Semantic Communications With Computer Vision Sensing for Edge Video Transmission

Yubo Peng, *Graduate Student Member, IEEE*, Luping Xiang, *Senior Member, IEEE*, Kun Yang, *Fellow, IEEE*, Kezhi Wang, *Senior Member, IEEE*, and Mérouane Debbah, *Fellow, IEEE*

***Abstract*—Despite the widespread adoption of vision sensors in edge applications, such as surveillance, video transmission consumes substantial spectrum resources. Semantic communication (SC) offers a solution by extracting and compressing information at the semantic level, but traditional SC without sensing capabilities faces inefficiencies due to the repeated transmission of static frames in edge videos. To address this challenge, we propose an SC with computer vision sensing (SCCVS) framework for edge video transmission. The framework first introduces a compression ratio (CR) adaptive SC (CRSC) model, capable of adjusting CR based on whether the frames are static or dynamic, effectively conserving spectrum resources. Simultaneously, we present a knowledge distillation (KD)-based approach to ensure the efficient learning of the CRSC model. Additionally, we implement a computer vision (CV)-based sensing model (CVSM) scheme, which intelligently perceives the scene changes by detecting the movement of the sensing targets. Therefore, CVSM can assess the significance of each frame through in-context analysis and provide CR prompts to the CRSC model based on real-time sensing results. Moreover, both CRSC and CVSM are designed as lightweight models, ensuring compatibility with resource-constrained sensors commonly used in practical edge applications. Experimental results show that SCCVS improves transmission accuracy by approximately 70% and reduces transmission latency by about 89% compared with baselines. We also deploy this framework on an NVIDIA Jetson Orin NX Super, achieving an inference speed of 14 ms per frame with TensorRT acceleration and demonstrating its real-time capability and effectiveness in efficient semantic video transmission.**

***Index Terms*—Semantic communication (SC), computer vision, video transmission, intelligence sensing.**

Received 21 May 2025; revised 13 November 2025; accepted 13 December 2025. Date of publication 22 December 2025; date of current version 7 May 2026. This work was supported in part by the Jiangsu Major Project on Fundamental Research under Grant BK20243059, in part by the Natural Science Foundation of China under Grant 62531008 and Grant 62301122, in part by Gusu Innovation Project under Grant ZXL2024360, in part by theHigh-Tech District of Suzhou City under Grant RC2025001, and in part by Xiaomi Young Scholar Award, and Nanjing University- China Mobile Communications Group Company, Ltd. Joint Institute. Recommended for acceptance by H. Tabassum. *(Corresponding author: Luping Xiang.)*

Yubo Peng, Luping Xiang, and Kun Yang are with the State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210008, China, and also with the Institute of Intelligent Networks and Communications (NINE), and the School of Intelligent Software and Engineering, Nanjing University (Suzhou Campus), Suzhou 215163, China (e-mail: ybpeng@smail.nju.edu.cn; luping.xiang@nju.edu.cn; kunyang@nju.edu.cn).

Kezhi Wang is with the Department of Computer Science, Brunel University London, UB8 3PH Uxbridge, U.K. (e-mail: Kezhi.Wang@brunel.ac.UK).

Mérouane Debbah is with the KU 6G Research Center, Department of Computer and Information Engineering, Khalifa University of Science and Technology, Abu Dhabi 127788, UAE, and also with Centrale Supelec, University Paris-Saclay, 91192 Gif-sur-Yvette, France (e-mail: merouane.debbah@ku.ac.ae).

Digital Object Identifier 10.1109/TMC.2025.3646710

I. Introduction

A. Backgrounds

With the rapid advancement of the Internet of Things (IoT), vision sensors are increasingly deployed to provide intelligent services, particularly in the surveillance domain. Video surveillance is highly valued not only for its ability to solve crimes but also for its potential role in crime prevention. As a result, numerous vision sensors are commonly installed in public spaces, malls, and residential areas [1], [2]. Compared to wired networks, wireless-based sensors offer greater flexibility in deployment, especially in geographically dispersed or complex environments, as they eliminate the need for physical cables [3]. While wireless transmission is more versatile, transmitting edge video, which typically involves large data sizes, imposes substantial spectrum resource demands. This hinders the development of wireless sensor-based edge applications [4].

Semantic communication (SC), a key technology for 6 G, significantly reduces the data transmission requirements by extracting and compressing information at the semantic level, while maintaining precision and relevance [5], [6]. Unlike traditional communication approaches, which prioritize error-free symbol delivery, SC focuses on achieving "semantic fidelity," effectively mitigating the "cliff effect" caused by decreasing signal-to-noise ratios (SNR) [7]. Consequently, SC offers a promising solution to the issue of spectrum scarcity. However, existing deep SC methods typically adopt a fixed compression ratio (CR) or perform uniform feature extraction for all frames, without considering temporal redundancy. This leads to unnecessary transmission of static or near-identical frames, causing spectrum inefficiency.

Radar can provide high-accuracy sensing for various applications [8], including autonomous vehicle driving, robot navigation, and indoor localization for virtual reality [9]. Therefore, radar sensing appears to offer a viable solution for detecting changes in scenes. However, this approach requires sensors to be equipped with advanced radar systems, which is difficult to implement with the widely deployed general-purpose sensors due to the high cost. Recently, the rapid development of deep learning, particularly in the field of computer vision (CV), has led to significant advancements in perception technologies. CV models, such as YOLOv10 [10] and FastSAM [11], can automatically extract features from large datasets and perform complex visual tasks such as image recognition, object detection, and semantic segmentation. Compared to the radar sensing

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

methods, CV offers substantial improvements in accuracy and robustness while running efficiently on standard hardware [12]. These advancements offer new opportunities for implementing cost-effective content-aware video transmission systems.

B. Related Work

To achieve high-efficiency video transmission, numerous studies have been conducted, focusing primarily on three aspects: video encoding, content-aware adaptive compression, and inference acceleration.

Traditional and learning-based video coding methods primarily aim to optimize rate-distortion performance. For example, Wu et al. [13] proposed a region of interest (ROI)-based video compression framework that integrates a texture-driven ROI extraction algorithm into the H.265/HEVC quad-tree structure, allowing differentiated encoding between ROI and non-ROI regions. Djelouah et al. [14] introduced an end-to-end neural video coding framework that performs optical flow–based temporal prediction in pixel space and encodes residuals in latent space, enabling unified compression of key and intermediate frames. Wang et al. [15] designed a deep video semantic transmission framework for end-to-end video transmission over wireless channels, utilizing nonlinear transforms and conditional coding to adaptively extract and transmit semantic representations across video frames using deep joint source-channel coding (JSCC). While these methods achieve significant compression gains, they are largely task-agnostic and content-unaware, limiting their effectiveness in scenarios where transmission bandwidth is constrained.

To address this, content- and task-aware compression techniques have been proposed to integrate semantic or AI-task relevance into the encoding process. Hu et al. [16] developed a content- and task-aware image compression framework for IoT cameras that jointly optimizes perceptual and inference quality under packet loss. Du et al. [17] leveraged server-side deep neural network (DNN) feedback to dynamically guide encoding decisions, while Xiao et al. [18] introduced gradient-based spatio-temporal adaptation to balance bitrate and semantic accuracy under time-varying network conditions. However, these approaches typically integrate video content analysis and codec optimization tightly into a single model, resulting in strong coupling between content perception and encoding. This results in the reliance on pre-trained task-specific models, which limits adaptability to unseen tasks or varied scenarios and reduces their scalability and practicality.

Additionally, recent work has also emphasized inference-aware and deployment-oriented designs to improve the end-to-end efficiency of video streaming. Du et al. [19] proposed AccMPEG, an accuracy-aware encoding framework that learns macroblock-level accuracy gradients to optimize encoding latency, DNN inference accuracy, and edge computational cost simultaneously. Wang et al. [20] developed Orchestra, a sensitivity-aware spatial quality adaptation framework that uses regional accuracy sensitivity to guide video zoning, quality selection, and estimation of frame-level accuracy. These methods are typically applied as post-processing optimization, meaning the models are deployed to edge devices first and then adapted for task-specific improvements. This approach introduces additional computational overhead on resource-constrained edge devices and, due to device heterogeneity, the same optimization strategy may not generalize effectively across different hardware.

C. Challenges

Based on the above analysis, despite significant advances in video compression and semantic-aware transmission, two key challenges remain for achieving high-efficiency video streaming:

1) *Coupling of Content Perception and Encoding:* Existing methods that integrate content analysis and codec optimization, whether task-aware compression frameworks or semantic-communication-based approaches (e.g., [16], [17] and [18]), are typically tightly coupled, jointly optimizing feature extraction and video encoding. Although this can enhance bandwidth efficiency and inference accuracy, such strong coupling limits adaptability to new tasks, heterogeneous devices, and dynamic network conditions, thereby reducing system scalability and generality.
2) *Edge Computation versus Low-Latency Transmission:* Balancing computational cost on edge devices with the need for low-latency video delivery remains challenging. Many inference-driven optimization methods, such as [19] and [20], rely on post-deployment fine-tuning or sensitivity estimation at the edge, introducing additional computational overhead and limiting generalization across devices with varying capabilities.

D. Contributions

To overcome the above challenges, we propose a Semantic Communication with CV Sensing (SCCVS) framework. The framework adopts a separation-based design that decouples scene perception from semantic encoding, thereby enhancing scalability and adaptability. In addition, each module employs lightweight AI models to meet the computational constraints of edge vision sensors. The key features of SCCVS, compared with existing methods, are summarized in Table I. Our main contributions are as follows:

1) *CR Adaptive Semantic Communication (CRSC):* We develop a compression-aware semantic communication model with two distinct encoding strategies: high-compression semantic encoding for static frames and low-compression encoding for scene-relevant frames. A knowledge distillation (KD)-based training approach allows the two encoding branches to learn from each other, enhancing semantic reconstruction quality under high compression while maintaining low transmission overhead.
2) *Computer Vision-based Sensing Model (CVSM):* We propose a CV-based sensing scheme that leverages object detection and semantic segmentation models to identify scene changes and provide CR guidance to the CRSC module. By decoupling perception from encoding, CVSM reduces the reliance on complex sensing equipment and enables adaptive encoding decisions in real time.

