IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 May 2025, accepted 5 June 2025, date of publication 10 June 2025, date of current version 26 June 2025.

*Digital Object Identifier 10.1109/ACCESS.2025.3578292*

RESEARCH ARTICLE

# Image Coding for Object Recognition Tasks Based on Contour Feature Learning With Flexible Object Selection

**TAKAHIRO SHINDO[1], (Graduate Student Member, IEEE), TAIJU WATANABE[1], YUI TATSUMI[2], AND HIROSHI WATANABE[1,2], (Life Member, IEEE)**

[1]Graduate School of Fundamental Science and Engineering, Waseda University, Shinjuku, Tokyo 169-8555, Japan
[2]School of Fundamental Science and Engineering, Waseda University, Shinjuku, Tokyo 169-8555, Japan

Corresponding author: Takahiro Shindo (taka_s0265@ruri.waseda.jp)

This work was supported by the National Institute of Information and Communications Technology (NICT), Japan, under Grant JPJ012368C05101.

**ABSTRACT** The consumption of image data by machines is rapidly increasing due to the growing adoption of image recognition technologies. This trend has accelerated research in image compression techniques tailored for machine processing. This emerging field, known as Image Coding for Machines (ICM), has gained significant attention in recent years. In particular, ICM is increasingly seen as essential for collaborative systems between edge devices and cloud AI. Since large AI models are challenging to deploy on edge devices, cloud AI services are made available to edge users, who can utilize them by transmitting images to the cloud. Cloud AI is expected to handle various tasks, including image generation and image recognition, with the latter being especially valuable for video and image analysis. Given its utility, image recognition models are anticipated to replace human analysts in applications such as farm and traffic monitoring. Moreover, since recognition models require only a small fraction of the total image data, developing specialized image compression methods for recognition can significantly enhance communication efficiency. However, applying conventional ICM methods to edge-cloud systems presents challenges, such as increased computational load on edge devices and limited versatility. In this paper, we address these challenges by proposing two novel image compression methods—SA-ICM and ST-ICM—designed for recognition models. These methods focus on preserving object contours within images while maintaining compatibility with various recognition models, without adding computational overhead to edge devices. Through experimental evaluations, we demonstrate the versatility and effectiveness of our proposed methods by comparing them with conventional approaches.

**INDEX TERMS** Edge-cloud system, image coding for machines, image coding for image recognition, image compression, learned image compression.

## I. INTRODUCTION

Image compression technology refers to techniques that reduce the amount of data needed to represent an image by eliminating redundancy in the image information. This technology is crucial for efficient image data storage and transfer while preserving image quality as much as possible. Due to its significance, numerous standardization bodies have been actively involved in the research and development of image compression methods. For instance, the Joint Photographic Experts Group introduced the first JPEG [1] in 1992 and has continuously maintained and updated it. Over the years, this group has also established other standards, such as JPEG 2000 [2], JPEG-XL [3], and JPEG-AI [4], significantly contributing to the widespread adoption of image compression technology. Similarly, the Moving Picture Experts Group (MPEG) primarily focuses on video compression standards but has also contributed to image compression

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

technologies [5], [6]. Video compression typically involves two types of compression: intra-frame (Intra) and inter-frame (Inter). Intra-compression is a technique used in video compression where each frame is compressed independently, without referring to other frames. It is equivalent to image compression, since intra-compression treats each video frame as a standalone image. As a result, MPEG has been involved in standardizing image compression technologies, developing formats like HEVC-Intra [7] and VVC-Intra [8]. Many of the image compression standards established by these organizations aim to represent images more faithfully to the original ones while minimizing the required data. In technical terms, these standards strive to achieve a high Peak Signal-to-Noise Ratio (PSNR) at low bit rates [9]. Compression technologies developed under these aims are widely applied in a variety of applications and enable smooth management and sharing of image data.

Conversely, limiting the intended use of images opens the door to exploring compression methods with improved performance. Extracting task-specific image information enables the elimination of redundant content, thereby achieving a more efficient and compact representation. In such cases, faithfully reproducing the original image becomes less critical, primarily focusing on encoding only the essential information as small as possible. Compression techniques tailored to typical image data applications are in high demand and often necessitate standardization. In this context, a particularly noteworthy research area is Image Coding for Machines (ICM) [10], [11], [12], [13]. Recent advancements in AI have led to remarkable improvements in the performance of image recognition models. These AI advancements catalyzed the utilization of image recognition models for image analysis, leading to increased image data processing requests for machines. This trend has created a pressing need for research into ICM to address these demands. Standards organizations have acknowledged this need and initiated efforts to standardize image compression methods designed for machine consumption. For example, the white paper associated with the JPEG-AI [4] standardization process highlights the goal of developing an image compression method applicable to machine tasks [14]. Similarly, MPEG is working on the standardization of Video Coding for Machines (VCM) and is actively exploring the technology through calls for proposals [15], [16], [17], [18], [19].

