EVOD-RoI: Adaptive Edge-Assisted Video Object Detection System Based on RoI Transmission and Processing

Xuejian Chi*
Shandong University
Qingdao, China
chixuejian@mail.sdu.edu.cn

Bowen Han*
Tongji University
Shanghai, China
bown@tongji.edu.cn

Yifei Zou†
Shandong University
Qingdao, China
yfzou@sdu.edu.cn

Yong Zhang
Chinese Academy of Sciences
Shenzhen, China
zhangyong@siat.ac.cn

Falko Dressler
Technische Universität Berlin
Berlin, Germany
dressler@ccs-labs.org

Dongxiao Yu
Shandong University
Jinan, China
dxyu@sdu.edu.cn

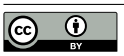
Abstract

With the continuous convergence of edge intelligence and multimedia applications, edge-assisted video object detection has attracted increasing attention. To enhance system quality of service (QoS), existing studies primarily focus on full-frame scheduling and resource allocation, often overlooking the dominant role of the region of interest (RoI) in detection accuracy and lacking robust detection models for regionally blurred scenarios. To address these challenges, this paper proposes *EVOD-RoI*, an edge-assisted video object detection system based on RoI transmission and processing. It adaptively adjusts detection location, non-RoI resolution, and model selection to reduce non-RoI resolution during edge upload, minimizing overall latency. Meanwhile, it improves accuracy by fine-tuning a model specifically for regionally blurred frames. Specifically, *EVOD-RoI* introduces three key innovations: (1) a spatiotemporal joint sliding window algorithm to determine the optimal RoI, which saves bandwidth during transmission stage; (2) regionally blurred feature-aware model fine-tuning to improve detection accuracy in processing stage; and (3) a deep reinforcement learning–based approach to dynamically and adaptively optimize transmission and resource scheduling under heterogeneous environments. Experiments on a real-world testbed demonstrate that, compared with SOTA methods, *EVOD-RoI* improves detection accuracy by 15.9%–30.1% while reducing overall system latency by 16.8%–41.8%, significantly enhancing QoS in resource-constrained edge scenarios.

CCS Concepts

• **Information systems** → **Multimedia streaming**; • **Computer systems organization** → *Distributed architectures*; • **Computing methodologies** → *Object detection*.

*Xuejian Chi and Bowen Han contributed equally to this work.
†Corresponding author.



MM '26, Rio de Janeiro, Brazil

ACM ISBN 979-8-4007-2213-4/2026/11
https://doi.org/10.1145/3767308.3835043

Keywords

Edge intelligence, video transmission and processing, region of interest, quality of service

ACM Reference Format:
Xuejian Chi, Bowen Han, Yifei Zou, Yong Zhang, Falko Dressler, and Dongxiao Yu. 2026. EVOD-RoI: Adaptive Edge-Assisted Video Object Detection System Based on RoI Transmission and Processing. In *Proceedings of the 34th ACM International Conference on Multimedia (MM '26), November 10–14, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3767308.3835043

1 Introduction

Edge-assisted video object detection involves resource-constrained terminal devices capturing video streams and offloading the computationally intensive detection tasks to nearby edge servers. It outperforms on-device methods by leveraging the ample computational and storage resources of edge servers [6–8], which allows for the deployment of large, high-precision models (e.g., YOLOv5l model with 46 million parameters [12]). Meanwhile, compared to remote cloud centers, the physical proximity of edge servers significantly reduces data transmission latency and mitigates privacy risks. Therefore, edge-assisted video object detection is increasingly playing a vital role in latency-sensitive scenarios such as intelligent transportation [26], industrial monitoring and inspection [3].

Owing to its significance, a series of studies has been conducted in recent years. Some early research predominantly treats the entire video frame as the basic unit for transmission and processing, overlooking the critical impact of intra-frame Regions of Interest (RoI[1]) on overall detection accuracy. For instance, [16, 18] reduce transmission latency by adjusting the resolution of the whole frame before uploading it to the edge server, and [1, 24] improve detection performance by transmitting only key frames. However, high-value information (e.g., dense objects and significant motion) is typically confined to specific local areas (RoIs) rather than spanning the entire frame, and these regions predominantly determine the detection accuracy, as further validated in Observation 1 of Section 2. Consequently, under limited bandwidth, a more efficient mechanism has emerged: preserving high resolution only within the RoIs

[1]RoI refers to spatial sub-regions containing high-value information (e.g., objects) critical for the target task.

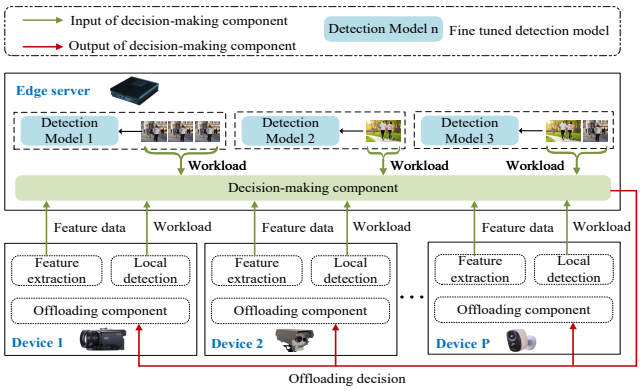


Figure 1: The component architecture of *EVOD-RoI*.

while applying lower resolution to the remaining areas to transmit more valuable information.

With the similar observations on RoI, recent studies have tried to adopt this RoI-based transmission mechanism to reduce bandwidth consumption, but still face two major gaps in practical deployment. Firstly, in the video transmission stage, the calibration of RoIs often lacks spatiotemporal adaptability. Existing methods typically rely on static spatial priors [2] or single-dimensional motion cues [21], struggling to dynamically and accurately identify optimal RoIs in videos with complex spatiotemporal changes. Secondly, in the subsequent video processing stage, the edge-deployed object detection models often fail to adapt to video frames with partially blurred regions. For example, [30] preserves original resolution for areas with dense small objects while compressing other regions. However, its detection model is still the original YOLOv5, trained on fully clear images. As shown in Observation 2 of Section 2, when applied to frames with non-RoI blur, the original YOLOv5 model experiences a decrease in detection accuracy by 7% to 10%. Therefore, achieving spatiotemporally adaptive RoI extraction alongside fine-tuning the detection model to handle regionally blurred video frames is indispensable for the system's overall accuracy and robustness.

To address the above issues, we propose *EVOD-RoI*, an **E**dge-assisted **V**ideo **O**bject **D**etection system driven by **RoI**, which jointly optimizes video transmission and processing. As shown in Fig. 1, *EVOD-RoI* consists of multiple terminal devices equipped with lightweight detection models and an edge server hosting larger and more accurate models. After capturing video frames, each terminal extracts frame-level features, such as frame difference and RoI statistics, and sends them to the edge server. Based on these features and the workloads of the detection models, the edge server determines the offloading and processing strategy for each frame. The main components of *EVOD-RoI* are described as follows.