TABLE I
COMPARISON OF SCCVS WITH EXISTING METHODS

Method	Content-Aware	Semantic Encoding	Adaptive Encoding	Inference Optimization	Perception-Encoding Decoupling
Wu et al. [13]	✓	✗	✓	✗	✗
Djelouah et al. [14]	✗	✓	✓	✗	✗
Wang et al. [15]	✗	✓	✗	✗	✗
Hu et al. [16]	✓	✗	✗	✗	✗
Du et al. [17]	✓	✗	✓	✗	✗
Xiao et al. [18]	✓	✗	✓	✗	✗
Du et al. [19]	✓	✗	✗	✓	✗
Wang et al. [20]	✓	✗	✓	✓	✗
Ours	✓	✓	✓	✓	✓

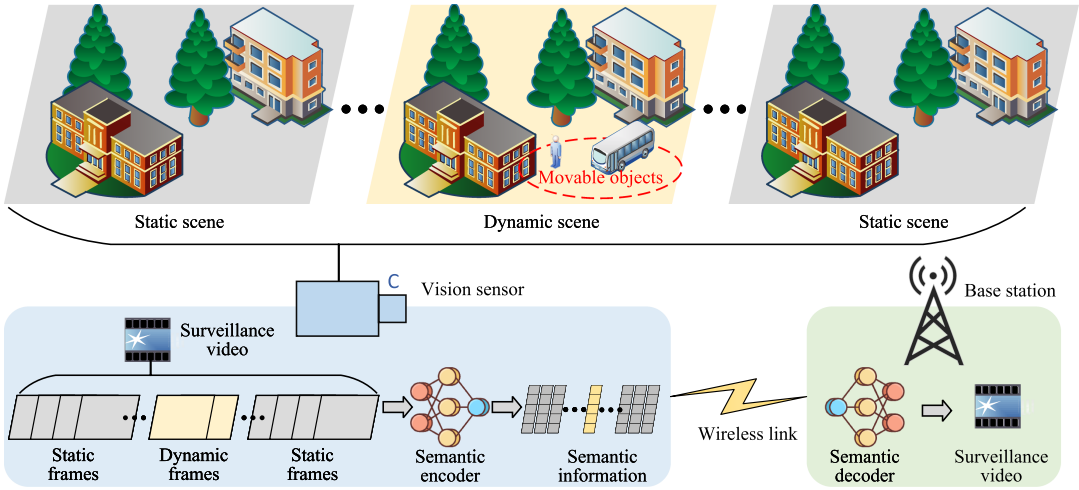


Fig. 1. The system model of SC with sensing for transmitting edge videos.

3) *Lightweight Design:* In the CRSC module, a lightweight vision transformer (ViT) and Kolmogorov-Arnold Networks (KAN) [21] are used for semantic extraction and encoding. In CVSM, quantized detection and segmentation models are employed to achieve real-time inference on general-purpose sensors, ensuring that the entire SCCVS framework operates efficiently under edge device resource constraints.
4) *Experimental Validation:* Simulations on the VIRAT Video Dataset [22] demonstrate that the proposed SCCVS framework improves transmission accuracy by approximately 70% and reduces transmission latency by about 89% compared with baseline methods. Moreover, when deployed on an NVIDIA Jetson Orin NX Super with TensorRT acceleration, SCCVS achieves a real-time inference speed of 14 ms per frame, validating the efficiency of its separation-based and lightweight design.

The structure of this paper is as follows. Section II provides a detailed description of the system model. Section III presents the proposed SCCVS framework, which mainly includes the CRSC and CVSM schemes. Section IV employs experimental simulations to evaluate the performance of the proposed methods. Lastly, Section V concludes this paper.

II. SYSTEM MODEL

As illustrated in Fig. 1, a video surveillance scenario is considered where the visual sensor has a fixed camera angle, such as in a parking lot, resulting in a static monitoring scene. The primary variations within the scene are due to movable objects like pedestrians and vehicles. These objects, however, are typically in motion for only a small fraction of the time and remain stationary for the majority of the observation period. Due to limited local storage capacity, the sensor must transmit the captured video data to a nearby base station (BS) via a wireless link. To address the issue of spectrum scarcity, an SC system with sensing capabilities at both the sensor and the BS is implemented for data transmission. In this system, the sensor functions as the transmitter, while the BS serves as the receiver. A semantic encoder is deployed at the sensor to extract and encode semantic information from the raw video data, and a semantic decoder is employed at the BS to decode the information and reconstruct the video. This approach transmits only high-density semantic information instead of raw video data, significantly reducing the bandwidth requirements.

A. Semantic Communication Model

In the proposed SC model, the captured raw video is transmitted frame-by-frame. This scenario is modeled as a point-to-point wireless image transmission system enabled by deep JSCC. The raw video is denoted as $\mathcal{V} = \{\mathbf{x}_i | i \in \{1, \ldots, V\}\}$, where $\mathbf{x}_i$ represents the ith frame, and V is the total number of frames. The primary objective is to reconstruct all video frames at the receiver (i.e., the BS) transmitted by the transmitter (i.e., the vision sensor) under varying channel SNR and CR conditions.

1) Encoder: Each frame is assumed to have a height H, width W, and depth C. The source bandwidth of each frame is defined as $m = H \times W \times C$, leading to a total source bandwidth of $m \cdot V$ for the entire raw video. At the transmitter, a semantic encoder coupled with signal modulation transforms the ith frame $\mathbf{x}_i$ into an n_i-dimensional complex vector $\mathbf{c}_i \in \mathbb{C}_i^n$. This process is formulated as:

$$\mathbf{c}_i = F_{\text{se}}(\mathbf{x}_i, r_i, \alpha), \tag{1}$$

where $F_{\text{se}}(\cdot)$ denotes the semantic encoder with parameters α, and $r_i = (m - n_i)/m$ is the CR for frame $\mathbf{x}_i$ [23]. Accordingly, the overall CR of the edge video is expressed as:

$$r = \frac{\sum_{i=1}^{V} r_i}{V}. \tag{2}$$

2) Wireless Channel: When transmitted over a wireless fading channel, the complex vector $\mathbf{c}_i$ is subject to transmission impairments, including distortion and noise. This transmission process can be modeled as:

$$\mathbf{y}_i = \mathbf{H} \cdot \mathbf{c}_i + \mathbf{N}, \tag{3}$$

where $\mathbf{y}_i$ is the received complex vector, $\mathbf{H}$ represents the channel gain between the transmitter and receiver, and $\mathbf{N}$ denotes the Additive White Gaussian Noise (AWGN). To enable end-to-end training of both the encoder and decoder, the channel model must support backpropagation. Consequently, the wireless channel is simulated using neural network-based approaches [24].

3) Decoder: Upon receiving the vector $\mathbf{y}_i$, the semantic decoder is responsible for reconstructing the corresponding frame. This reconstruction process can be expressed as:

$$\hat{\mathbf{x}}_i = F_{\text{sd}}(\mathbf{y}_i, r_i, \beta), \tag{4}$$

where $F_{\text{sd}}(\cdot)$ denotes the semantic decoder parameterized by β, and $\hat{\mathbf{x}}_i$ represents the reconstructed ith frame. Upon completing the transmission of all frames, the reconstructed video, denoted as $\hat{\mathcal{V}}$, is obtained.

B. Delay Model

During the uplink transmission of the complex vector from the vision sensor to the BS, the transmission rate can be expressed as [25]:

$$v = B \log_2 (1 + \phi), \tag{5}$$

where B indicates the bandwidth and ϕ denotes the SNR. The transmission delay for the ith frame is then given by:

$$t_i = \frac{Z(\mathbf{c}_i)}{v}, \tag{6}$$

where $Z(\mathbf{c}_i)$ represents the number of bits required to transmit the complex vector $\mathbf{c}_i$ to the BS. Consequently, the total transmission delay for the edge video can be calculated as:

$$T = \sum_{i=1}^{V} t_i. \tag{7}$$

C. Problem Formulation

Considering that edge video frames often contain high levels of redundancy with limited valuable information, traditional metrics that assess the consistency of every frame between the raw and reconstructed videos may not be appropriate. To more accurately assess the performance of video SC, it is essential to focus on minimizing differences in valuable frames while accounting for transmission delays. Thus, the objective function of the proposed SC system for edge video can be formulated as:

$$\min_{\alpha,\beta,\mathcal{R}} \frac{1}{V} \sum_{i=1}^{V} (\mathbf{x}_i - \hat{\mathbf{x}}_i)^2 \cdot (1 - r_i) + \zeta T, \tag{8a}$$

$$\text{s.t. } r_i \in \{0, 1\}, \forall i \in \{1, \ldots, V\}, \tag{8b}$$

where $\mathcal{R} = \{r_i | i = 1, \ldots, V\}$ represents the set of CR for each frame, and ζ denotes an adjustment coefficient. The constraint in Eq. (8b) indicates that each frame is either compressed or not.

To solve the optimization problem in Eq. (8a), we propose the SCCVS framework. On the one hand, the CRSC module is designed to minimize the distortion term $(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2$ by optimizing the parameters α and β during model training. On the other hand, the CVSM module dynamically senses frame changes to optimize r_i, thereby reducing the transmission delay associated with static frames.