Despite significant research efforts, ICM methods have yet to be standardized or widely adopted in practice, and several challenges remain. The first challenge lies in ensuring the general applicability of ICM methods. Many of the currently proposed approaches are optimized for specific recognition models, limiting their versatility and preventing their use with other models [20]. This lack of flexibility is particularly problematic in a system of Collaborative Intelligence between edge devices and cloud AI. In these scenarios, images captured at the edge are often processed and analyzed by various recognition models hosted in the cloud server [21], [22], [23], [24]. Since the cloud hosts a diverse array of recognition models, edge devices need image compression methods that are robust and adaptable to changes in these models. The second challenge is to limit the computational complexity of the encoder. In many practical use cases, it is assumed that the encoder for ICM methods will be implemented on edge devices, while the decoder will operate in the cloud. Given the limited computing power of edge devices, it is crucial to keep the computational load of the encoder low [25]. Designs that impose a heavy processing burden on image encoding are unsuitable for ICM methods in such contexts. These challenges stem from the practical considerations of real-world applications. Research that focuses solely on improving the compression performance of ICM methods risks overlooking these critical implementation constraints.

In this paper, we propose a novel ICM method designed to address these challenges in real-world use cases. The method leverages Learned Image Compression (LIC), a neural network-based image compression model [26], [27], [28]. By training the LIC model to learn the shape of objects, a compression model is developed that accurately preserves the contours of objects within images. This approach allows the LIC model to discard other parts of the image, thereby significantly reducing the amount of image data. At the same time, since the shape and position of objects are faithfully retained, the compressed images remain suitable for a wide range of recognition tasks. The proposed model does not impose any additional computational burden compared to standard LIC models, nor is it tailored to specific recognition models. In other words, it can be seamlessly adopted in target applications while addressing the aforementioned challenges. Moreover, the proposed method offers superior compression performance, achieving higher efficiency compared to existing ICM methods. Through experimental evaluation, we demonstrate that the proposed method is applicable to a variety of recognition models. We also measure its compression performance and compare it against other ICM methods to validate its effectiveness.

## II. RELATED WORKS

### A. ICM IN EDGE-CLOUD SYSTEM

The collaboration between edge devices and cloud AI enables users to leverage state-of-the-art large models [21], [22], [23], [24]. Thanks to the cloud's vast data capacity and seamless update capabilities, a wide range of new image recognition models can be hosted and accessed, regardless of their model size. To use these models, images must be transmitted from edge devices to the cloud. In this process, image coding methods are expected to play a critical role [29]. However, as discussed in Chapter 1, ICM method faces challenges in terms of versatility and computational cost when implemented in edge-cloud systems. To ensure easy access to large models for everyone, research is needed to address these challenges in ICM methods while striving to enhance compression performance.