Component I: End-side Feature Extraction. This component aims to accurately localize the RoI of each video frame and extract key features such as inter-frame differences. As illustrated in Observation 1 of Section 2, RoIs vary significantly in spatial location and scale across different video types. Traditional methods, such as inter-frame difference-based motion detection [21] or prior knowledge-based region identification [2], cannot jointly capture temporal and spatial information, and thus fail to effectively determine the optimal RoI[2] for each frame. To address this issue, we introduce a spatiotemporal joint sliding window algorithm. It first uses a temporal window to analyze motion intensity across frames and identify regions with significant motion, and then applies a spatial window to evaluate object density and locate densely populated areas. By integrating temporal and spatial information, the algorithm adaptively determines the optimal RoI for diverse video content.

Component II: Decision-making with Global View. This module aims to determine the optimal joint strategy for RoI-based transmission and processing. Traditional methods typically treat complete video frames as the decision unit, whereas fine-grained RoI transmission can save bandwidth more effectively. As shown in Observation 3 of Section 2, offloading more frames can improve detection accuracy, but limited bandwidth may require reducing the transmission resolution. Moreover, as noted in Section 5, previous studies mainly optimize either video transmission or processing, while overlooking their coupling in edge-assisted video analytics. To achieve joint optimization, we design a tailored Advantage Actor-Critic (A2C) framework with a composite action space that jointly determines offloading decisions, resolution levels, and model selection, together with a latency-accuracy-aware reward function. This mechanism captures the interdependencies among decision variables, dynamically balances computational loads across edge and end devices, and learns globally optimized strategies.

Component III: Edge-side Detection Models. This component performs high-precision detection on video frames using RoI features. Off-the-shelf detection models (e.g., original YOLO) are typically trained on uniformly resolved images and thus lack the ability to prioritize high-value information within RoIs. As indicated in Observation 2 of Section 2, directly applying these general-purpose models to regionally blurred frames limits further accuracy improvement. To address this bottleneck, *EVOD-RoI* employs a targeted fine-tuning strategy. By constructing a dataset with regionally blurred characteristics, the model can reliably extract features from sharp RoIs while remaining robust to degraded backgrounds. This strategy adapts the model to RoI-based transmission and enhances its feature extraction capability under spatially varying resolutions.

In *EVOD-RoI*, when the terminal device receives the decision made by the edge server, it immediately performs frame offloading. Frames that do not need to be uploaded will be processed by the local detection component, while frames that require uploading will have their non-ROI resolution reduced according to the strategy and transmitted to the edge for detection. Finally, the device integrates the local and edge detection results and output the final decision.

The key contribution of our work is the design and implementation of *EVOD-RoI*, a holistic framework that systematically co-designs video transmission, model optimization, and model selection for edge-assisted object detection. First, we develop a novel spatiotemporal sliding window algorithm that integrates temporal motion intensity and spatial object density. This method overcomes the limitations of traditional static priors, enabling the adaptive identification of optimal RoIs across diverse video transmission scenarios. Crucially, to maximize the utility of transmitted information, we develop a targeted model fine-tuning methodology that enables

[2]Optimal RoI refers to the region maximizing the aggregate score of motion intensity and object density, representing the richest semantic information.

the edge model to effectively adapt to complex environments, such as regionally blurred frames, thereby enhancing feature extraction efficiency and overall detection accuracy during video processing stage. Finally, the entire system is orchestrated by an A2C agent, which learns the complex coupling dynamics between the video transmission and processing stages to achieve their joint optimization. Extensive experiments validate the superiority of our design, demonstrating that *EVOD-RoI* system improves detection accuracy by 15.9%–30.1% and reduces latency by 16.8%–41.8% compared to existing SOTA methods.

2 Motivation Study

To provide sufficient support for our efficient RoI-based edge-assisted video object detection system, we conducted motivation experiments that yielded three key observations (obs. for short). These experiments used a pre-trained YOLOv5 model [12] on two video types: highway vehicle footage (Video 1 [33]) and urban pedestrian videos (Video 2 [31]). We evaluated system performance using a weighted metric of detection accuracy and latency (detailed in Section 3). To ensure accuracy, the RoI in each video frame was manually labeled in motivation part.

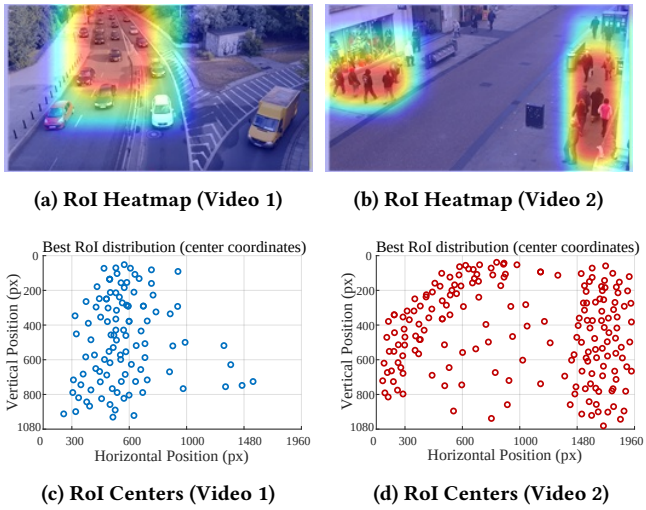
2.1 Obs. 1: Heterogeneous RoI Distribution

First, in many cases, the RoI occupies only a small portion of a video frame but contains the vast majority of valid objects. As intuitively revealed by the heatmaps in Fig. 2(a) and (b), high-value information is highly concentrated in specific local areas (i.e., "heat centers") rather than spanning the entire frame. Furthermore, the spatial distribution of RoIs is highly heterogeneous across different scenarios. As illustrated by the scatter plots in Fig. 2(c) and (d), which track the RoI centers sampled at 2-second intervals, the spatial locations and clustering patterns of optimal RoIs vary significantly between the highway and urban scenarios. Finally, even within the same scenario, the RoI exhibits significant dynamic fluctuation over time. As depicted by the RoI variation curves over a 4-minute duration in Fig. 3, the RoI size experiences drastic shifts across different time segments. For instance, the RoI ratio in Video 1 drops drastically from 73.2% at 18 seconds to merely 11.4% at 156 seconds. Despite these severe spatiotemporal variations, the consistently high object coverage (>86%) confirms that the dynamic RoI effectively captures the vast majority of valid information.

This observation indicates that target objects are highly concentrated within the RoI, which dominating the overall detection accuracy. Moreover, its spatiotemporal variations across scenarios necessitate an efficient and adaptive RoI extraction mechanism.

2.2 Obs. 2: Necessity of Fine-tuning

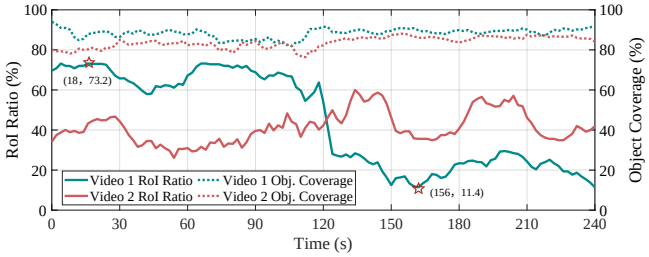
Standard pre-trained models exhibit poor detection performance on frames with mixed resolutions. At identical transmission sizes, preserving high resolution within the RoI while compressing only the background should retain more critical features than compressing the entire frame. However, the empirical results in Table 1 contradict this expectation. As shown in the table, when the transmission size is constrained to the same level, the RoI-based hybrid compression (−*hybrid*.) strategy consistently underperforms the uniform compression (−*uniform*.) strategy. For instance, for Video 1 at a size of 1.96 MB, Video 1(−*hybrid*.) achieves an accuracy of 85.3%, which is lower than the 87.1% achieved by Video 1(−*uniform*.). Similarly, for Video 2 at 2.10 MB, the Video 2(−*hybrid*.) lags behind



(a) RoI Heatmap (Video 1) (b) RoI Heatmap (Video 2)

(c) RoI Centers (Video 1) (d) RoI Centers (Video 2)

Figure 2: Statistics of spatiotemporal RoI heterogeneity.



