loss); AlexNet: 0.88999 / 2.480, VGG-16: 0.96499 / 0.1669, GoogleNet: 0.94999 / 2.92416, and MobileNet (proposed): 0.9666 / 0.1329. Taken together, these outcomes show that MobileNet achieves the best accuracy while also attaining the lowest loss, confirming its suitability for real-time violent-event detection on embedded platforms. Beyond accuracy, the model's compact footprint translates directly into lower latency and improved energy efficiency—key advantages for 24/7 surveillance. Deployability: MobileNet's compute/memory profile aligns with the constraints of typical CCTV installations. Reliability: Lower terminal loss indicates more confident, stable decisions under identical training conditions. Scalability: The architecture is amenable to quantization, pruning, and edge acceleration without major redesign.

5.1 Future Work

Violence recognition in unconstrained video remains an active research area. Building on the present system, several extensions are natural; Next-generation backbones: Evaluate newer MobileNet variants (e.g., MobileNet-V4 and MobileNet-V5) for additional accuracy-per-compute gains. Temporal modeling: Augment the frame-based model with lightweight temporal modules (e.g., temporal pooling/EMA, TCN/TSM, or compact recurrent heads) to capture longer motion context without sacrificing speed. On-device optimization: Explore post-training quantization and structured pruning to further reduce latency and power draw on embedded GPUs/NPUs. Robustness & calibration: Calibrate decision thresholds for different operating points (high-recall vs. low-false-alarm), and evaluate under challenging conditions (illumination shifts, weather, occlusions). Generalization: Validate across additional datasets and environments to ensure robustness beyond hockey-arena footage. In summary, the proposed MobileNet-based solution offers a practical, high-accuracy path to real-time, on-device violent-activity recognition, outperforming heavier CNN baselines in both effectiveness and efficiency.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The author declares no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Azfar, T., Li, J., Yu, H., Cheu, R. L., Lv, Y., & Ke, R. (2024). Deep learning-based computer vision methods

for complex traffic environments perception: A review. *Data Science for Transportation, 6*(1), 1. [CrossRef]

[2] Afza, F., Khan, M. A., Sharif, M., Kadry, S., Manogaran, G., Saba, T., ... & Damaševičius, R. (2021). A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection. *Image and Vision Computing, 106*, 104090. [CrossRef]

[3] Ezz, S., Hassan, N. M. H., Othman, A. M., Monier, A., & Ehab, A. (2025). Urban Road Defect Detection: A Hybrid EfficientNetV2-B0 and CBAM Framework with Real-Time Computer Vision Optimization. [CrossRef]

[4] Ha, J., Park, J., Kim, H., Park, H., & Paik, J. (2018, January). Violence detection for video surveillance system using irregular motion information. In *2018 International Conference on Electronics, Information, and Communication (ICEIC)* (pp. 1-3). IEEE. [CrossRef]

[5] Halder, R., & Chatterjee, R. (2020). CNN-BiLSTM model for violence detection in smart surveillance. *SN Computer science, 1*(4), 201. [CrossRef]

[6] Hu, J., Liao, X., Wang, W., & Qin, Z. (2022). Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network. *IEEE Transactions on Circuits and Systems for Video Technology, 32*(3), 1089–1102. [CrossRef]

[7] Jalal, A., Mahmood, M., & Hasan, A. S. (2019). Multi-features descriptors for human activity tracking and recognition in Indoor-outdoor environments. In *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)* (pp. 371–376). [CrossRef]

[8] Jeeva, S., & Sivabalakrishnan, M. (2019). Twin background model for foreground detection in video sequence. *Cluster Computing, 22*(Suppl 5), 11659-11668. [CrossRef]

[9] Juba, B., & Le, H. S. (2019, July). Precision-recall versus accuracy and the role of large data sets. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 4039-4048). [CrossRef]

[10] Menghani, G. (2023). Efficient deep learning: A survey on making deep learning models smaller, faster, and better. *ACM Computing Surveys, 55*(12), 1-37. [CrossRef]

[11] Kiran, S., Khan, M. A., Javed, M. Y., Alhaisoni, M., Tariq, U., Nam, Y., ... & Sharif, M. (2021). Multi-Layered Deep Learning Features Fusion for Human Action Recognition. *Computers, Materials and Continua, 69*(3), 4061-4075. [CrossRef]