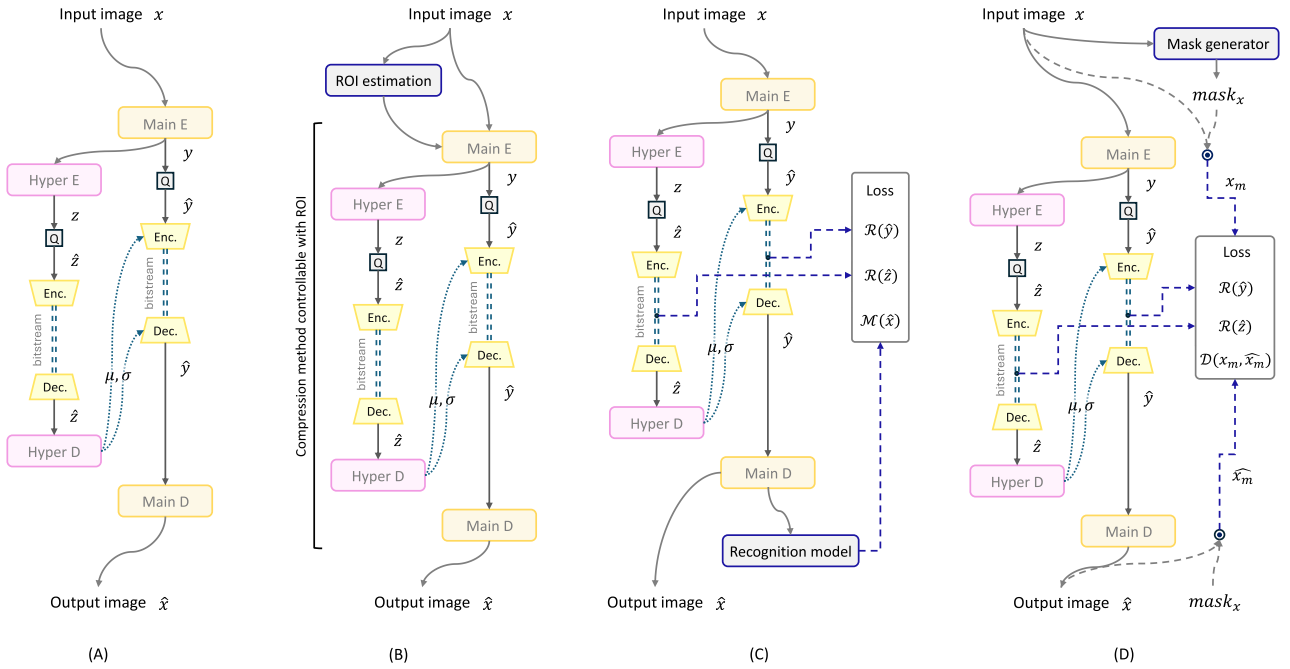


**FIGURE 1.** Image processing flow for each image compression method. (A): Standard image compression method for reconstruction of the entire image, as well as Task-Optimization-based ICM and Region-Learning-based ICM during the inference stage. (B): ROI-based ICM, (C): Task-Optimization-based ICM during the training stage, (D): Region-Learning-based ICM during the training stage.

As a first step, we investigate the applicability of existing ICM methods in the edge-cloud system. Existing ICM approaches can be broadly classified into three categories: ROI-based ICM, Task-Optimized ICM, and Region-Learning-based ICM. Each of these methods offers specific advantages and disadvantages, with their effectiveness varying depending on the use case. In this chapter, we explore related research in these three categories, examining their application to edge-cloud systems in Sections II-A, II-B, and II-C, respectively.

### B. ROI-BASED ICM

Region-of-Interest (ROI) is a concept for defining specific regions in an image that are of particular importance. Several ICM methods, known as ROI-based ICM methods, have been proposed based on this concept, as illustrated in Fig. 1 (B). In these methods, the core assumption is that object regions within an image are crucial for the recognition model. Accordingly, these methods prioritize bit allocation to these regions by utilizing an ROI map [30], [31], [32], [33]. This approach allows object regions to be faithfully represented while disregarding less relevant parts of the image, thereby creating an efficient image representation tailored for machine use. ROI-based bit allocation modules have been integrated into certain Learned Image Compression (LIC) models and standards designed for human perception, such as HEVC and VVC [34], [35]. By leveraging this module, these compression methods can be extended to image compression specifically designed for machines. Moreover, since general detection and classification models tend to focus on object regions within an image, ROI-based ICM methods can serve as effective compression techniques across various recognition models.

However, ROI-based ICM methods are not well-suited for application in edge-cloud systems. This limitation arises from the necessity of ROI estimation on the edge side. Since ROI estimation plays a crucial role in determining the compression performance of ROI-based methods, deploying an accurate ROI estimation model—typically an object detection model—on edge devices is required. Implementing large models on edge devices is often challenging due to constraints in memory capacity and computational power. In fact, these limitations are a key reason for leveraging cloud AI in the first place. Therefore, assuming that ROI estimation models can be deployed on edge devices contradicts this premise and should be avoided. In summary, ROI-based ICM methods are not an ideal choice for facilitating the connection between edge devices and cloud AI.

### C. TASK-OPTIMIZATION-BASED ICM

The ICM method, which optimizes the LIC model based on the recognition model, provides an effective approach to obtaining image representations tailored for machines [36]. The core concept of this method is to design a loss function that preserves the accuracy of the recognition task during the training process of the LIC model. [37], [38], [39], [40]. Initially, the loss function of the LIC model, designed for the faithful reconstruction of an image, is expressed as follows:

$$\mathcal{L} = \mathcal{R}(\hat{\boldsymbol{y}}) + \mathcal{R}(\hat{\boldsymbol{z}}) + \lambda \cdot mse(\boldsymbol{x}, \hat{\boldsymbol{x}}). \quad (1)$$