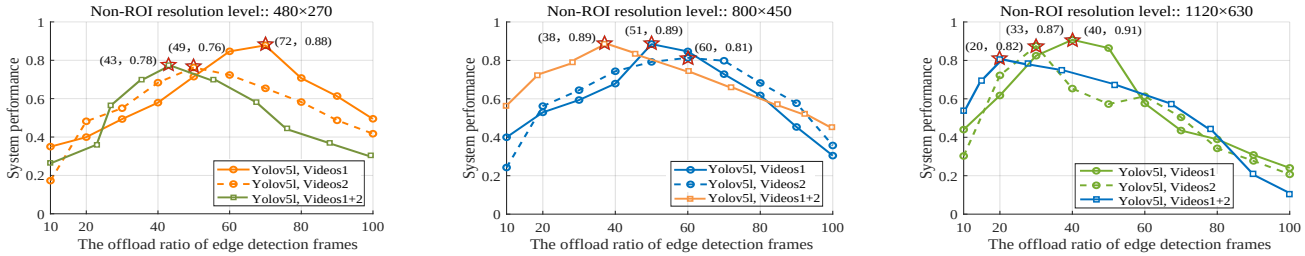
Notes:"RoI Ratio" denotes the spatial proportion of the extracted Region of Interest relative to the entire video frame area. "Object Coverage" is defined as the ratio of the number of objects within the RoI to the total number of objects in the entire frame .

Figure 3: RoI and object distribution statistics.

Table 1: Impact of non-RoI Compression Ratio.

Video Compression Ratio	Accuracy	Missed/False	Size
Video 1 (0%)	92.4%	2.4%	2.35 MB
Video 2 (0%)	90.6%	3.2%	2.56 MB
Video 1-uniform. (16.6%)	87.1%	4.4%	1.96 MB
Video 1-hybrid. (30.0%)	**85.3%**	**6.8%**	**1.96 MB**
Video 2-uniform. (18.0%)	83.6%	7.2%	2.10 MB
Video 2-hybrid. (30.0%)	**81.2%**	**7.9%**	**2.10 MB**
Video 1-uniform. (36.2%)	76.9%	8.1%	1.58 MB
Video 1-hybrid. (60.0%)	**75.8%**	**8.6%**	**1.58 MB**
Video 2-uniform. (33.5%)	73.8%	8.9%	1.64 MB
Video 2-hybrid. (60.0%)	**73.2%**	**9.5%**	**1.64 MB**

Notes:"Video x-uniform." refers to uniform entire frame compression. "Video x-hybrid." denotes RoI-based hybrid compression, which preserves the RoI at original resolution while heavily compressing the non-RoI. "Accuracy" denotes the YOLOv5 model's detection accuracy; "Missed/False" is the average rate of missed detections (FN) and false alarms (FP).



(a) Non-RoI resolution level: 480×270 **(b) Non-RoI resolution level: 800×450** **(c) Non-RoI resolution level: 1120×630**

Figure 4: The detection results for different videos using different non-RoI resolution levels.

the Video 2($-uniform.$) by 2.4% and suffers a higher Missed/False error rate (7.9% vs. 7.2%). After increasing the compression ratio, there are still similar situations.

This observation indicates that the pre-trained model is highly sensitive to distribution shifts caused by mixed frames with clear RoI and blurred non-RoI. The abrupt quality transition disrupts the model's ability to integrate global context, causing a more severe performance penalty than uniform degradation. These findings underscore the necessity of training a detection model specifically adapted to regional blur.

2.3 Obs 3: Complexity of Decision Making

The key decision variables, including video type, model selection, offload ratio, and non-RoI resolution level, are highly nonlinear and strongly coupled, making joint optimization essential for maximizing system performance. Optimizing any variable independently often leads to a globally suboptimal strategy. Fig. 4 shows system performance, measured as the weighted sum of detection accuracy and delay, under different edge offload ratios and non-RoI resolution settings. As more frames are processed at the edge, system performance first increases and then decreases, exhibiting a clear nonlinear trend. Moreover, the optimal offload ratio changes with the resolution level, indicating a strong interaction between non-RoI resolution and edge processing frequency. Therefore, a strategy that is optimal at one resolution may degrade performance at another.

To further examine these interdependencies, we analyzed the optimal strategies under different detection models (YOLOv5s, YOLOv5l, and YOLOv5x) and video types (Video 1 and Video 2), as detailed in Table 1 of Appendix A. The results reveal two key interactions. (1) Coupling with model and video content: the optimal offloading strategy is highly sensitive to both the computational capacity of the selected edge model and the spatiotemporal characteristics of the video. Even at the same resolution, changes in model complexity or video scenario require different offload ratios to maximize system utility. (2) Coupling with system load: resource contention substantially changes the optimal policy. Moving from a single-device environment to a multi-device shared setting requires a much lower frame offloading ratio to alleviate network congestion and queuing delays at the edge server.

This observation indicates that the decision variables in the system are not independent, but exhibit strong coupling and mutual constraints, which makes it infeasible to obtain the optimal strategy through single-variable adjustment alone, and instead requires joint consideration of the synergistic effects among multiple variables.

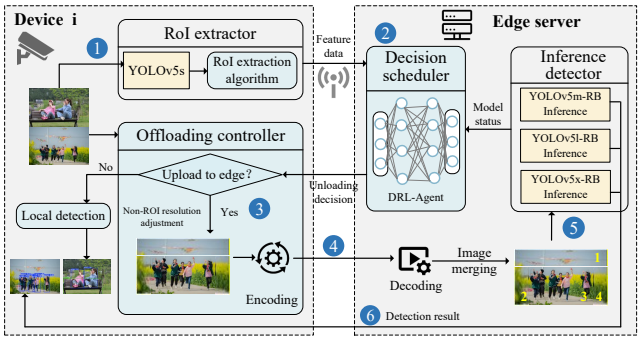


Figure 5: The detailed implementation of *EVOD-RoI*.

3 System Design of our EVOD-RoI

In this section, we present *EVOD-RoI*, an **E**dge-assisted **V**ideo **O**bject **D**etection system based on **RoI**-aware transmission and processing. It consists of multiple terminal devices and an edge server. Each terminal device is equipped with a lightweight detection model, while the edge server hosts n more powerful models. For each terminal devices, captured video frames can be processed locally or offloaded to the edge. Local processing avoids transmission latency but yields lower accuracy, whereas edge offloading leverages larger model capacity for higher accuracy at the cost of transmission delay. To maximize system's quality of service (QoS), *EVOD-RoI* dynamically adjusts frame detection location based on bandwidth and resource conditions, and determines strategies like the non-RoI resolution level for uploaded frames.