III. Proposed SCCVS Framework

A. Overview

In practical scenarios, addressing the substantial spectrum resource consumption caused by video transmission from vision sensors is crucial. To this end, we introduce the SCCVS framework, which integrates SC and CV sensing technologies to achieve efficient video transmission. As depicted in Fig. 2, the framework consists of two primary modules:

1) CRSC for Edge Video Transmission: For each video frame, the CRSC module first utilizes a lightweight ViT to extract semantic information from a given frame $\mathbf{x}_i$, generating a high-dimensional semantic representation $\mathbf{s}_i$. A KAN is then employed to compress $\mathbf{s}_i$ based on the specified CR r_i, resulting in a semantic encoding $\mathbf{e}_i$ with either a high or low CR. This semantic encoding is subsequently modulated into a complex vector $\mathbf{c}_i$ for wireless transmission. At the receiver side, the complex vector $\mathbf{y}_i$, which may include noise distortions, is demodulated and processed by a KAN and ViT-based semantic decoder to reconstruct the frame $\hat{\mathbf{x}}_i$. The CRSC scheme is detailed in **Algorithm 2**. Upon completing the transmission of all frames, the reconstructed video $\hat{\mathcal{V}}$ is obtained. Through this innovative SC approach, the CRSC module achieves highly efficient data transmission for edge video.

2) CVSM for Edge Video Sensing: To accurately detect changes within the scene and dynamically adjust the CR in the CRSC module, while minimizing the sensing cost for vision sensors, the CVSM module employs CV-based models to analyze video frames. Specifically, the framework employs an object detection model (ODM) to detect movable objects,

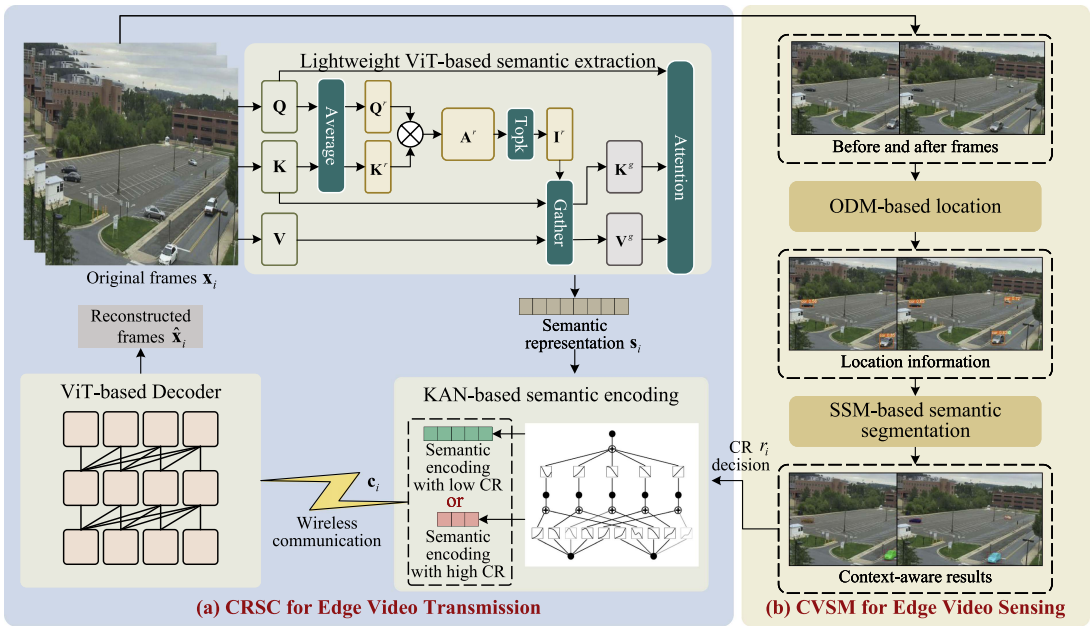


Fig. 2. The illustration of the SCCVS framework composition: (a) the CRSC and (b) the CVSM modules.

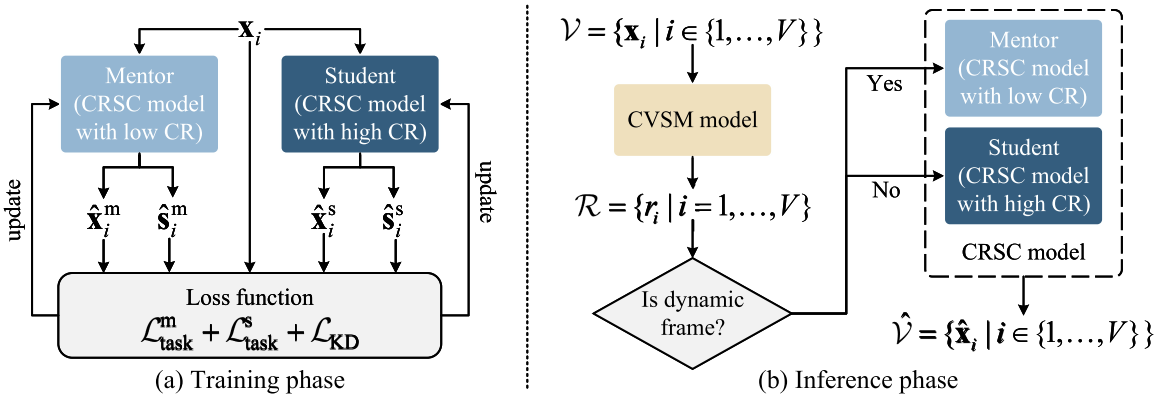


Fig. 3. Dataflow of the proposed SCCVS framework in (a) the training phase and (b) the inference phase.

such as pedestrians and vehicles, within the frames. A semantic segment model (SSM) is then applied to segment key targets, isolating their corresponding pixel sets. The identified elements are subsequently analyzed across frames to detect contextual changes in the scene. Based on these results, the CR r_i for the current frame is determined and fed into the CRSC module to guide the semantic encoding process. **Algorithm 4** provides a detailed description of the CVSM scheme. By CV-based image processing techniques, we reduce the high sensing costs typically associated with radar-based equipment.

Assuming the $\mathcal{D}$ represents the training dataset. To facilitate understanding, Fig. 3 illustrates the data flow of the proposed SCCVS framework, and **Algorithm 1** outlines the workflow.

B. CRSC for Edge Video Transmission

To enable efficient transmission of edge video while addressing the sensor's resource constraints, we propose the CRSC scheme. Since the semantic encoder is deployed on the sensor side, we adopt a lightweight ViT for semantic feature extraction, as it efficiently captures global spatial dependencies with minimal parameters compared to conventional CNNs. For semantic encoding, we employ a KAN, which provides high nonlinear fitting capability and strong generalization with compact architectures, further reducing model complexity. This combination effectively minimizes the computational burden while ensuring high-quality video reconstruction. Considering the high redundancy in video frames, two SC models with distinct CRs are designed to handle static and dynamic frames separately. Moreover, to enhance transmission robustness, especially under high compression, we introduce a KD mechanism to enable mutual learning between the two SC models. The key modules of the CRSC scheme are detailed as follows:

1) Light ViT-Based Semantic Extraction: As illustrated in Fig. 2(a), we present a lightweight ViT that incorporates a bilevel routing attention (BRA) mechanism [26], which efficiently

Algorithm 1: SCCVS Framework Workflow.

Input: Raw video $\mathcal{V}$, training dataset $\mathcal{D}$.
Output: Reconstructed video $\hat{\mathcal{V}}$, semantic encoder's parameters α, semantic encoder's parameters β.
Inference Phase:
1: **for** $i = 1, 2, \ldots, V$ **do**
2: Sense the current frame $\mathbf{x}_i$ and determine the corresponding CR r_i using **Algorithm 4**.
3: Reconstruct the frame $\hat{\mathbf{x}}_i$ based on **Algorithm 2** and the determined CR r_i.
4: **end for**
5: Combine all reconstructed frames to obtain the final reconstructed video $\hat{\mathcal{V}}$.
6: Assess the transmission quality of the video SC based on the objective function in Eq. (8a).
Training Phase:
7: Obtain the trained parameters: α and β, by training the CRSC model according to **Algorithm 3**, using $\mathcal{D}$.

captures both global semantic dependencies and local structural cues.

First, each input frame $\mathbf{x}_i$ is divided into $S \times S$ non-overlapping patches through a patch embedding layer, resulting in $\mathbf{x}_i^r \in \mathbb{R}^{S^2 \times HW/S^2 \times C}$. Linear projections are then applied to obtain the query, key, and value tensors:

$$\mathbf{Q} = \mathbf{x}_i^r \mathbf{W}^q, \quad \mathbf{K} = \mathbf{x}_i^r \mathbf{W}^k, \quad \mathbf{V} = \mathbf{x}_i^r \mathbf{W}^v, \tag{9}$$

where $\mathbf{W}^q$, $\mathbf{W}^k$, and $\mathbf{W}^v$ denote the corresponding learnable projection matrices.

Then, to model global dependencies, region-level queries and keys, $\mathbf{Q}^r$ and $\mathbf{K}^r$, are obtained by averaging token features within each patch. Their correlation matrix, $\mathbf{A}^r = \mathbf{Q}^r (\mathbf{K}^r)^{\mathrm{T}}$, reflects semantic affinities between regions. To mitigate the influence of irrelevant or noisy regions, only the top-k strongest semantic connections are retained through a routing index matrix $\mathbf{I}^r = \mathrm{topk}(\mathbf{A}^r)$. This region-level routing establishes a coarse semantic graph that guides subsequent fine-grained attention.

Next, based on $\mathbf{I}^r$, semantically related key-value pairs are selectively gathered for detailed token-level attention:

$$\mathbf{K}^g = \mathrm{gat}texther\left(\mathbf{K}, \mathbf{I}^r\right), \quad \mathbf{V}^g = \mathrm{gather}\left(\mathbf{V}, \mathbf{I}^r\right), \tag{10}$$

The final output of the BRA layer is computed as:

$$\mathbf{O} = \mathrm{Attention}\left(\mathbf{Q}, \mathbf{K}^g, \mathbf{V}^g\right) + \mathrm{LCE}(\mathbf{V}), \tag{11}$$

where the first term performs attention restricted to routed regions, and $\mathrm{LCE}(\cdot)$ denotes the local context enhancement term that reinforces neighborhood consistency and prevents semantic fragmentation.