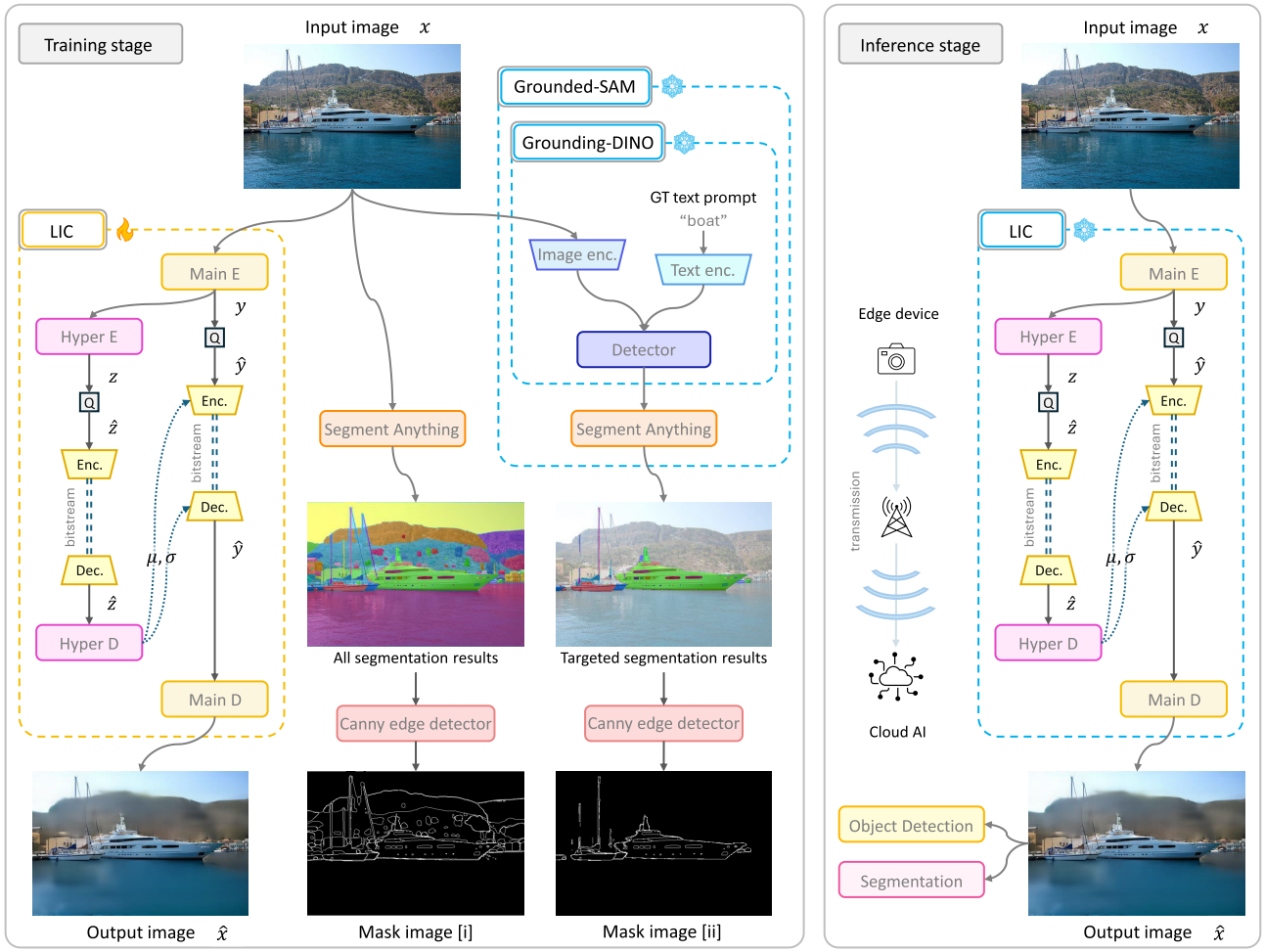


**FIGURE 2.** **Image processing flow of the proposed methods during the training and inference stages. Left: Training stage. Mask image [i] and Mask image [ii] are binary masks that are utilized in SA-ICM and ST-ICM training processes, respectively. These binary masks are generated using Segment Anything and Grounded-SAM with fixed weights. Right: Inference stage. In this stage, no binary mask is required, only a pre-trained compression model is run to decode the image for image recognition.**