Fig. 5 shows an overview of *EVOD-RoI*. ① The **RoI extractor** annotates RoIs in captured frames and extracts frame-level features (e.g., frame difference, RoI ratio), which are uploaded to the edge server to determine detection strategies. ② The edge server's **decision scheduler** inputs these features and the model's queueing status into a DRL agent to adaptively determine the detection location, non-RoI resolution level, and model selection for each frame. ③ The **offloading controller** receives these decisions, assigns local processing frames to the terminal, and compresses uploaded frames based on the non-RoI resolution, preserving only the RoI's original resolution. ④ After encoding, these frames are transmitted to the edge. ⑤ The edge server decodes and merges

the RoI and non-RoI regions, feeding the reconstructed frame into the **inference detector** using the selected model (e.g., YOLOv5x-RB) for object detection. ⑥ Finally, edge results are sent back and integrated with local results to complete detection for the entire video segment.

3.1 RoI Extractor

We use window $W_i = (x_i, y_i, w, h)$ to represent a single RoI candidate region, where (x_i, y_i) denotes the top-left coordinate of the i-th RoI, and w, h represent its width and height, respectively. The core idea of RoI determination is that regions containing more objects and exhibiting higher motion speed are more likely to be informative. Therefore, we define the number of objects within W_i as: $C_i = \sum_{t=t_0}^{t_0+\tau-1} Count_{det}(W_i, t)$.

Here, τ denotes the length of the temporal window (e.g., 5 consecutive frames, $\tau = 5$), and $Count_{det}(W_i, t)$ represents the number of detected objects within the candidate window W_i in the t-th frame, obtained using a lightweight detection model running on the terminal device.

The motion speed is defined as:

$$V_i = \frac{1}{\tau - 1} \sum_{t=t_0}^{t_0+\tau-1} \left[\frac{1}{wh} \sum_{(x,y)\in W_i} |I_t(x, y) - I_{t-1}(x, y)| \right], \quad (1)$$

$I_t(x, y)$ denotes the grayscale value of pixel (x, y) in the t-th frame.

Finally, we obtain the score function $Score_i = \alpha C_i + \beta V_i$ for each candidate region. If $Score_i > \theta$, the window W_i is marked as a candidate RoI, where α and β are weighting factors, and θ is a constant threshold determined empirically, which may vary across different video types.

We design a spatiotemporal joint sliding window algorithm (the detailed pseudo-code is provided in Algorithm 1 of Appendix B) to obtain the RoI. A sliding window mechanism is applied to partition the video frames in both temporal and spatial dimensions, and each candidate window W_i is scored according to Eq. (1) (Lines 1-12 of Algorithm 1). After computing the scores for all candidate windows, those with scores exceeding the threshold θ are spatially accumulated onto a two-dimensional heatmap of the same size as the original frame (Lines 13-16). The heatmap is then normalized and thresholded to generate a binary mask (Lines 17-18). Connected component analysis is performed on the mask to identify high-response contiguous regions, from which the minimum bounding rectangles are extracted as the final set of RoI candidate boxes (Lines 19-25): $\mathcal{R}_k = \{(x_j, y_j, w_j, h_j) \mid j = 1, \ldots, N_k\}$, where N_k denotes the number of RoIs extracted from k-th frame. The final RoIs are adaptive in both position and size based on the video content.

3.2 Decision Scheduler

3.2.1 Definition and Formulation of Optimization Problem. To optimize the transmission and scheduling strategy, it is necessary to construct a performance evaluation metric and accordingly formalize the optimization problem. We consider two metrics: latency and accuracy, which are formally defined as follows.

System Latency. The latency of a single video frame consists of two parts: transmission latency and inference latency. We introduce a binary decision variable, $X_k \in \{0, 1\}$ for the k-th frame, where $X_k = 0$ represents local processing and $X_k = 1$ signifies offloading to the edge. When the k-th frame is processed locally (i.e., $X_k = 0$), there is no transmission latency, so the latency includes only the inference latency L_k^{loc}. When the frame is offloaded to the edge for processing (i.e., $X_k = 1$), the latency comprises both transmission latency D_k and inference latency L_k^{edg}, and D_k is defined as follows: $D_k = \frac{E_{RoI} + \rho_k (E_{total} - E_{RoI})}{B_{net}}$, where E_{RoI} denotes the number of pixels in the RoI, E_{total} denotes the total number of pixels in the entire frame. ρ_k represents the resolution level of the non-RoI, defined as the pixel ratio after downsampling relative to the original resolution. A smaller ρ_k means a lower resolution in the background, leading to lower data volume and reduced transmission latency. Therefore, the latency of k-th frame be expressed as follows: $\mathcal{L}_k = (1 - X_k) L_k^{loc} + X_k \left(D_k + L_k^{edg} \right)$.

Detection Accuracy. We use Intersection over Union (IoU) [27] as the evaluation metric for object detection accuracy in the k-th video frame. It is important to note that a single frame typically contains multiple objects. Therefore, the detection accuracy $\mathcal{A}_k$ of the k-th frame is defined as the average IoU of all objects within that frame: $\mathcal{A}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} IoU_i$, where N_k denotes the total number of ground truth objects in the k-th frame, and IoU_i denotes the IoU value corresponding to the i-th object.

The objective of *EVOD-RoI* is to achieve real-time and accurate video object detection by optimizing the trade-off between latency and accuracy. The optimization problem can be formulated as:

$$\max_{X_k, \rho_k, m_k} \quad \sum_{k=1}^{K} \lambda \mathcal{A}_k - (1 - \lambda) \mathcal{L}_k \quad (2)$$

$$s.t. \quad X_k \in \{0, 1\}, \ \forall k \in [1, \ldots, K], \quad (3)$$

$$m_k \in \{1, 2, \ldots, n\}, \quad \forall k \in [1, \ldots, K], \quad (4)$$

$$0 < \rho_k \leq 1, \quad \forall k \in [1, \ldots, K], \quad (5)$$

$$\mathcal{L}_k \leq L_{max}, \ \forall k \in [1, \ldots, K], \quad (6)$$

$$\mathcal{A}_k \geq A_{min}, \ \forall k \in [1, \ldots, K], \quad (7)$$

where λ denotes the weighting factor between latency and accuracy, K denotes total frame count and m_k denotes number of edge detection models. Constraint (3) ensures each frame is processed either locally or at the edge, enforcing exclusive assignment. Constraint (4) selects from n fine-tuned YOLO variants with different parameter scales for edge processing. Constraint (5) defines non-RoI resolution levels, while constraints (6) and (7) specify the maximum latency (L_{max}) and minimum accuracy (A_{min}) requirements, respectively.

As shown in Observation 3 of Section 2, multiple decision variables (e.g., detection location, non-RoI resolution, and model selection) exhibit strong coupling and nonlinear relationships with system performance (accuracy and latency). Lacking explicit functional expressions among these variables, traditional modeling methods like linear programming are ineffective in finding optimal configurations. To address this, we introduce a learning-based policy optimization approach. By leveraging deep reinforcement learning, the system learns optimal strategies from complex, dynamic contextual information, enabling adaptive optimization of transmission and scheduling decisions.

3.2.2 Deep Reinforcement Learning Agent. The above problem is formulated as a Markov Decision Process (MDP) [15, 23], defined by the four-tuple $< S, A, P, R >$, where S, A, P, and R represents the

state space, the action space, the transition probability function, and the reward function, respectively.