Finally, a linear projection layer F_{Proj} is applied to map the aggregated output into a high-level semantic feature space:

$$\mathbf{s}_i = F_{\mathrm{Proj}}(\mathbf{O}). \tag{12}$$

Overall, the BRA mechanism-based lightweight ViT enables selective attention among semantically correlated regions, effectively reducing quadratic attention complexity while enhancing interpretability. By coupling region-level routing with local context enhancement, it preserves essential structural details and mitigates over-sparsification.

2) KAN-Based Semantic Encoding: First, the semantic representation $\mathbf{s}_i$ is flattened into an n-dimensional vector, $\mathbf{s}_i^{\mathrm{F}} = [\mathbf{s}_{i,1}^{\mathrm{F}}, \mathbf{s}_{i,2}^{\mathrm{F}}, \ldots, \mathbf{s}_{i,n}^{\mathrm{F}}] \in \mathbb{R}^n$. Based on the Kolmogorov–Arnold theorem, any continuous function $f(\cdot)$ can be represented as:

$$f(\mathbf{s}_{i,1}^{\mathrm{F}}, \mathbf{s}_{i,2}^{\mathrm{F}}, \ldots, \mathbf{s}_{i,n}^{\mathrm{F}}) = \sum_{q=1}^{2n+1} \varphi_q \left(\sum_{p=1}^{n} \psi_{qp}(\mathbf{s}_i^{\mathrm{F}}) \right), \tag{13}$$

where φ_q and ψ_{qp} are learnable nonlinear functions used for feature mapping and combination. Eq (13) provides the theoretical basis for KAN by demonstrating that any n-dimensional function can be constructed through a set of one-dimensional functions.

Next, KAN applies a nonlinear mapping to each component $\mathbf{s}_i^{\mathrm{F}}$, generating intermediate feature representations. This mapping is performed by the function ψ_{qp}, defined as:

$$h_{qp} = \psi_{qp}(\mathbf{s}_i^{\mathrm{F}}), \tag{14}$$

where h_{qp} represents the mapped output for each input component x_p. This mapping is applied to each dimension $p = 1, 2, \ldots, n$ producing distinct intermediate features h_{qp}. These intermediate features are then combined to form the final output representation. For each q, the combined feature u_q is computed as:

$$u_q = \sum_{p=1}^{n} h_{qp}. \tag{15}$$

Finally, a set of nonlinear functions, φ_q is applied to the combined features u_q, producing the final semantic encoding:

$$\mathbf{e}_i = \sum_{q=1}^{2n+1} \varphi_q(u_q) = \sum_{q=1}^{2n+1} \varphi_q \left(\sum_{p=1}^{n} \psi_{qp}(\mathbf{s}_i^{\mathrm{F}}) \right), \tag{16}$$

where $\mathbf{e}_i \in \mathbb{R}^{m \cdot r_i}$ represents the semantic encoding based on the given CR r_i. After modulating on $\mathbf{e}_i$, the complex vector $\mathbf{c}_i$ is obtained and transmitted over the wireless channel, to be decoded at the receiver.

Unlike conventional multilayer perceptrons (MLPs) that rely on high-dimensional matrix multiplications, KAN decomposes multivariate mappings into combinations of low-dimensional nonlinear functions, significantly reducing parameter count and computational overhead. Therefore, the KAN-semantic encoder not only ensures theoretical completeness in representing complex semantic relationships but also provides a compact and efficient encoding for resource-constrained edge devices.

The inference phase of CRSC is summarized in **Algorithm 2**.

3) KD-Based Model Training: To account for the varying nature of video content, we design two distinct SC models: one with a high CR, r_{high}, to process static frames, and another with a low CR, r_{low}, to process dynamic frames. Given that high CR can lead to significant semantic loss, we introduce KD during the training process to improve the performance of the high-CR SC model. KD is a transfer learning approach technique that

Algorithm 2: Inference phase of CRSC.

Input: The ith frame $\mathbf{x}_i$, the ith frame's CR r_i.
Output: The reconstructed ith frame $\hat{\mathbf{x}}_i$.
1: Extract the semantic representation $\mathbf{s}_i$ using Eqs. (9)–(12).
2: Perform the semantic encoding to obtain $\mathbf{e}_i$ using Eqs. (13)–(16) with the given CR r_i.
3: Generate the complex vector $\mathbf{c}_i$ by modulating $\mathbf{e}_i$.
4: Receive the complex vector $\mathbf{y}_i$ according to Eq. (3).
5: Reconstruct the frame $\hat{\mathbf{x}}_i$ using Eq. (4).

utilizes a mentor-student framework to transfer knowledge from a well-performing mentor model to a less capable student model. In this context, the low-CR SC model serves as the mentor, while the high-CR SC model acts as the student. The training process for the mentor and student models using KD is illustrated in Fig. 3(a), and is described as follows:

Distill Knowledge from Hard Labels: Both the mentor and student models calculate the loss between their outputs and the corresponding hard labels, which are defined by the specific task at hand [27]. In this case, since the focus is on video frame reconstruction, the hard labels correspond to the original video frames. Let the input frame be $\mathbf{x}_i$; the task losses for the mentor and student models are defined as follows:

$$\mathcal{L}^{\mathrm{m}}_{\mathrm{task}} = \mathrm{MSE}(\mathbf{x}_i, \hat{\mathbf{x}}^{\mathrm{m}}_i), \tag{17}$$

$$\mathcal{L}^{\mathrm{s}}_{\mathrm{task}} = \mathrm{MSE}(\mathbf{x}_i, \hat{\mathbf{x}}^{\mathrm{s}}_i), \tag{18}$$

where $\hat{\mathbf{x}}^{\mathrm{m}}_i$ and $\hat{\mathbf{x}}^{\mathrm{s}}_i$ denote the reconstructed frames generated by the mentor and student models after transmission, respectively. The mean-square error function, $\mathrm{MSE}(\cdot)$, is employed to ensure pixel-level consistency between the original frame and the reconstructed frames. In summary, these task losses provide direct task-specific supervision to guide both the mentor and student models during training.

Distill Knowledge from Soft Labels: In addition to hard label distillation, knowledge transfer between the mentor and student models also occurs through soft labels, such as the semantic encodings produced by each model [27]. Given that incorrect prediction from either the mentor or student model could negatively impact the other during KD, we implement an adaptive approach that adjusts the distillation loss based on the quality of the predicted hard labels (i.e., Eqs. (17) and (18)). The adaptive distillation loss for the student model is expressed as:

$$\mathcal{L}_{\mathrm{KD}} = \frac{\mathrm{KL}(\hat{\mathbf{s}}^{\mathrm{s}}_i, \hat{\mathbf{s}}^{\mathrm{m}}_i)}{\mathcal{L}^{\mathrm{m}}_{\mathrm{task}}}, \tag{19}$$

where $\mathrm{KL}(\cdot)$ represents the Kullback–Leibler divergence, and $\hat{\mathbf{s}}^{\mathrm{s}}_i$ and $\hat{\mathbf{s}}^{\mathrm{m}}_i$ are the semantic representations reconstructed by the student and mentor models, respectively. Specifically, $\hat{\mathbf{s}}^{\mathrm{s}}_i$ is generated from the mentor's semantic encoding $\mathbf{e}^{\mathrm{m}}_i \in \mathbb{R}^{m \cdot r_{\mathrm{low}}}$, while $\hat{\mathbf{s}}^{\mathrm{m}}_i$ is derived from the student's semantic encoding $\mathbf{e}^{\mathrm{s}}_i \in \mathbb{R}^{m \cdot r_{\mathrm{high}}}$.

Both the mentor and student models are trained by minimizing a combination of task and KD losses using the stochastic gradient descent (SGD) optimizer [28]. Here, G denotes the number of training epochs, and $\mathcal{D}$ represents the training dataset. The parameters α_{m} and β_{m} refer to the semantic encoder and decoder in the mentor model, while α_{s} and β_{s} represent the corresponding parameters in the student model. The training process for CRSC is outlined in **Algorithm 3**.

Algorithm 3: Training Phase of CRSC.

Input: Training dataset $\mathcal{D}$.
Output: The mentor model's semantic encoder and decoder parameters, α_{m}, β_{m}, and the student model's semantic encoder and decoder parameters α_{s}, β_{s}.
1: **for** each epoch in G **do**
2: **for** each batch sample in $\mathcal{D}$ **do**
3: Compute task losses $\mathcal{L}^{\mathrm{m}}_{\mathrm{task}}$ and $\mathcal{L}^{\mathrm{s}}_{\mathrm{task}}$ using Eqs. (17) and (18).
4: Compute KD loss $\mathcal{L}_{\mathrm{dis}}$ using Eq. (19).
5: Update α_{m} and β_{m} by minimizing $\mathcal{L}^{\mathrm{m}}_{\mathrm{task}}$ using SGD optimizer.
6: Update α_{s} and β_{s} by minimizing $\mathcal{L}^{\mathrm{s}}_{\mathrm{task}} + \mathcal{L}_{\mathrm{KD}}$ using the SGD optimizer.
7: **end for**
8: **end for**

C. CVSM for Edge Video Sensing

The CRSC model lacks inherent sensing capabilities and cannot autonomously determine whether a given frame is static or dynamic, making it unable to adjust the CR in real time. Traditional radar-based sensing systems require specialized hardware, which significantly increases deployment cost and energy consumption on edge sensors. To overcome this limitation, we propose the CVSM scheme, which leverages vision-based lightweight sensing to intelligently detect scene changes and guide the transmission process of CRSC. Specifically, for each video frame, we employ an ODM, designed based on the YOLO architecture [29], to locate movable targets. Its efficient convolutional backbone and one-stage detection design allow fast inference and compact deployment on resource-limited edge devices. Following this, the SSM is applied to isolate key pixel regions within each frame. The SSM adopts a lightweight attention-based network that focuses on semantically relevant areas while maintaining low computational complexity, ensuring responsiveness for continuous video sensing. Finally, contextual information is compared across frames to detect scene changes, allowing CR adjustments to be dynamically performed within the CRSC system. To further enhance efficiency, quantization is applied to both ODM and SSM, ensuring real-time execution without compromising sensing precision. The overall CVSM process is as follows:

1) ODM-Based Object Location: First, the current frame $\mathbf{x}_i$ is processed to obtain feature maps, denoted as $\mathbf{F}_i \in \mathbb{R}^{H' \times W' \times C'}$, where H', W', and C' are the dimensions of the feature maps. This process can be expressed as:

$$\mathbf{F}_i = \mathrm{Conv}(\mathbf{x}_i), \tag{20}$$

where $\mathrm{Conv}(\cdot)$ represents the convolution operation.