In (1), $\boldsymbol{x}$ represents the original image, and $\hat{\boldsymbol{x}}$ represents the output image. $\hat{\boldsymbol{y}}$ is a quantized feature of $\boldsymbol{x}$, and $\hat{\boldsymbol{z}}$ is a latent feature to condition $\hat{\boldsymbol{y}}$. $\mathcal{R}(\hat{\boldsymbol{y}})$ and $\mathcal{R}(\hat{\boldsymbol{z}})$ are the bitrates of $\hat{\boldsymbol{y}}$ and $\hat{\boldsymbol{z}}$, respectively. *mse* represents the mean squared error function and $\lambda$ is a constant to control the rate [26], [27]. The roles of these variables in the LIC model are shown in Fig. 1 (A). Task-Optimization-based ICM modifies the original loss function by incorporating the impact of recognition accuracy. The updated loss function is expressed as follows:

$$\mathcal{L} = \mathcal{R}(\hat{\boldsymbol{y}}) + \mathcal{R}(\hat{\boldsymbol{z}}) + \lambda_1 \cdot mse(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda_2 \cdot \mathcal{M}(\hat{\boldsymbol{x}}). \quad (2)$$

In (2), $\mathcal{R}$, *mse*, $\hat{\boldsymbol{y}}$, $\hat{\boldsymbol{z}}$, $\boldsymbol{x}$, and $\hat{\boldsymbol{x}}$ have the same meaning as those functions, variables in (1). $\mathcal{M}(\hat{\boldsymbol{x}})$ is the loss to improve recognition accuracy, and is calculated by inputting the decoded image $\hat{\boldsymbol{x}}$ into the recognition model. $\lambda_1$ and $\lambda_2$ are constants to control the rate [10]. The roles of these variables in the Task-Optimization-based ICM method are shown in Fig. 1 (C). Optimizing the LIC model based on recognition accuracy alone is challenging, making it difficult to achieve effective learning. To address this, distortion loss ($mse(\boldsymbol{x}, \hat{\boldsymbol{x}})$) is incorporated to simplify the learning process. Additionally, in some cases, the following equation is used as a loss function, leveraging the feature extractor to further mitigate the learning difficulty:

$$\mathcal{L} = \mathcal{R}(\hat{\boldsymbol{y}}) + \mathcal{R}(\hat{\boldsymbol{z}}) + \lambda_1 \cdot mse(\boldsymbol{x}, \hat{\boldsymbol{x}}) + \lambda_2 \cdot mse(\mathcal{F}(\boldsymbol{x}), \mathcal{F}(\hat{\boldsymbol{x}})). \quad (3)$$

In (3), each variable or function has the same meaning as in (1) and (2). $\mathcal{F}$ represents the feature extractor of the recognition model. Generally, the backbone of the recognition model is exploited as a feature extractor. Instead of directly targeting recognition accuracy, the LIC model is optimized using intermediate features extracted from the recognition model [11], [20]. Leveraging features from the shallow layers of the model significantly reduces the difficulty of training the LIC model. The Task-Optimization-based ICM method incorporates these advancements to optimize the LIC model specifically for the recognition model. As a result, the ICM

method delivers high compression performance as an image compression technique tailored to the specific recognition model employed during the optimization process.

The primary limitation of this ICM approach is that it can only be applied to a specific recognition model. This is because a specific recognition model is introduced to optimize the compression model. If only a single recognition model is intended for use, this ICM approach is an excellent choice. However, in scenarios where multiple recognition models are anticipated, a more robust ICM approach capable of adapting to changes in recognition models is necessary. In such cases, the Task-Optimization-based ICM method falls short of meeting the required flexibility.

### D. REGION-LEARNING-BASED ICM

The image data required by image recognition models typically corresponds to specific regions within the image. To leverage this characteristic, the Region-Learning-based ICM method is employed to train the LIC model, focusing on compressing only the regions of the image that are necessary for the recognition model. [41], [42]. The training process of this ICM method is shown in Fig. 1 (D). The loss function is defined by the following equation:

$$\mathcal{L} = \mathcal{R}(\hat{\boldsymbol{y}}) + \mathcal{R}(\hat{\boldsymbol{z}}) + \lambda \cdot mse(\boldsymbol{x} \odot mask_x, \hat{\boldsymbol{x}} \odot mask_x). \quad (4)$$

In (4), $mask_x$ is the binary mask image corresponding to image $\boldsymbol{x}$. This method faithfully decodes only the mask regions used during the training process. Regions outside the mask are not learned, making them undecodable and thereby reducing the bit rate. While this ICM method resembles the ROI-based ICM method, it differs in that it does not impose additional computational costs during testing. The mask image is computed only during training and is not required during testing. Therefore, the processing flow during the inference stage of the LIC model is shown in Fig. 1 (A). Additionally, since the LIC model is not optimized for a specific recognition model, it can serve as an image compression method compatible with various recognition models. A notable limitation of the Region-Learning-based ICM method is the need for fine-tuning the recognition model. Recognition accuracy can be enhanced by further training pre-trained recognition models with decoded images.