State. The state space includes frame-level features (frame difference Δ_k and RoI ratio E_k), transmission bandwidth B_{net}, and edge model task queue lengths $Q_k = \{q_k^{(1)}, q_k^{(2)}, q_k^{(3)}\}$. These constitute the system state $S = [\Delta_k, E_k, B_{net}, q^{(1)}, q^{(2)}, q^{(3)}]$. This design provides the agent with sufficient information to assess network load, resource contention, and video characteristics for informed decision-making. For example, if the RoI ratio is low but the queue length for the large model is high, the agent may prefer to process the frame locally or select a lightweight model of edge server to reduce inference latency.

Action. For each video frame, the action is defined as a composite decision comprising three elements: detection location, non-RoI resolution level, and detection model selection. Specifically, the action $A = [X_k, \rho_k, m_k]$ includes $X_k \in \{0, 1\}$, which determines whether the frame is processed locally or offloaded to the edge server; $\rho_k \in (0, 1]$, which indicates the resolution level of the non-RoI region, defined as the ratio between the downsampled and original resolution; and $m_k \in \{1, 2, 3\}$, which denotes the index of the detection model selected on the edge server, typically ordered from lightweight to heavyweight variants. For example, an action $A_k = [1, 0.5, 2]$ means that the current frame will be transmitted to the edge server, with the non-RoI downsampled to half its original resolution, and inference will be performed using model 2.

Reward. The optimization objective of *EVOD-RoI* is to maximize detection accuracy while minimizing system latency as much as possible. Based on this objective, the reward function is defined as follows: $r_k = \lambda \mathcal{A}_k - (1 - \lambda)\mathcal{L}_k$.

Advantage Actor-Critic. The DRL agent's decision-making procedure is implemented using an A2C model. In each decision step corresponding to the k-th video frame, the Actor network maps the state S_k to the composite action A_k, which jointly determines the offloading decision, resolution level, and model selection. To effectively capture the dependencies among these coupled variables, the policy network is designed with a multi-head output structure. Simultaneously, the Critic network estimates the state value function $V(S_k)$, which represents the expected cumulative discounted reward: $V(S_k) = \mathbb{E}\left[\sum_{j=0}^{\infty} \gamma^j r_{k+j} \mid S_k\right]$, where γ is the discount factor controlling the trade-off between immediate and future rewards and r_k is our designed latency-accuracy aware reward.

The learning process is driven by the advantage signal, $\delta_k = r_k + \gamma V(S_{k+1}) - V(S_k)$. To ensure stable convergence and encourage exploration, the network parameters are updated by minimizing the following joint loss function:

$$L = -\log \pi(A_k|S_k)\delta_k + \beta_c \delta_k^2 - \beta_e H(\pi(A_k|S_k)), \tag{8}$$

where the three terms represent the policy gradient loss, the value estimation error, and the entropy regularization term, respectively. Through this mechanism, the DRL agent effectively learns to maximize the cumulative return and adaptively achieve an optimal balance between real-time performance and detection precision.

3.3 Offloading Controller

The offloading controller executes task distribution based on the decision scheduler's strategy. For locally processed frames, the entire frame is kept at original resolution for on-device inference without transmission. For offloaded frames, the controller preserves the RoI (extracted by the RoI Extractor) at its original resolution, while the non-RoI is downsampled according to the resolution level specified by the Decision Scheduler. The RoI and non-RoI are separately encoded and uploaded along with position information (including RoI coordinates, original image size, and non-RoI resolution level). Upon receiving the data, the edge server decodes the non-RoI, performs upsampling, and reconstructs a regionally blurred frame of the original size by merging the RoI and non-RoI based on the position information for subsequent object detection.

3.4 Inference Detector

To ensure the edge-side detection model maintains high accuracy for video frames with resolution-reduced non-RoI, we adopt a tailored data preprocessing and fine-tuning strategy. This enhances model robustness to regionally degraded inputs, enabling accurate object detection even when non-RoI is downsampled and blurred.

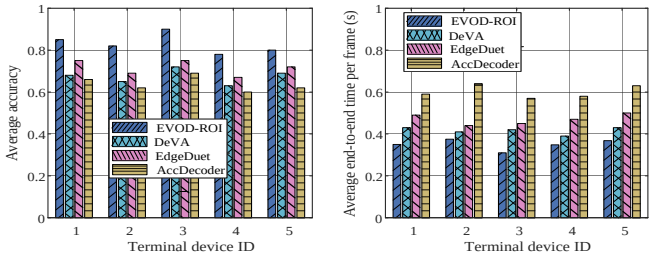
3.4.1 Regional Preprocessing Strategy. We use the object detection subset of the BDD100K [36] dataset as our training corpus. Each frame is first processed using spatiotemporal sliding window algorithm, which determines the RoI $\mathcal{P}$ based on object density and motion, and separates it from the background (non-RoI $\mathcal{B}$).

To simulate realistic region blurred video frames, we merge full-resolution $\mathcal{P}$ with resized $\mathcal{B}$ into a full-size frame. Specifically, during training, we generate four versions of each image using downsample ratios of 1.00, 0.58, 0.42, and 0.25, with proportions of 20%, 30%, 30%, and 20% (a 2:3:3:2 distribution). This exposes the model to various levels of non-RoI reduction, improving its adaptation to background information loss while preserving complete feature representations through full-resolution data. The resulting training image $\tilde{I}$ combines the original RoI with a resized and then upsampled non-RoI, constructed as follows:

$$\tilde{I}(x, y) = \begin{cases} I(x, y), & \text{if } (x, y) \in \mathcal{P} \\ \text{Upsample}(\text{Resize}(I|_{\mathcal{B}}, r)), & \text{if } (x, y) \in \mathcal{B} \end{cases}$$

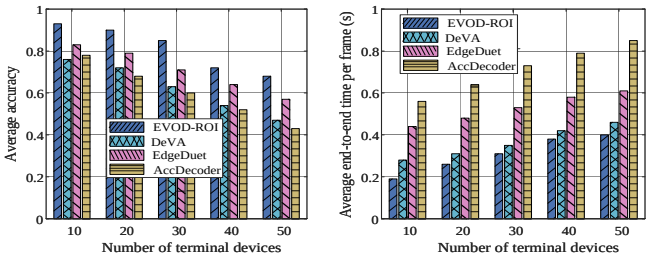
where r represents one of the predefined downsampling ratios, and "Upsample" restores $\mathcal{B}$ back to the original spatial size for seamless frame merging.

3.4.2 Fine-Tuning the Detection Model. Based on the regionally processed training set described above, various YOLO-series models are fine-tuned on the regionally processed training set to generate resolution-aware versions for deployment on the edge server. These models are trained from pre-trained weights using a unified setting that mirrors the input characteristics encountered during deployment. Each model is trained for up to 100 epochs using standard object detection losses, including CIoU loss for bounding box regression, objectness loss, and multi-class classification loss. A cosine annealing learning rate schedule is applied, starting from an initial learning rate of 1×10^{-4}, and early stopping is triggered based on validation IoU.



(a) Average accuracy. (b) Average end-to-end delay.

Figure 6: Performance comparison with SOTA methods.



(a) Average accuracy. (b) Average end-to-end delay.

Figure 7: Scalability analysis under varying scales.