Then, each anchor box is parameterized by its width w_a and height h_a, customized to fit the objects detected in the feature maps $\mathbf{F}_i$. For the jth anchor box, ODM predicts a bounding box offset $\mathbf{t}_j = (t_j^x, t_j^y, t_j^w, t_j^h)$ and a confidence score c_j. The offset $\mathbf{t}_j$ adjusts the anchor box to better fit the detected object, which is formulated as follows:

$$(x_j, y_j, w_j, h_j) = (\sigma(t_j^x) + x_j^g, \sigma(t_j^y) + y_j^g, w_a \mathrm{e}^{t_j^w}, h_a \mathrm{e}^{t_j^h}), \tag{21}$$

where (x_j^g, y_j^g) denotes the grid cell's top-left coordinate, $\sigma(\cdot)$ is the sigmoid activation, and the exponential term ensures adaptive scaling of box size. This compact parameterization enhances the regression flexibility and significantly improves detection precision.

Finally, the set $\mathcal{B}_i$ of all the detected boxes in $\mathbf{x}_i$ is as the output:

$$\mathcal{B}_i = \{(x_j, y_j, w_j, h_j, c_j) | j \in \{1, 2, \ldots, O_i\}\}, \tag{22}$$

where (x_j, y_j, w_j, h_j) represents the coordinates of the jth box, c_j is the confidence score for the detected object, and O_i represents the number of detected objects. Thus, $\mathcal{B}_i$ provides the location of all the movable targets in the current frame $\mathbf{x}_i$.

2) SSM-Based Semantic Segmentation: The frame $\mathbf{x}_i$, along with the location data $\mathcal{B}_i$ (obtained from ODM), is provided as input to SSM, denoted as F_Γ, to perform segmentation [24]:

$$F_\Gamma : (\mathbf{x}_i, \mathcal{B}_i) \to (\mathbf{M}_i, \mathbf{S}_i, \mathbf{L}_i), \tag{23}$$

where $\mathbf{M}_i$ represents the generated binary mask with dimensions (H, W), indicating whether a pixel belongs to the target object (1) or not (0). Additionally, $\mathbf{S}_i$ represents the Intersection over Union (IoU) score, measuring the overlap between the mask and the ground truth annotation, while $\mathbf{L}_i$ provides the class label of the detected object. The resulting mask $\mathbf{M}_i$ is treated as the sensing result for frame $\mathbf{x}_i$.

To detect changes between consecutive frames, we define the difference between the sensing results of two consecutive frames, $\mathbf{M}_{i-1}$ and $\mathbf{M}_i$, as follows:

$$\eta_i = \frac{1}{HW} \sum_{a=1}^{H} \sum_{b=1}^{W} |\mathbf{M}_{i,a,b} - \mathbf{M}_{i-1,a,b}|, \tag{24}$$

where $\mathbf{M}_{i,a,b}$ denotes each pixel in the mask. Since $\mathbf{M}_{i,a,b} \in \{0, 1\}$, $\eta_i < \epsilon$ indicates that there are no changes between frames, classifying $\mathbf{M}_i$ as a static. ϵ represents a threshold, default as $1e-4$ in this paper. If changes are detected, the frame is labeled as dynamic. Based on this classification, the appropriate CR $r_i \in \{r_{\text{low}}, r_{\text{high}}\}$ is assigned to the frame. The workflow for CVSM is outlined in **Algorithm 4**.

3) Quantization for Real-Time Sensing: To ensure that both the ODM and SSM can operate efficiently on edge devices with constrained computational and memory resources, we apply post-training quantization techniques to reduce the model sizes and accelerate inference without significant performance degradation. Specifically, we quantize the weights and activations of both models from 32-bit floating-point precision to 8-bit integers.

Algorithm 4: CVSM.

Input: Detected boxes set $\mathcal{B}_i$, the ith frame $\mathbf{x}_i$.
Output: The ith frame's CR r_i.

```
Obtain the location information B_i for frame x_i using Eqs. (20)–(22).
Based on B_i, obtain the sensing results M_i using Eq. (23).
Calculate the difference η_i between the current and previous frames using Eq. (24).
if η_i == 0 then
    r_i = r_low.
else
    r_i = r_high.
end if
```

Let $\mathbf{w} \in \mathbb{R}^n$ be the original full-precision weights in a given layer. The quantization process maps $\mathbf{w}$ to an 8-bit integer representation $\mathbf{w}_q \in \mathbb{Z}^n$ through the following affine transformation:

$$\mathbf{w}_q = \text{round}\left(\frac{\mathbf{w} - \mu_w}{s_w}\right), \tag{25}$$

$$\mathbf{w} \approx s_w \cdot \mathbf{w}_q + \mu_w, \tag{26}$$

where s_w is the scale factor and μ_w is the zero-point offset that ensures zero is representable in the quantized range. The same transformation is applied to activations during inference. This quantization process reduces both memory footprint and computational overhead.

IV. Experimental Simulations

This section describes the simulation dataset, parameter settings, and evaluation results. The simulations are conducted on a server equipped with an Intel Xeon CPU (2.3 GHz, 256 GB RAM) and two NVIDIA RTX 4090 GPUs (24 GB SGRAM each), leveraging the PyTorch framework to implement the proposed schemes.

A. Simulation Settings

1) Dataset Setup: To evaluate the proposed methods, we employ the VIRAT Video Dataset [22], which contains a variety of surveillance videos captured in different scenarios. The dataset is divided into two primary activity categories: single-object and two-object scenarios, involving both humans and vehicles. During the training phase, we capture one frame per second from each video, resulting in approximately 9,600 RGB images. This dataset is used to train the CRSC model as described in **Algorithm 3**. During inference, two consecutive video clips from distinct scenes (see Fig. 4) are used as test samples and processed according to **Algorithm 1**.

2) Parameters Settings: For the system model, the bandwidth is set to $B = 1$ KHz, and the SNR is varied between 0 dB and 25 dB. In the inference phase, we assess the SC model using different fixed SNR values. During training, the SNR is randomly varied in each forward propagation to improve the

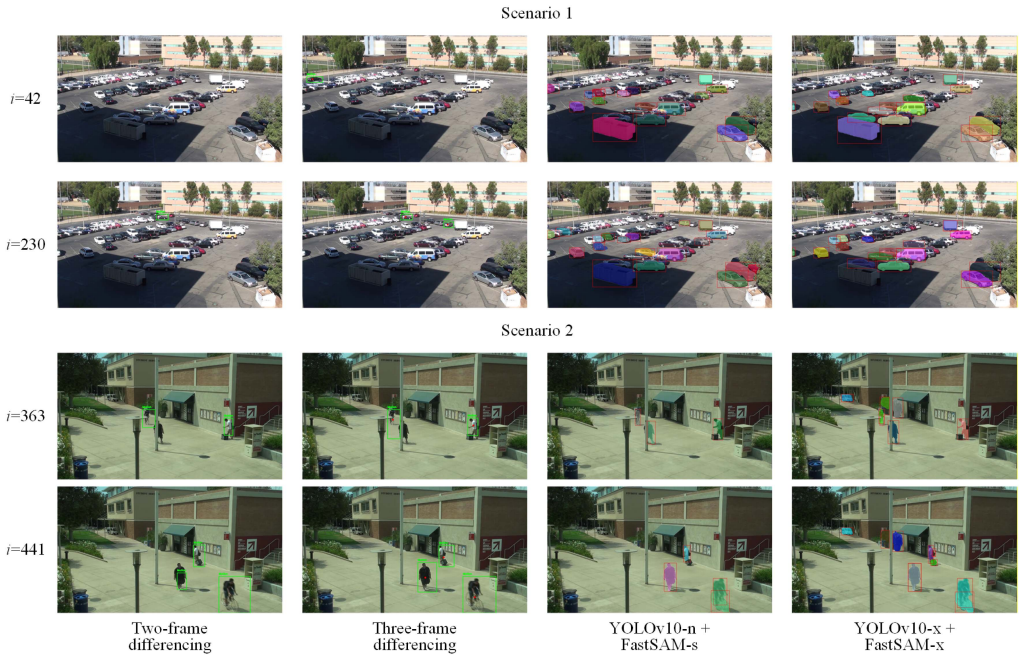


Fig. 4. Visualization of sensing results of different schemes in two scenarios.

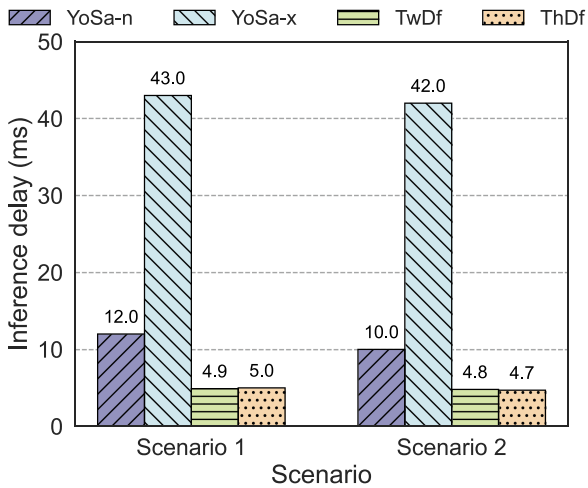


Fig. 5. Inference delay of different sensing schemes.

robustness of the CRSC model against channel noise. In the inference phase, we evaluate the SC model under fixed SNR conditions. Additionally, for the CR, we make the following simplifications: when $r_i = r_{\text{low}}$, the length of the semantic encoding $\mathbf{e}_i$ is set to 256. Conversely, when $r_i = r_{\text{high}}$, the length of $\mathbf{e}_i$ is set to 16. This implies that for static frames, only 6.25% of the data volume required for dynamic frames is transmitted.