This limitation can be easily addressed in a cooperative system between cloud AI and edge devices. Fine-tuned recognition models are prepared in the cloud, allowing users to freely select and utilize the desired recognition models. The Region-Learning-based ICM method can also be applied to new recognition models, ensuring continuous usability and adaptability.

## III. PROPOSED METHOD

### A. OVERVIEW OF THE PROPOSED METHOD

We propose an image compression model designed to accurately decode the shape and position of objects within an image. This proposal is based on the hypothesis that object shape information is crucial for image recognition models. In contrast, detailed object textures are deemed unnecessary for recognition tasks. By discarding such details, the proposed model aims to achieve a lower bit rate. To implement the proposed compression model, we adopt the Region-Learning-based ICM method, selected from the three ICM approaches mentioned earlier, due to its strong suitability for edge-cloud systems. A binary mask is generated to identify background edges and object contours in the image, which is then applied as defined in (4). By accurately preserving the shape and position of objects while discarding extraneous details, our proposed models achieve high recognition accuracy at low bit rates. In addition, we also propose method to further enhance image compression performance for real-world applications.

We propose two ICM methods: SA-ICM and ST-ICM. First, Section III-B outlines our previous work on SA-ICM. Next, the extended version, ST-ICM, is introduced in Section III-C, followed by its application in Section III-D.

### B. SA-ICM

We utilize the Segment Anything Model (SAM) [43] to create mask images that accurately capture the shape and position of objects and backgrounds within an image [44], [45], [46]. SAM is a segmentation model capable of identifying any region, regardless of the type of object, as its name suggests. Its versatility and high accuracy across diverse objects stem from extensive training on vast amounts of image data using semi-supervised learning. Although SAM does not perform classification tasks, meaning it cannot explicitly label the segmented regions, its high segmentation accuracy is a notable advantage.

Leveraging this strength, we use SAM to generate mask images representing the shape and position of objects and backgrounds, incorporating them into the training of the LIC model. The process of mask creation and LIC training is depicted on the left side (training stage) of Fig. 2. In the mask creation process, the image is first fed into SAM to generate a segmentation map. During this step, the confidence threshold for segment is set to a constant value ($\alpha$). The resulting segmentation map is then processed using the Canny edge detector to extract segment boundaries. In the created binary mask, the boundary pixels are assigned a value of 1, and all other regions are set to 0. This mask generation process can be described using the following equation:

$$mask_x = canny(sam(\boldsymbol{x}, \alpha)). \quad (5)$$

In (5), the functions *sam* and *canny* represent region segmentation using SAM and Canny edge detection, respectively. An example of a generated mask is shown in Fig. 3 (d). As demonstrated in this figure, the binary mask effectively captures not only the objects but also the shape and position of background elements, such as mountains. The LIC model is trained by incorporating the generated mask image into the loss function described in (4). The objective of this training is to optimize the LIC model to faithfully decode only the