3.4.3 Deployment in EVOD-RoI. All fine-tuned detection models are deployed within the inference detector. Uploaded video frames are routed to the corresponding model instance for object detection according to the strategy generated by the decision scheduler. This ensures robustness and accuracy across diverse conditions, particularly when frames exhibit partially degraded non-RoI.

4 Performance Evaluation

4.1 System Settings

4.1.1 Testbed. Our platform includes five NVIDIA Jetson TX2 devices as terminals and a Supermicro M12SWA-TF edge server, equipped with an AMD 5995WX CPU and three NVIDIA RTX 3090 GPUs. The terminals process five real-world YouTube video clips [31–35] and run a lightweight YOLOv5s locally. In contrast, the edge server runs fine-tuned, high-accuracy models (YOLOv5m-RB, YOLOv5l-RB, and YOLOv5x-RB) trained on our custom regionally blurred dataset. To establish ground truth, we used a large YOLOv5x6 model for pre-labeling, followed by manual verification and correction of all labels and missed small objects, consistent with previous studies [10, 38]. Devices communicate over a campus LAN with a typical bandwidth of 10-16 MB/s. At each time epoch, the weight factor λ is uniformly sampled from the range [0.2, 0.7].

4.1.2 Metrics. We assess the system performance using the following metrics: (1) Accuracy: The overall detection accuracy in five video types; (2) Time: The end-to-end latency, which encompasses algorithm execution time, model inference time, and video transmission time.

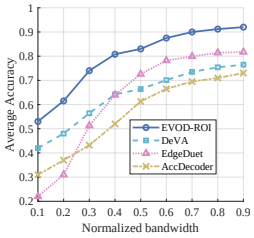


Figure 8: The impact of bandwidth changes.

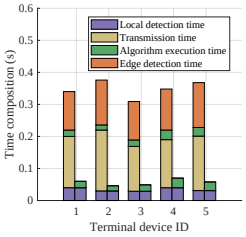


Figure 9: Time composition of *EVOD-RoI* components.

4.2 System performance

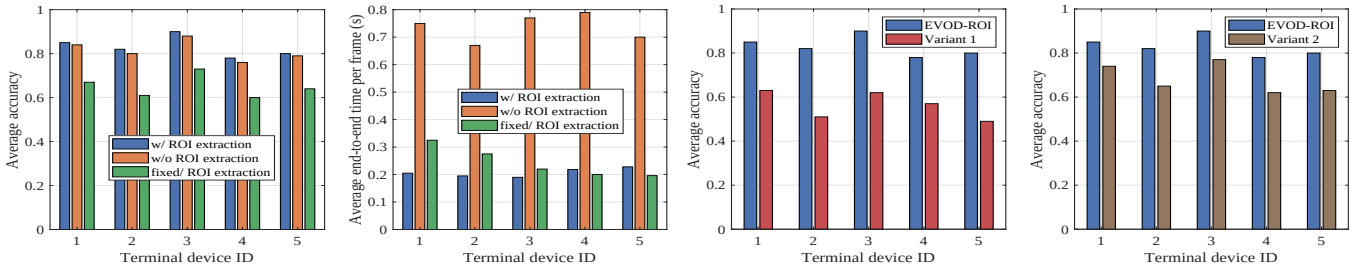
4.2.1 Benchmarks. We compare *EVOD-RoI* against the following three state-of-the-art (SOTA) approaches: (1) DeVA, dynamically adjusts the resolution of offline videos and optimizes quantization parameters for both RoI and non-RoI [4]; (2) EdgeDuet, adopts tile-level parallel processing to segment video frames and uploads selected tiles to the edge for detection [30]; (3) AccDecoder, adaptively selects a small number of key frames for full-frame transmission and applies super-resolution enhancement to improve accuracy [37].

4.2.2 Comparison with State-of-the-Art. Fig. 6 compares the average detection accuracy and end-to-end latency across the four evaluated approaches. As shown in Fig. 6(a), *EVOD-RoI* achieves a 15.9%-30.1% improvement in average detection accuracy over the other three methods. Fig. 6(b) illustrates that *EVOD-RoI* also reduces the average end-to-end latency by 16.8%–41.8% compared to the alternatives. Moreover, both accuracy and latency exhibit varying trends across different video types. These results collectively demonstrate that *EVOD-RoI* outperforms existing SOTA methods in both detection accuracy and latency control.

4.2.3 Performance in Large-scale scenarios. To evaluate the performance of *EVOD-RoI* in large-scale scenarios, we vary the number of terminal devices from 10 to 50, with each device processing heterogeneous video streams. As shown in Fig. 7(a) and (b), *EVOD-RoI* consistently achieves 15.3%-32.9% higher average accuracy and 16.4%-56.9% lower end-to-end delay than the other methods across all scales, validating its robustness under large-scale deployment.

4.2.4 The impact of bandwidth changes. Fig. 8 illustrates the performance of each scheme under dynamic normalized bandwidth conditions. The results demonstrate that *EVOD-RoI* consistently outperforms the other three baselines across various bandwidth levels. Notably, when the normalized bandwidth is 0.1, *EVOD-RoI* achieves a 11.8%–31.3% improvement in detection accuracy compared to the other baseline methods, highlighting its robustness and performance superiority in bandwidth-fluctuating environments.

4.2.5 Running time of each component. Fig. 9 presents the runtime breakdown of each component within the *EVOD-RoI* system. For frames offloaded to the edge, the total latency comprises local processing time (including YOLOv5s inference and RoI extraction, ~ 4%), transmission time (~ 55%), DRL-based scheduling overhead (~ 3%), and edge-side inference (~ 38%). In contrast, non-offloaded frames involve only local detection and scheduling. The results indicate that both RoI extraction and DRL execution contribute minimally to the overall latency, underscoring the real-time adaptability and efficiency of *EVOD-RoI*'s scheduling mechanism.



(a) Average accuracy. **(b) Average end-to-end delay.** **(c) Average accuracy.** **(d) Average accuracy.**

Figure 10: Ablation study on RoI extractor, decision scheduler and inference detector.

4.3 The Ablation Study of *EVOD-RoI* System

4.3.1 Adaptive RoI extraction. To verify the RoI extraction module's role, we evaluated three system variants, keeping other modules fixed: the full *EVOD-RoI* system with adaptive RoI extraction (Algorithm 1); (2) **w/o RoI extraction**: a version uploading entire frames directly; and (3) **fixed/ RoI extraction**: a variant using a static RoI (one-third of the frame). As shown in Fig. 10(a) and (b), our adaptive approach improves detection accuracy by 2.1% and 27.6% and reduces end-to-end latency by 71.8% and 14.8% compared to the w/o and fixed/ RoI variants, respectively. These results underscore the effectiveness and necessity of dynamic RoI extraction for overall system performance.

4.3.2 Adaptive decision scheduling. To evaluate the pivotal role of the DRL-based decision scheduler in enhancing detection accuracy, we performed a component-level replacement while keeping the rest of the *EVOD-RoI* system unchanged. Specifically, we compared the complete *EVOD-RoI* system with Variant 1, where the DRL agent is replaced by a greedy algorithm. This strategy optimizes only the immediate reward for each individual frame, ignoring the cumulative system stability and future queuing status. As illustrated in Fig. 10(c), the full *EVOD-RoI* system consistently outperforms Variant 1, achieving an average accuracy improvement of 47.2%.