3) Comparison Schemes: To assess the efficiency of different perception front-ends for semantic feature extraction, four representative detection or motion estimation methods are evaluated:

- YOLOv10-n+FastSAM-s (YoSa-n) [10], [11]: A lightweight combination of object detection (YOLOv10-n) and segmentation (FastSAM-s), enabling real-time perception with low computational overhead.
- YOLOv10-x+FastSAM-x (YoSa-x) [10], [11]: A more advanced configuration employing larger variants of YOLOv10 and FastSAM to achieve higher detection accuracy at the expense of greater model complexity.
- Two/Three Frame differencing (TwDf/ThDf) [30]: A classical motion-based method that identifies moving regions by calculating pixel-wise differences between consecutive frames. The three-frame version further enhances robustness against noise and background fluctuations.

For the video transmission, several conventional and learning-based schemes are compared to validate the channel robustness of SCCVS:

- H.265+LDPC [15]: A conventional hybrid video compression and channel coding pipeline, where H.265 performs source compression and LDPC codes are applied for channel protection with different coding rates (1/3, 1/2, and 2/3). All transmissions adopt 16QAM modulation to ensure consistent bandwidth utilization across settings.
- DeepJSCC-V [31]: A deep joint source–channel coding model designed for end-to-end image and video transmission, featuring predictive and adaptive semantic representation.

To analyze the contribution of key components in the proposed framework, the following ablated variants are implemented:

- SCCVS (w/o CVSM): In this variant, the CVSM is disabled, and the CRSC model is used for all frames without receiving CR prompts from CVSM, resulting in uniform low-CR transmission.
- SCCVS (w/o KD): KD is excluded during training, and the high-CR and low-CR models are trained independently.

4) Evaluation Metrics: The performance of the proposed SCCVS framework is evaluated from four perspectives: pixel fidelity, perceptual quality, distribution similarity, and

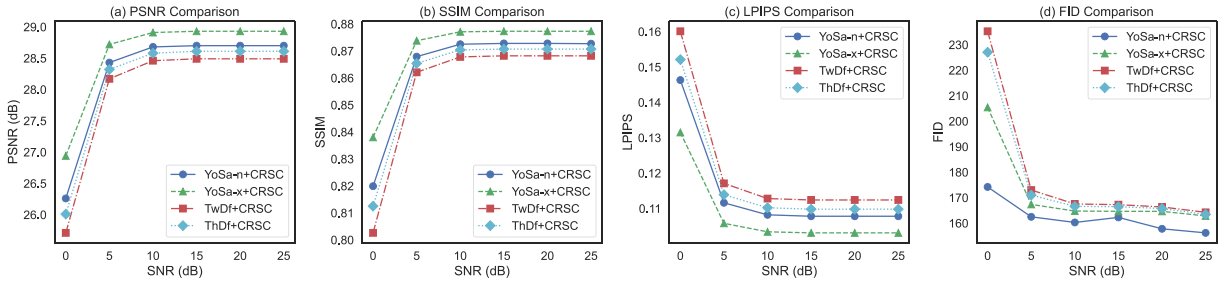


Fig. 6. Performance comparison of CRSC assisted by different sensing schemes.

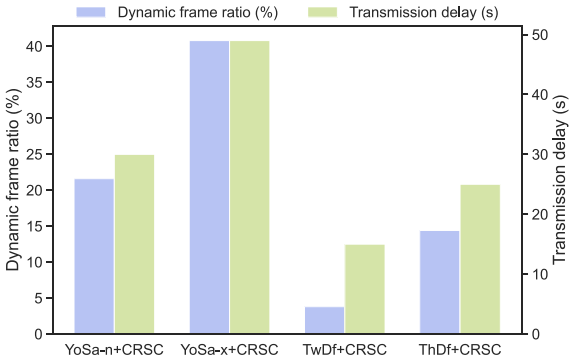


Fig. 7. Transmission delay and identified dynamic frame ratio of CRSC assisted by different sensing schemes.

semantic integrity. Therefore, we consider the following metrics:

- Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM): evaluate the pixel-level fidelity between the reconstructed and reference videos. Higher values of PSNR and SSIM indicate better reconstruction quality and structural preservation.
- Learned Perceptual Image Patch Similarity (LPIPS) [32]: measures perceptual similarity by comparing deep features extracted from pretrained neural networks, providing a more human-aligned perception metric.
- Fréchet Inception Distance (FID) [33]: quantifies the distribution similarity between the reconstructed and original videos in the feature space, serving as an indicator of semantic consistency and visual realism.

B. Evaluation of Scenario Sensing

This subsection aims to verify the effectiveness of the neural network-based visual sensing modules (i.e., the ODM and SSM) in accurately capturing dynamic scene information. We compare multiple sensing schemes in terms of their sensing quality and inference latency to demonstrate their impact on the overall performance of the SC system.

As shown in Fig. 4, we visualize the sensing results of different schemes across two representative scenarios. Specifically, two frames from each video are selected for visualization: the 42nd and 230th frames from the first video and the 363rd and 441st frames from the second video. We observe that traditional methods, such as TwDf and ThDf, can roughly identify motion regions but often fail to detect small or partially occluded targets. In contrast, the neural network-based YoSa-n and YoSa-x methods, which integrate object detection with semantic segmentation models, are capable of capturing much finer-grained changes within the scene. This allows the sensing module to locate key dynamic objects more accurately and even perceive subtle contextual variations. Notably, although YoSa-x achieves the highest detection accuracy, its increased network complexity leads to a considerably higher inference delay, as shown in Fig. 5. YoSa-n, on the other hand, maintains competitive sensing performance while achieving a significantly lower delay, thereby achieving a better trade-off between accuracy and efficiency.

To further evaluate how different sensing schemes affect the overall semantic transmission process, we integrate them into the CRSC model and analyze both reconstruction quality and transmission efficiency. As illustrated in Fig. 6, the results show that YoSa-based sensing notably enhances CRSC performance compared to traditional frame-differencing approaches, achieving higher PSNR, SSIM, and lower perceptual distortion. Fig. 7 demonstrates that the sensing method directly influences both the proportion of frames identified as dynamic and the overall transmission latency. Specifically, while YoSa-x recognizes the highest number of dynamic frames, it incurs the longest delay. TwDf and ThDf, on the other hand, have lower latency but miss most subtle dynamic changes, identifying only a small proportion of dynamic frames (e.g., less than 15%). In contrast, YoSa-n achieves a moderate dynamic frame ratio and maintains a much lower transmission delay, achieving the most favorable balance between perception sensitivity and transmission efficiency.

Therefore, we select the YoSa-n as the CVSM in our SC-CVS framework to achieve an optimal balance between sensing precision and real-time responsiveness.

C. Evaluation of Video Transmission

To verify the effectiveness of the proposed SCCVS framework in video transmission, we compare it with both traditional and AI-based transmission methods.

The comparison is conducted under two distinct scenarios, and the results are illustrated in Fig. 8 and Fig. 9. In the low-SNR regime, where transmission errors are more severe, SC-CVS exhibits overwhelming superiority in terms of both pixel-level fidelity (PSNR and SSIM) and perceptual quality (LPIPS and FID). Compared with traditional H.265+LDPC systems that suffer from severe quality degradation or even complete decoding failure at low SNRs, SCCVS maintains stable and

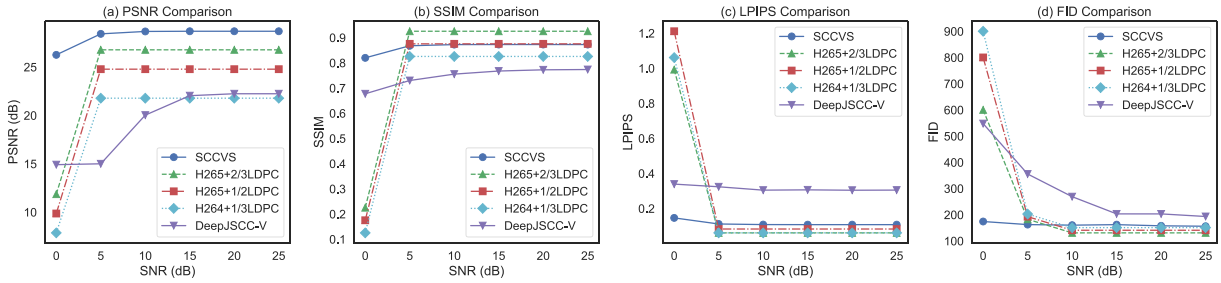


Fig. 8. Comparison of transmission performance of different schemes in scenario 1.

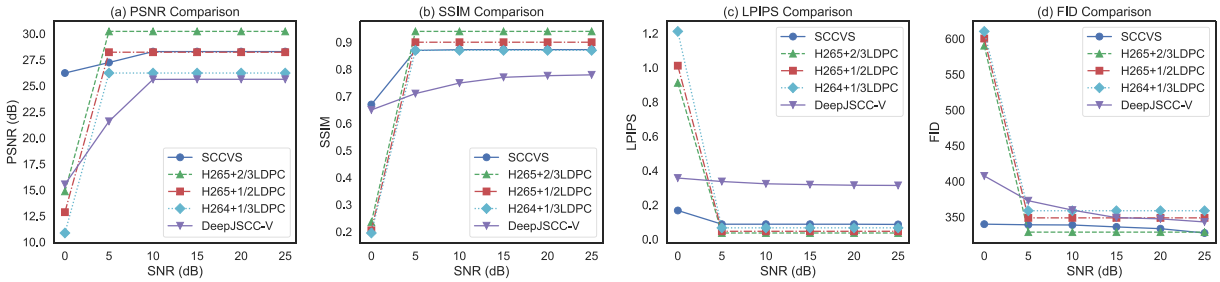


Fig. 9. Comparison of transmission performance of different schemes in scenario 2.