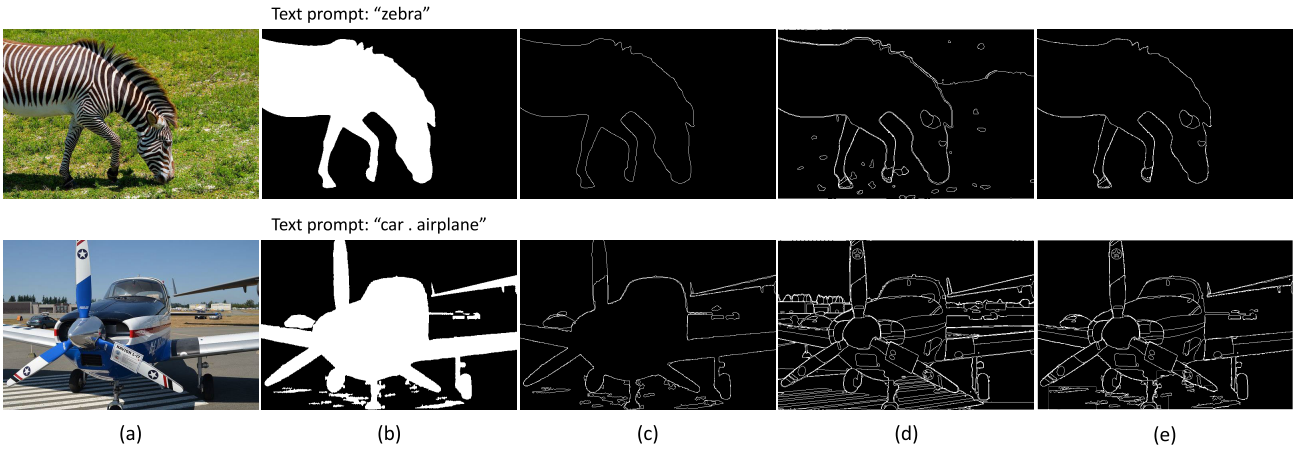


**FIGURE 3.** Examples of mask images and their corresponding original images. (a): Original images. (b): Binary masks generated by Grounded-SAM. (c): Edges obtained by applying the Canny edge detector to the mask (b). (d): Edges obtained by applying the Canny edge detector to the segmentation map output by Segment Anything. These masks are used in the training process of SA-ICM. (e): Binary masks computed as the Hadamard product of the expanded mask (b) and the edges in (d). These masks are utilized in the training process of ST-ICM.

boundaries of the regions identified in the segmentation map, while discarding other textures.

### C. ST-ICM

Masks generated by Segment Anything include the boundaries of every object in the image. As a result, the LIC model optimized with these binary masks is capable of accurately decoding all background edges and object shapes. However, for object detection and instance segmentation models, information about every segment boundary is not always necessary. For instance, boundary information for image elements such as oceans or mountains, which are irrelevant for object detection tasks, becomes redundant. To address this and enhance the compression performance of SA-ICM, we propose a new ICM method, ST-ICM, which focuses on compressing the boundaries of arbitrary target objects. By limiting the information in the binary mask to the boundaries of target objects, background information can be discarded, leading to more efficient compression.

The process of creating a binary mask for ST-ICM training flow is illustrated in Fig. 2. This process utilizes Grounded-SAM [47], a model that performs segmentation for objects belonging to arbitrary classes. Grounded-SAM combines SAM with a detection model for arbitrary objects (Grounding-DINO [48]) to generate segmentation maps for any specified object. In the first half of the binary mask creation process, we acquire segments of arbitrary objects using Grounded-SAM, as detailed in steps a)-c) below:

a) Text prompt and the original image are input into Grounding-DINO, which detects the specified object in the image based on the text. We utilize the ground truth class label of the dataset as the text prompt.
b) The image region within the detected bounding box is passed to SAM to obtain a segmentation result.
c) The similarity between the segments generated by SAM and the text prompts is evaluated, and a segmentation map ($sm$) is output for segments that exceed the text similarity threshold.

The segmentation map ($sm$) resulting from this process is shown in Fig. 3 (b), and its application to the Canny edge detector is illustrated in Fig. 3 (c). The binary mask obtained at this stage captures the object's contour but excludes background information and internal texture details.

In contrast, the mask used for SA-ICM training, shown in Fig. 3 (d), contains a small amount of edge information within the object. To replicate the recognition accuracy achieved with SA-ICM, the edge information inside the object is incorporated into the binary mask shown in Fig. 3 (c). This is achieved through the following additional steps d)-f):

d) The mask region of $sm$ is slightly expanded using a morphological transformation, producing an extended mask image denoted as $sm'$.
e) A binary mask ($mm$), as shown in Fig. 3 (d), is generated using (5). This mask (mm) is exactly the same as the binary mask applied in the training stage of SA-ICM.
f) The Hadamard product (element-wise multiplication) of $mm$ and $sm'$ is computed.

Through these steps, a binary mask is generated, as shown in Fig. 3 (e). This mask excludes background boundary information but includes the contours of the target object and a small amount of internal edge details. Finally, these masks are incorporated into (4) to train the LIC model. The goal of this training is to optimize the LIC model to decode only the shape and limited details of the target object.

### D. APPLICATIONS OF ST-ICM

By defining specific use cases for the ICM method, it is possible to further enhance image compression

performance through the application of ST-ICM. As described in Section III-C, ST-ICM is designed to faithfully decode the shape and position of various objects, and the binary mask in its training process contains a great deal of object shape information. However, in practical use cases of the ICM method within edge-cloud systems, the focus is often on detecting only specific objects. In such scenarios, Grounding-DINO [48] can be utilized to develop ST-ICM application models optimized for the targeted objects. For example, farm management may require the detection of sheep and cattle, while traffic monitoring may need to detect vehicles and persons. In these cases, decoding the shapes of irrelevant objects is unnecessary. By devising input prompts to Grounding-DINO, decoding of unnecessary image parts can be avoided. Specifically, in the image processing step described in a), compression for person detection can be achieved by setting "person" as a text prompt. This approach discards information about other objects, enabling image representation for person recognition at a lower bit rate.