4.3.3 Inference detector. To assess the fine-tuned models, we conducted an ablation study comparing the complete *EVOD-RoI* system with Variant 2, which uses original, unmodified models. As shown in Fig. 10(d), compared to Variant 2, *EVOD-RoI* improves the average detection accuracy by 12.4% while maintaining the same inference time. To further illustrate fine-tuning impacts on regionally blurred frames, Table 2 reports accuracy gains for Video 1 (results for other videos are in the Appendix C). Specifically, YOLOv5m-RB, YOLOv5l-RB, and YOLOv5x-RB outperform their baselines by 9.8%, 9.8%, and 10.9%, respectively, demonstrating the effectiveness of fine-tuning in handling partially degraded visual inputs.

Table 2: Performance of different detection models

Video Type	Detection Model	Accuracy	Missed	False
Video 1	YOLOv5m-RB	**80.9%**	**6.8%**	**7.1%**
	YOLOv5m	71.1%	10.5%	9.3%
	YOLOv5l-RB	**85.4%**	**5.2%**	**6.1%**
	YOLOv5l	75.6%	8.2%	8.7%
	YOLOv5x-RB	**91.7%**	**2.1%**	**1.3%**
	YOLOv5x	80.8%	7.4%	8.1%

5 Related Work

Most existing edge-assisted video analytics treat the entire video frame as the basic unit for transmission and processing [18, 29]. These full-frame studies employ diverse strategies such as bitrate adaptation [13, 19, 28], keyframe identification [1, 11, 29], and resolution optimization [9, 18, 22] to ensure stability, coupled with edge-side resource allocation [17, 25, 40], super-resolution [20, 37], and adaptive model configuration [10, 28] to improve accuracy. To further enhance efficiency, recent research has begun focusing on RoI-based bandwidth reduction via frame partitioning [30, 39] or region extraction [4, 5]. However, two critical gaps remain in current RoI-based approaches: first, RoI identification typically relies on static priors or simple motion cues, failing to adapt to the complex spatiotemporal variations inherent in diverse video content [2, 21]; second, in the processing stage, standard detection models lack robustness against the resolution heterogeneity of reconstructed hybrid frames, leading to suboptimal inference performance [14].

Our work addresses these issues through three key advancements. First, a spatiotemporal sliding window integrating motion and density for dynamic RoI capture. Second, we introduce a targeted fine-tuning strategy on regionally blurred datasets, explicitly enhancing model robustness against background degradation. Third, RoI-aware joint optimization of transmission and processing. By combining these elements, our approach overcomes the constraints of previous isolated and static strategies.

6 Conclusion

We propose *EVOD-RoI*, an edge-assisted video object detection system based on RoI-aware transmission and scheduling. It jointly improves detection accuracy and system latency by preserving high-resolution RoIs while downsampling non-RoI regions. Specifically, *EVOD-RoI* employs an adaptive RoI extraction algorithm to capture spatiotemporally heterogeneous RoIs, a DRL agent to jointly optimize video transmission and processing decisions, and a targeted fine-tuning strategy based on regionally blurred data to enhance detection accuracy under spatially varying quality. Experiments on a real-world testbed show that *EVOD-RoI* outperforms SOTA methods and significantly improves system QoS. Future work will extend the framework to multi-camera collaborative video analytics in complex edge environments.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (No. 2023YFB2703600), National Natural Science Foundation of China under Grant 62572280, U24A20244.

References

[1] Md Adnan Arefeen and Md Yusuf Sarwar Uddin. 2021. Towards Resource-Efficient Detection-Driven Processing of Multi-Stream Videos. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*. 843–845.

[2] Shubham Chaudhary, Aryan Taneja, Anjali Singh, Purbasha Roy, Sohum Sikdar, Mukulika Maity, and Arani Bhattacharya. 2024. TileClipper: Lightweight Selection of Regions of Interest from Videos for Traffic Surveillance. In *Proceedings of the USENIX Annual Technical Conference*. 967–984.

[3] Chen Chen, Bin Liu, Shaohua Wan, Peng Qiao, and Qingqi Pei. 2021. An Edge Traffic Flow Detection Scheme Based on Deep Learning in an Intelligent Transportation System. *IEEE Transactions on Intelligent Transportation Systems* 22, 3 (2021), 1840–1852.

[4] Shutong Chen, Jingwen Yin, Ruichao Zhong, and Fangming Liu. 2025. DeVA: An Edge-Assisted Video Analytics Framework for Depth Estimation. *IEEE Transactions on Mobile Computing* 24, 12 (2025), 13177–13190.

[5] Yan Cheng, Peng Yang, Ning Zhang, and Jiawei Hou. 2023. Edge-Assisted Lightweight Region-of-Interest Extraction and Transmission for Vehicle Perception. In *Proceedings of the IEEE Global Communications Conference*. 1054–1059.

[6] Xuejian Chi, Honglong Chen, Guoxin Li, Zhichen Ni, Nan Jiang, and Feng Xia. 2024. EDSP-Edge: Efficient Dynamic Edge Service Entity Placement for Mobile Virtual Reality Systems. *IEEE Transactions on Wireless Communications* 23, 4 (2024), 2771–2783.

[7] Xuejian Chi, Honglong Chen, Zhichen Ni, Haiyang Sun, Peng Sun, and Dongxiao Yu. 2025. H-STEP: Heuristic Stable Edge Service Entity Placement for Mobile Virtual Reality Systems. *IEEE Transactions on Mobile Computing* 24, 8 (2025), 7377–7388.

[8] Xuejian Chi, Xiaoya Jin, and Dapeng Jiao. 2026. TSP-DRL: High-Robustness Service Deployment for Mobile Virtual Reality. *Ad Hoc Networks* 183 (2026).

[9] Penglin Dai, Yangyang Chao, Xiao Wu, Kai Liu, and Songtao Guo. 2024. Context-Aware Offloading for Edge-Assisted On-Device Video Analytics Through Online Learning Approach. *IEEE Transactions on Mobile Computing* 23, 12 (2024), 12761–12777.

[10] Xiangxiang Dai, Zeyu Zhang, Peng Yang, Yuedong Xu, Xutong Liu, and John C. S. Lui. 2024. AxiomVision: Accuracy-Guaranteed Adaptive Visual Model Selection for Perspective-Aware Video Analytics. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 7229–7238.

[11] Mengxi Hanyao, Yibo Jin, Zhuzhong Qian, Sheng Zhang, and Sanglu Lu. 2021. Edge-Assisted Online On-Device Object Detection for Real-Time Video Analytics. In *Proceedings of the IEEE Conference on Computer Communications*. 1–10.

[12] Glenn Jocher et al. 2020. YOLOv5 by Ultralytics. Accessed: July 27, 2025. https://github.com/ultralytics/yolov5

[13] Xishuo Li, Shan Zhang, Yuejiao Huang, Xiao Ma, Zhiyuan Wang, and Hongbin Luo. 2024. Towards Timely Video Analytics Services at the Network Edge. *IEEE Transactions on Mobile Computing* 23, 11 (2024), 10443–10459.

[14] Liqun Lin, Mingxing Wang, Jing Yang, Keke Zhang, and Tiesong Zhao. 2024. Toward Efficient Video Compression Artifact Detection and Removal: A Benchmark Dataset. *IEEE Transactions on Multimedia* 26 (2024), 10816–10827.