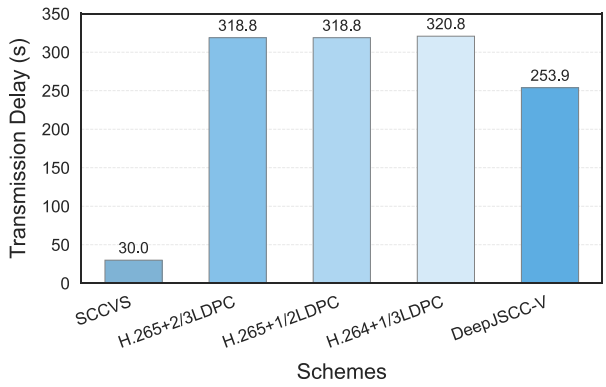


Fig. 10. Comparison of transmission delay of different video transmission schemes.

high-quality video reconstruction. This improvement arises from the semantic-aware visual sensing and adaptive transmission mechanisms, which enable the model to prioritize key information under noisy channel conditions. In the high-SNR regime, SCCVS continues to achieve performance comparable to that of H.265+2/3LDPC while preserving strong robustness and low transmission delay. Although DeepJSCC-V shows competitive results in mid-range SNRs, its overall reconstruction quality and feature consistency remain inferior to SCCVS, especially when evaluated through perceptual metrics such as FID and LPIPS.

Moreover, Fig. 10 shows that SCCVS achieves a significant advantage in efficiency. Specifically, the average transmission delay of SCCVS is only 30 s, which is far lower than that of H.265+LDPC. Even compared with the deep learning-based DeepJSCC-V method, SCCVS still exhibits a remarkable reduction in latency while ensuring higher perceptual fidelity.

Overall, these results demonstrate that the proposed SCCVS framework effectively integrates intelligent sensing and SC to achieve robust and efficient video transmission. It not only surpasses conventional codec-based methods in noisy environments but also approaches their upper-bound performance in high-quality transmission scenarios, demonstrating its strong adaptability and scalability across diverse channel conditions.

D. Ablation Analysis

To investigate the effectiveness of the key components in the proposed SCCVS framework, we conduct ablation experiments under various SNR conditions.

Fig. 11 shows the experimental results on a video of scenario 1. We can observe that the proposed SCCVS maintains stable reconstruction quality across all SNR levels. In comparison, since all frames are transmitted at a low compression ratio, SCCVS (w/o CVSM) achieves slightly higher pixel-level scores. However, this design significantly increases transmission delay, with the total time for sending one test video reaching 95 s, compared to only 30 s for SCCVS. On the other hand, SCCVS (w/o KD) shows a clear performance degradation compared with the full model, especially under low SNRs. This reflects the effectiveness of the KD-based training method in improving semantic representations between high-CR and low-CR models, thereby improving overall reconstruction fidelity and robustness under noisy conditions.

Overall, these results demonstrate that both the CVSM and KD-based training methods play essential roles in achieving adaptive, high-quality, and low-latency semantic video transmission within the SCCVS framework.

E. Evaluation of Lightweight

To verify the effectiveness of the proposed lightweight design, we analyze the computational complexity and parameter size of the modules deployed on the edge device. Specifically, this

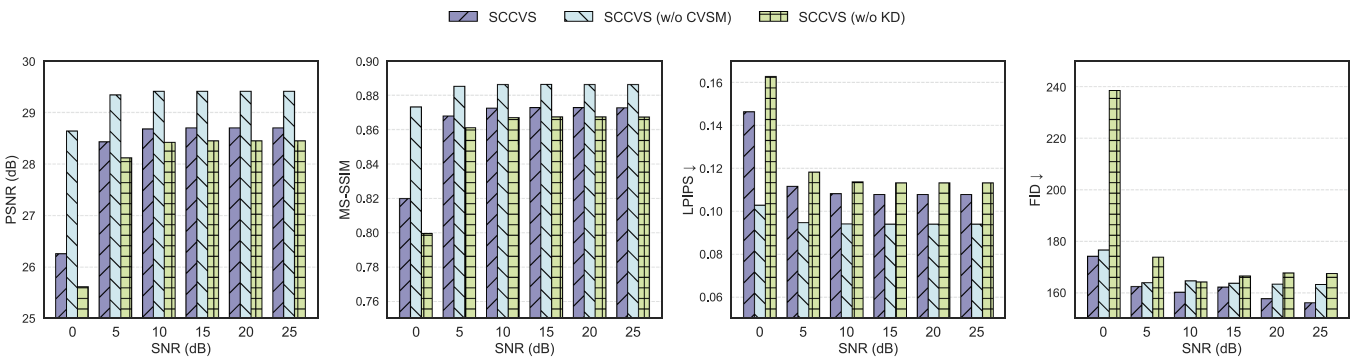


Fig. 11. Performance comparison results of ablation experiments.

TABLE II
MODEL COMPLEXITY AND INFERENCE DELAY OF SCCVS MODULES

Module	CVSM		CRSC			Total
	ODM	SSM	Sem. Enc.	Ch. Codec	Sem. Dec.	
Params (M)	2.29	11.78	13.14	1.18	111.75	140.14
GFLOPs	6.7	42.4	2.6	0.007	4.6	56.3
Delay w/o TensorRT (ms)	43.21	23.80	92.10	1.45	-	160.56
Delay with TensorRT (ms)	1.09	1.08	11.10	0.12	-	13.39

evaluation includes the ODM, SSM, and the major components of the CRSC system, i.e., the lightweight ViT-based semantic encoder and KAN-based channel codec. These modules are deployed on an NVIDIA Jetson Orin NX Super platform, which integrates an 8-core Arm Cortex-A78AE v8.2 64-bit CPU (2 MB L2 + 4 MB L3 cache) and an NVIDIA Ampere GPU with a maximum frequency of 1,173 MHz, offering up to 157 TOPS of AI computing power and 16 GB LPDDR5 memory [34]. Additionally, during deployment, TensorRT-based optimization and inference acceleration [35] are employed to fully exploit the parallel computing capability of the embedded GPU, thereby significantly reducing latency and improving runtime efficiency. This setup provides a representative environment for evaluating device performance under resource-constrained edge conditions.

In Table II, we summarize the parameter size, GFLOPs, and inference delay for each module. Specifically, the edge-side components have a parameter size of 140.14 M and a computational complexity of 56.3 GFLOPs. Without TensorRT optimization, the end-to-end inference delay reaches 160.56 ms/frame. In contrast, with TensorRT acceleration on the Jetson Orin NX Super, it is reduced to approximately 13.39 ms/frame, demonstrating a speed-up of over 10×. These results indicate that the lightweight ViT-based semantic encoder achieves an excellent balance between semantic representation capacity and computational efficiency, thanks to its compact transformer architecture and adaptive token aggregation strategy. The KAN-based channel codec further enhances efficiency by replacing conventional MLP layers with functional KANs, significantly reducing parameter redundancy while preserving expressive power for robust channel adaptation. Although the CVSM module's SSM contains a relatively larger number of parameters due to its dense feature extraction process, its inference delay still satisfies real-time requirements for visual perception tasks on edge devices.

In summary, the experimental findings verify that the proposed SCCVS achieves a well-optimized trade-off between accuracy and efficiency. The introduction of TensorRT-based acceleration dramatically reduces inference latency, making the lightweight ViT-based and KAN-based components highly feasible for large-scale deployment in edge-side intelligent communication scenarios.

V. CONCLUSION

To mitigate the high spectrum resource demands associated with vision sensors transmitting edge video, we propose the SCCVS framework. This framework introduces a CRSC model that intelligently adjusts the CRs of video frames based on real-time sensing results, optimizing the balance between compression efficiency and semantic fidelity. Additionally, the CVSM scheme is incorporated, leveraging CV techniques to detect changes in the edge scenes and assess the contextual importance of each frame. This enables CVSM to guide the CRSC model in applying lower CRs to dynamic frames while assigning higher CRs to static frames. Furthermore, both the CRSC and CVSM models are designed with lightweight architectures, making the framework particularly suitable for resource-constrained sensors in real-world edge applications. Experimental results show that the proposed SCCVS framework achieves approximately 70% higher transmission accuracy and 89% lower latency compared with baselines. When deployed on an NVIDIA Jetson Orin NX Super with TensorRT acceleration, it attains a real-time inference speed of 14 ms per frame, demonstrating its efficiency and suitability for edge deployment.

Future research will explore multimodal sensing (e.g., combining vision and radar signals) to further enhance scene understanding and compression decision-making. We also aim to incorporate online learning and model adaptation techniques to ensure long-term stability and adaptability of the CRSC and CVSM models in continuously changing environments. Finally, extending SCCVS to support collaborative sensing across multiple edge nodes, under a federated or distributed framework, will be another key direction, allowing for more intelligent and cooperative video compression strategies in large-scale sensor networks.

REFERENCES

[1] P. Gallo, S. Pongnumkul, and U. Quoc Nguyen, "Blocksee: Blockchain for iot video surveillance in smart cities," in *Proc. IEEE Int. Conf. Environ. Elect. Eng. IEEE Ind. Commercial Power Syst. Europe*, 2018, pp. 1–6.

[2] P. Sivalakshmi, U. Kavitha, R. Usha, O. Pattanaik, S. Maniraj, and C. Srinivasan, "Smart retail store surveillance and security with cloud-powered video analytics and transfer learning algorithms," in *Proc. 2nd Int. Conf. Intell. Cyber Phys. Syst. Internet Things*, 2024, pp. 242–247.