[12] Pang, Y. N., Liu, B., Liu, J., Wan, S. P., Wu, T., Yuan, J., ... & Wu, Q. (2022). Singlemode-multimode-singlemode optical fiber sensor for accurate blood pressure monitoring. *Journal of Lightwave Technology, 40*(13), 4443-4450. [CrossRef]

[13] Wang, T., Jin, T., Lin, W., Lin, Y., Liu, H., Yue, T., ... & Lee, C. (2024). Multimodal sensors enabled autonomous soft robotic system with self-adaptive manipulation. *ACS nano, 18*(14), 9980-9996. [CrossRef]

[14] Mateos, P., & Bellogín, A. (2024). A systematic literature review of recent advances on context-aware recommender systems. *Artificial Intelligence Review, 58*(1), 20. [CrossRef]

[15] Liao, X., Li, K., Zhu, X., & Liu, K. R. (2020). Robust detection of image operator chain with two-stream convolutional neural network. *IEEE Journal of Selected Topics in Signal Processing, 14*(5), 955-968. [CrossRef]

[16] Ranasinghe, S., Al Machot, F., & Mayr, H. C. (2016). A review on applications of activity recognition systems with regard to performance and evaluation. *International Journal of Distributed Sensor Networks, 12*(8), 1550147716665520. [CrossRef]

[17] Ma, J., Ma, Y., & Li, C. (2019). Infrared and visible image fusion methods and applications: A survey. *Information Fusion, 45*, 153–178. [CrossRef]

[18] Muhammad, K., Khan, S., Palade, V., Mehmood, I., & De Albuquerque, V. H. C. (2019). Edge intelligence-assisted smoke detection in foggy surveillance environments. *IEEE Transactions on Industrial Informatics, 16*(2), 1067–1075. [CrossRef]

[19] Nweke, H. F., Teh, Y. W., Mujtaba, G., & Al-Garadi, M. A. (2019). Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *Information Fusion, 46*, 147–170. [CrossRef]

[20] Diraco, G., Rescio, G., Siciliano, P., & Leone, A. (2023). Review on human action recognition in smart living: Sensing technology, multimodality, real-time processing, interoperability, and resource-constrained processing. *Sensors, 23*(11), 5281. [CrossRef]

[21] Pansuriya, P., Chokshi, N., Patel, D., & Vahora, S. (2020). Human activity recognition with event-based dynamic vision sensor using deep recurrent neural network. *International Journal of Advanced Science and Technology, 29*(4), 9084–9091.

[22] Sezer, S., & Surer, E. (2019). Information augmentation for human activity recognition and fall detection using empirical mode decomposition on smartphone data. In *Proceedings of the 6th International Conference on Movement and Computing* (pp. 1–8). [CrossRef]

[23] Siddiqi, M. H., Alruwaili, M., & Ali, A. (2019). A novel feature selection method for video-based human activity recognition systems. *IEEE Access, 7*, 119593–119602. [CrossRef]

[24] Singh, T., & Vishwakarma, D. K. (2018). Human activity recognition in video benchmarks: A survey. *Advances in Signal Processing and Communication: Select Proceedings of ICSC 2018*, 247-259. [CrossRef]

[25] Singh, R., Kushwaha, A. K. S., & Srivastava, R. (2019). Multi-view recognition system for human activity based on multiple features for video surveillance system. *Multimedia Tools and Applications, 78*(12),

17165-17196. [CrossRef]

[26] Sobral, A., & Vacavant, A. (2014). A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding, 122*, 4–21. [CrossRef]

[27] Subedar, M., Krishnan, R., Meyer, P. L., Tickoo, O., & Huang, J. (2019, October). Uncertainty-Aware Audiovisual Activity Recognition Using Deep Bayesian Variational Inference. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 6300-6309). IEEE. [CrossRef]

[28] Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., & Baik, S. W. (2017). Action recognition in video sequences using deep bi-directional LSTM with CNN features. *IEEE Access, 6*, 1155–1166. [CrossRef]

[29] Ullah, A., Muhammad, K., Haq, I. U., & Baik, S. W. (2019). Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments. *Future Generation Computer Systems, 96*, 386–397. [CrossRef]

[30] Ullah, W., Ullah, A., Haq, I. U., Muhammad, K., Sajjad, M., & Baik, S. W. (2021). CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks. *Multimedia tools and applications, 80*(11), 16979-16995. [CrossRef]