[15] Fangxin Liu, Junjie Wang, Ning Yang, Zongwu Wang, Junping Zhao, Li Jiang, and Haibing Guan. 2025. ASTER: Adaptive Dynamic Layer-Skipping for Efficient Transformer Inference via Markov Decision Process. In *Proceedings of the 33rd ACM International Conference on Multimedia*. 11853–11861.

[16] Yong Liu, Jinshan Pan, Yinchuan Li, Qingji Dong, Chao Zhu, Yu Guo, and Fei Wang. 2025. UltraVSR: Achieving Ultra-Realistic Video Super-Resolution with Efficient One-Step Diffusion Space. In *Proceedings of the 33rd ACM International Conference on Multimedia*. 7785–7794.

[17] Weibin Ma and Lena Mashayekhy. 2024. Video Offloading in Mobile Edge Computing: Dealing With Uncertainty. *IEEE Transactions on Mobile Computing* 23, 11 (2024), 10251–10264.

[18] Taslim Murad, Anh Nguyen, and Zhisheng Yan. 2022. DAO: Dynamic Adaptive Offloading for Video Analytics. In *Proceedings of the 30th ACM International Conference on Multimedia*. 3017–3025.

[19] Bin Qian, Zhenyu Wen, Junqi Tang, Ye Yuan, Albert Y. Zomaya, and Rajiv Ranjan. 2023. OsmoticGate: Adaptive Edge-Based Real-Time Video Analytics for the Internet of Things. *IEEE Trans. Comput.* 72, 4 (2023), 1178–1193.

[20] Jiahao Shen, Hao Sheng, Shuai Wang, Ruixuan Cong, Da Yang, and Yang Zhang. 2024. Blockchain-Based Distributed Multiagent Reinforcement Learning for Collaborative Multiobject Tracking Framework. *IEEE Trans. Comput.* 73, 3 (2024), 778–788.

[21] Jiangang Shen, Hongzi Zhu, Liang Zhang, Yunzhe Li, Shan Chang, Jie Wu, and Minyi Guo. 2025. DiVE: Differential Video Encoding for Online Edge-Assisted Video Analytics on Mobile Agents. In *Proceedings of the IEEE International Conference on Distributed Computing Systems*. 243–252.

[22] Can Wang, Sheng Zhang, Yu Chen, Zhuzhong Qian, Jie Wu, and Mingjun Xiao. 2020. Joint Configuration Adaptation and Bandwidth Allocation for Edge-Based Real-Time Video Analytics. In *Proceedings of the IEEE Conference on Computer Communications*. 257–266.

[23] Jianyi Wang, Mai Xu, Lai Jiang, and Yuhang Song. 2021. Attention-Based Deep Reinforcement Learning for Virtual Cinematography of 360-Degree Videos. *IEEE Transactions on Multimedia* 23 (2021), 3227–3238.

[24] Runjie Wang, Kemi Chen, Shuijie Li, Mingkai Chen, and Tiesong Zhao. 2025. Efficient Semantic Codec for Real-Time Vibrotactile Transmission. In *Proceedings of the 33rd ACM International Conference on Multimedia*. 12092–12101.

[25] Rui Wang, Yixue Hao, Yiming Miao, Long Hu, and Min Chen. 2024. RT3C: Real-Time Crowd Counting in Multi-Scene Video Streams via Cloud-Edge-Device Collaboration. *IEEE Transactions on Services Computing* 17, 4 (2024), 1739–1752.

[26] Yuan Wang, Xiaona Lv, Xiaohu Gu, and Dan Li. 2025. A New Edge Computing Method Based on Deep Learning in Intelligent Transportation System. *IEEE Transactions on Vehicular Technology* 74, 2 (2025), 2210–2219.

[27] Wenquan Xu, Haoyu Song, Linyang Hou, Hui Zheng, Xinggong Zhang, Chuwen Zhang, Wei Hu, Yi Wang, and Bin Liu. 2021. SODA: Similar 3D Object Detection Accelerator at Network Edge for Autonomous Driving. In *Proceedings of the IEEE Conference on Computer Communications*. 1–10.

[28] Mingxuan Yan, Yi Wang, Xuedou Xiao, Zhiqing Luo, Jianhua He, and Wei Wang. 2023. Think before You Leap: Content-Aware Low-Cost Edge-Assisted Video Semantic Segmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 9224–9233.

[29] Yuting Yan, Sheng Zhang, Xiaokun Wang, Ning Chen, Yu Chen, Yu Liang, Mingjun Xiao, and Sanglu Lu. 2024. VisFlow: Adaptive Content-Aware Video Analytics on Collaborative Cameras. In *Proceedings of the IEEE Conference on Computer Communications*. 2019–2028.

[30] Zheng Yang, Xu Wang, Jiahang Wu, Yi Zhao, Qiang Ma, Xin Miao, Li Zhang, and Zimu Zhou. 2023. EdgeDuet: Tiling Small Object Detection for Edge Assisted Autonomous Mobile Vision. *IEEE/ACM Transactions on Networking* 31, 4 (2023), 1765–1778.

[31] YouTube. 2012. AVG-TownCentre: Original Video. https://www.youtube.com/watch?v=pC0FLXv5p1U. Accessed: March 30, 2025.

[32] YouTube. 2019. GeoVision VD8700 Face Recognition IP Camera Demo Video. https://www.youtube.com/watch?v=Szt_kHWKwNo. Accessed: March 30, 2025.

[33] YouTube. 2022. 4K Road Traffic Video for Object Detection and Tracking. https://www.youtube.com/watch?v=MNn9qKG2UFI. Accessed: March 30, 2025.

[34] YouTube. 2023. 4K Camera Example for Traffic Monitoring Road, Interchange. https://www.youtube.com/watch?v=Y7Kv_oJtuJE. Accessed: March 30, 2025.

[35] YouTube. 2023. Road Traffic Video for Object Recognition. https://www.youtube.com/watch?v=wqctLW0Hb_0. Accessed: March 30, 2025.

[36] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2020. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2636–2645.

[37] Tingting Yuan, Liang Mi, Weijun Wang, Haipeng Dai, and Xiaoming Fu. 2023. AccDecoder: Accelerated Decoding for Neural-Enhanced Video Analytics. In *Proceedings of the IEEE Conference on Computer Communications*. 1–10.

[38] Ben Zhang, Xin Jin, Sylvia Ratnasamy, John Wawrzynek, and Edward A. Lee. 2018. AWStream: Adaptive Wide-Area Streaming Analytics. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 236–252.

[39] Lei Zhang, Haobin Zhou, Haiyang Wang, and Laizhong Cui. 2024. TBSR: Tile-Based 360-Degree Video Streaming with Super-Resolution on Commodity Mobile Devices. In *Proceedings of the IEEE Conference on Computer Communications*. 501–510.

[40] Yingying Zhao, Mingzhi Dong, Yujiang Wang, Da Feng, Qin Lv, Robert P. Dick, Dongsheng Li, Tun Lu, Ning Gu, and Li Shang. 2022. A Reinforcement-Learning-Based Energy-Efficient Framework for Multi-Task Video Analytics Pipeline. *IEEE Transactions on Multimedia* 24 (2022), 2150–2163.