[3] M. Kashyap, Z. Naaz, N. Bansal, and V. Sharma, "Performance evaluation of iot architecture for heterogeneous networks," in *Proc. 14th Int. Conf. Comput. Commun. Netw. Technol.*, 2023, pp. 1–8.

[4] D. Xu et al., "Edge intelligence: Empowering intelligence to the edge of network," *Proc. IEEE Proc. IRE*, vol. 109, no. 11, pp. 1778–1837, Nov. 2021.

[5] F. Jiang et al., "Large AI model-based semantic communications," *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 68–75, Jun. 2024.

[6] F. Jiang et al., "Large AI model empowered multimodal semantic communications," *IEEE Commun. Mag.*, vol. 63, no. 1, pp. 76–82, Jan. 2025.

[7] X. Luo, H.-H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 210–219, Feb. 2022.

[8] A. Liu et al., "A survey on fundamental limits of integrated sensing and communication," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 994–1034, Secondquarter 2022.

[9] C. Luo, J. Hu, L. Xiang, and K. Yang, "Reconfigurable intelligent sensing surface aided wireless powered communication networks: A sensing-then-reflecting approach," *IEEE Trans. Commun.*, vol. 72, no. 3, pp. 1835–1848, Mar. 2024.

[10] A. Wang et al., "Yolov10: Real-time end-to-end object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, A. Globerson et al., Eds., 2024, vol. 37, pp. 107984–108011. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/file/c34ddd05eb089991f06f3c5dc36836e0-Paper-Conference.pdf

[11] X. Zhao et al., "Fast segment anything," 2023, *arXiv:2306.12156*.

[12] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Trans. Signal Process.*, vol. 69, pp. 2663–2675, 2021.

[13] Y. Wu, P. Liu, Y. Gao, and K. Jia, "Medical ultrasound video coding with h.265/hevc based on roi extraction," *PLoS One*, vol. 11, no. 11, pp. 1–13, 2016, doi:10.1371/journal.pone.0165698.

[14] A. Djelouah, J. Campos, S. Schaub-Meyer, and C. Schroers, "Neural inter-frame compression for video coding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 6421–6429.

[15] S. Wang et al., "Wireless deep video semantic transmission," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 1, pp. 214–229, Jan. 2023.

[16] P. Hu, J. Im, Z. Asgar, and S. Katti, "Starfish: Resilient image compression for aiot cameras," in *Proc. 18th Conf. Embedded Netw. Sens. Syst.*, New York, NY, USA:Association for Computing Machinery, 2020, pp. 395–408, doi: 10.1145/3384419.3430769.

[17] K. Du et al., "Server-driven video streaming for deep learning inference," in *Proc. Annu. Conf. ACM Special Int. Group Data Commun. Appli., Techno., Architectures, Protoc. Comput. Commun., Ser.*, New York, NY, USA: Association for Computing Machinery, 2020, pp. 557–570, doi: 10.1145/3387514.3405887.

[18] X. Xiao, Y. Zuo, M. Yan, W. Wang, J. He, and Q. Zhang, "Task-oriented video compressive streaming for real-time semantic segmentation," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 14396–14413, Dec. 2024.

[19] K. Du, Q. Zhang, A. Arapin, H. Wang, Z. Xia, and J. Jiang, "Accmpeg: Optimizing video encoding for accurate video analytics," in *Proc. Mach. Learn. Syst.*, D. Marculescu, Y. Chi, and C. Wu, Eds., 2022, vol. 4, pp. 450–466. [Online]. Available: https://proceedings.mlsys.org/paper_files/paper/2022/file/853f7b3615411c82a2ae439ab8c4c96e-Paper.pdf

[20] W. Wang, B. Wang, L. Zhang, and H. Huang, "Sensitivity-aware spatial quality adaptation for live video analytics," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 8, pp. 2474–2484, Aug. 2022.

[21] Z. Liu et al., "KAN: Kolmogorov–Arnold networks," in *Proc. 13th Int. Conf. Learn. Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=Ozo7qJ5vZi

[22] S. Oh et al., "A large-scale benchmark dataset for event recognition in surveillance video," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3153–3160.

[23] D. B. Kurka and D. Gündüz, "DeepJSCC-f: Deep joint source-channel coding of images with feedback," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 178–193, May 2020.

[24] Y. Peng et al., "Large generative model assisted 3D semantic communication," *IEEE Trans. Wireless Commun.*, early access, Oct. 29 2025, doi: 10.1109/TWC.2025.3624573.

[25] Y. Peng, F. Jiang, L. Dong, K. Wang, and K. Yang, "Personalized federated learning for gai-assisted semantic communications," *IEEE Trans. Cogn. Commun. Netw.*, early access, Jul. 8 2025, doi: 10.1109/TCCN.2025.3586904.

[26] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. W. Lau, "Biformer: Vision transformer with bi-level routing attention," in *Proc. IEEE/CVF Conf. Comput. Vis. pattern Recognit.*, 2023, pp. 10323–10333.

[27] A. Jafari, M. Rezagholizadeh, P. Sharma, and A. Ghodsi, "Annealing knowledge distillation," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics: Main Volume*, Apr. 2021, pp. 2493–2504. [Online]. Available: https://aclanthology.org/2021.eacl-main.212/

[28] Z. Zhang, "Improved adam optimizer for deep neural networks," in *Proc. 2018 IEEE/ACM 26th Int. Symp. Qual. Serv.*, 2018, pp. 1–2.

[29] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of YOLO algorithm developments," *Procedia Comput. Sci.*, vol. 199, pp. 1066–1073, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922001363

[30] S. S. Sengar and S. Mukhopadhyay, "A novel method for moving object detection based on block based frame differencing," in *Proc. 3rd Int. Conf. Recent Adv. Inf. Technol.*, 2016, pp. 467–472.

[31] W. Zhang, H. Zhang, H. Ma, H. Shao, N. Wang, and V. C. Leung, "Predictive and adaptive deep coding for wireless image transmission in semantic communication," *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5486–5501, Aug. 2023.

[32] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[33] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo, "Reliable fidelity and diversity metrics for generative models," in *Proc. 37th Int. Conf. Mach. Learn.*, H. D. III and A. Singh, Eds., 13–18 Jul. 2020, vol. 119, pp. 7176–7185. [Online]. Available: https://proceedings.mlr.press/v119/naeem20a.html

[34] J. Yuan et al., "Mobile foundation model as firmware," in *Proc. 30th Annu. Int. Conf. Mobile Comput. Netw.*, New York, NY, USA: Association for Computing Machinery, 2024, pp. 279–295, doi: 10.1145/3636534.3649361.

[35] Y. Zhou, Z. Guo, Z. Dong, and K. Yang, "Multi-accelerator neural network inference via tensorrt in heterogeneous embedded systems," in *Proc. IEEE 48th Annu. Comput., Softw., Appl. Conf.*, 2024, pp. 463–472.



Yubo Peng (Graduate Student Member, IEEE) received the BS and MS degrees from Hunan Normal University, Changsha, China, in 2019 and 2024, respectively. He is currently working toward the doctor's degree with the School of Intelligent Software and Engineering, Nanjing University. His research interests include multimodal ISAC and semantic communications.



Luping Xiang (Senior Member, IEEE) received the BEng (Hons.) degree from Xiamen University, China, in 2015, and the PhD degree from the University of Southampton, in 2020. From 2020 to 2021, he was a research fellow with the Next Generation Wireless Group, University of Southampton. In 2021, he was with the University of Electronic Science and Technology of China as a faculty member, and in 2024, he was with Nanjing University as an assistant professor. His research interests include native intelligence in wireless communication, end-to-end transmission technology, computer vision, and integrated sensing and communication transmission.



Kun Yang (Fellow, IEEE) received the PhD degree from the Department of Electronic & Electrical Engineering of University College London, U.K. He is currently a chair professor with Nanjing University and also an affiliated professor with the University of Essex. He has authored or coauthored more than 500 papers and filed 50 patents. His research interests include wireless networks and communications, communication-computing cooperation, and new AI (artificial intelligence) for wireless. He serves on the editorial boards of a number of IEEE journals, such as IEEE WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON NANOBIOSCIENCE. He is a deputy editor-in-chief of IET Smart Cities Journal. He has been a Judge of the GSMA GLOMO Award at World Mobile Congress – Barcelona since 2019. He was a distinguished lecturer of IEEE ComSoc from 2020 to 2021, a recipient of 2024 IET Achievement Medals, and a recipient of 2024 IEEE CommSoft TC's Technical Achievement Award. He is a member of Academia Europaea, fellow of IET, and a distinguished member



Kezhi Wang (Senior Member, IEEE) received the PhD degree in engineering from the University of Warwick, U.K. He was with the University of Essex and Northumbria University, U.K. He is currently a professor with the Department of Computer Science, Brunel University London, U.K. His research interests include wireless communications, mobile edge computing, and machine learning.



Mérouane Debbah (Fellow, IEEE) is currently a professor with the Khalifa University of Science and Technology, Abu Dhabi, UAE, where he is the founding director of KU 6 G Research Center. He is a frequent keynote speaker at international conferences in the fields of telecommunications and artificial intelligence. His research interests include intersection of fundamental mathematics, algorithms, statistics, information theory, and communication sciences, with a particular focus on random matrix theory and learning algorithms. In communications, he has played a key role in the development of small cells (4G), massive MIMO (5G), and large intelligent surface (6G) technologies. In artificial intelligence, he is well known for his contributions to large language models, distributed AI systems for networks, and semantic communications. He was the recipient of numerous prestigious honors, prizes, and more than 50 IEEE Best Paper Awards for his contributions to both fields. He is a fellow of WWRF, EURASIP, AAIA, Institut Louis Bachelier, and AIIA, and a Membre Émérite of SEE. He is the Chair of IEEE Emerging Technology Initiative on Large Generative AI Models in Telecom (GenAINet) and member of the Marconi Prize Selection Advisory Committee.