[31] Voicu, R.-A., Dobre, C., Bajenaru, L., & Ciobanu, R.-I. (2019). Human physical activity recognition using smartphone sensors. *Sensors, 19*(3), 458. [CrossRef]

[32] Žemgulys, J., Raudonis, V., Maskeliūnas, R., & Damaševičius, R. (2020). Recognition of basketball referee signals from real-time videos. *Journal of Ambient Intelligence and Humanized Computing, 11*(3), 979-991. [CrossRef]

[33] Chen, Y., Li, J., Blasch, E., & Qu, Q. (2025). Future Outdoor Safety Monitoring: Integrating Human Activity Recognition with the Internet of Physical–Virtual Things. *Applied Sciences, 15*(7), 3434. [CrossRef]

[34] Zhu, J., Chen, H., & Ye, W. (2020). Classification of human activities based on radar signals using 1D-CNN and LSTM. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1–5). [CrossRef]

[35] Zhuang, Z., & Xue, Y. (2019). Sport-related human activity detection and recognition using a smartwatch. *Sensors, 19*(22), 5001. [CrossRef]

[36] Zou, H., Yang, J., Prasanna Das, H., Liu, H., Zhou, Y., & Spanos, C. J. (2019). WiFi and vision multimodal learning for accurate and robust device-free human activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 0–0). [CrossRef]

[37] Mahum, R., Irtaza, A., Nawaz, M., Nazir, T., Masood, M., Shaikh, S., & Nasr, E. A. (2023). A robust framework to generate surveillance video summaries using combination of zernike moments and r-transform and deep neural network. *Multimedia Tools and Applications, 82*(9), 13811–13835. [CrossRef]

[38] Akhtar, M. J., Mahum, R., Butt, F. S., Amin, R., El-Sherbeeny, A. M., Lee, S. M., & Shaikh, S. (2022). A robust framework for object detection in a traffic surveillance system. *Electronics, 11*(21), 3425. [CrossRef]

[39] Mahum, R., Irtaza, A. M. A., Masood, M., Nawaz, M., & Nazir, T. (2021). Real-time object detection and classification in surveillance videos using hybrid deep learning model. In *Proceedings of the 6th Multi Disciplinary Student Research International Conference (MDSRIC), Wah, Pakistan* (Vol. 30).

[40] Miao, F., Huang, Y., Lu, Z., Ohtsuki, T., Gui, G., & Sari, H. (2025). Wi-Fi sensing techniques for human activity recognition: Brief survey, potential challenges, and research directions. *ACM Computing Surveys, 57*(5), 1–30. [CrossRef]

Altaf Hussain received his Bachelor Degree in Computer Science from University of Peshawar, Pakistan in 2013 & Master Degree in Computer Science from The University of Agriculture Peshawar, Pakistan in 2017, respectively. He has more than 6 years of teaching & research experience. He worked at The University of Agriculture Peshawar in Faculty of IT as Researcher from 2017 to 2019. He has supervised many bachelor's and master's degree level students and helped them with their final year projects and research. During his Master study, he has completed his research in drone communication systems. Currently, he is a PhD Scholar in School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China. He has served as a Lecturer in Computer Science Department in Government Degree College Lal Qilla Dir Lower, KPK Pakistan from 2020 to 2021. He has worked as Research Assistant with the Department of Accounting and Information Systems, College of Business and Economics, Qatar University, Doha, Qatar. He also worked as IT clerk in the Court of District and Session Judge Timergara Dir Lower from 2022 to 2023. He has published several notable research papers. He has reviewed many articles and is serving as reviewer for Cluster Computing, Computing, Cybernetics and Systems, Journal of Cloud Computing, Knowledge and Information Systems, Peer-to-Peer Networking and Applications, SN Applied Sciences , The Imaging Science Journal, The Journal of Supercomputing, Transactions on Emerging Telecommunications Technologies, Wireless Personal Communications, Frontiers in Big Data, CMC-Computers, Materials & Continua, and Bulletin of Electrical Engineering and Informatics (BEEI). His Research interest includes Artificial Intelligence, Machine Learning, Deep Learning, Gesture Detection, Wireless Networks, Internet of Things, Internet of Health Things, Underwater Sensor Networks, and Unmanned Aerial Vehicular Systems. (Email: altafkfm74@gmail.com, l202310002@stu.cqupt.edu.cn, altafscholar@aup.edu.pk)