Grounding-DINO supports arbitrary text input by incorporating BERT [49] text encoder. It can also support arbitrary images by using Swin-transformer [50] as an image encoder. Leveraging this feature, binary masks specific to any target object can be generated. By training LIC models with these customized masks, it is possible to develop ICM methods optimized for recognition tasks focused on specific objects.

## IV. EXPERIMENT

### A. IMPLEMENTATION OF THE PROPOSED COMPRESSION MODELS

To evaluate the effectiveness of the proposed methods, we verify their image compression performance through experiments. This section provides a detailed explanation of the training methods for SA-ICM and ST-ICM. We begin by describing the training process for SA-ICM. During mask creation, Segment Anything is used in combination with the Canny edge detector, as formulated in (5). In this experiment, the parameter $\alpha$ in (5) is set to 0.78. For training, we use 118,287 images from the COCO-train dataset [51]. The LIC model employed is LIC-TCM [52], which is trained using the loss function defined in (4). In this function, five constant $\lambda$ values are assigned to control the compression rate: 0.02, 0.03, 0.04, 0.05, and 0.06. Initially, $\lambda$ is set to 0.05, and the model is trained for 50 epochs to obtain the corresponding LIC model. Subsequently, the other $\lambda$ values are applied, and the LIC models for these values are obtained by fine-tuning the initial model for an additional 25 epochs per $\lambda$ value.

Next, we describe the training process for ST-ICM. Binary mask creation follows the mask generation steps a)–f) outlined in Section III-C. In the first half of the process (steps a)–c)), both the box threshold and the text threshold are set to 0.2 when masks are generated using Grounded-SAM. These thresholds correspond to the confidence level for bounding box detection in Grounding-DINO and its similarity to the text prompt, respectively. For the text prompts in step a), we apply the ground truth class labels from the COCO dataset. In step e), where SAM is used for image segmentation, the parameter $\alpha$ is set to 0.78, consistent with the SA-ICM implementation. As with SA-ICM training, the COCO-train dataset is used for training, and LIC-TCM is applied as the LIC model. The loss function follows (4), where four values— 0.03, 0.04, 0.05, and 0.06—are assigned to $\lambda$. Initially, $\lambda$ is set to 0.05, and the model is trained. The remaining $\lambda$ values are then applied sequentially, with the model fine-tuned to obtain versions corresponding to each $\lambda$ value.

### B. IMPLEMENTATION OF IMAGE RECOGNITION MODELS

To assess the image recognition accuracy of the decoded images, multiple recognition models are implemented. To demonstrate the versatility of the proposed ICM method across different recognition architectures, we employ several types of models. Specifically, we adopt YOLOv5 [53] and Mask R-CNN [54] for object detection and Mask R-CNN [54] for instance segmentation. For training and testing YOLOv5, we use the official implementation from the Ultralytics repository, selecting YOLOv5m as the model size. For Mask R-CNN, we utilize MMDetection [55]. We train Mask R-CNN with FPN (Feature Pyramid Networks) based on a prepared scheduler.

All recognition models are trained using compressed images generated by the ICM method. Specifically, images from the COCO-train dataset are compressed using both SA-ICM and ST-ICM, and these decoded images are used as training data for the recognition models. For training data generation, we use SA-ICM and ST-ICM models with a $\lambda$ value of 0.05 in the loss function (4). By training the recognition models on these datasets, we aim to improve image recognition accuracy by developing models that are well-suited for SA-ICM and ST-ICM, respectively.

### C. IMAGE COMPRESSION PERFORMANCE

To evaluate compression performance, we use 5,000 images from the COCO-val dataset. We apply the trained SA-ICM and ST-ICM to the images of the COCO-val dataset to obtain the coded images. Fig. 4 (B) and 4 (D) display the images compressed by each ICM method, with the corresponding bitrate indicated in the bottom right corner of each figure. From these images, it is evident that many textures have been discarded, while object contours remain well-preserved. Notably, the background in the ST-ICM decoded images appears more blurred compared to those from SA-ICM, leading to a lower bitrate for ST-ICM. By selectively discarding parts of the image, as seen in these compressed versions, the image can be represented at a reduced bitrate. Fig. 4 (C) and 4 (E) illustrate the results of object detection using YOLOv5 on the SA-ICM and ST-ICM compressed images, respectively. These results confirm that the compressed images can still be effectively used for image recognition tasks.


**FIGURE 4.** Example of decoded images and the results of detection by yolov5 on those images.

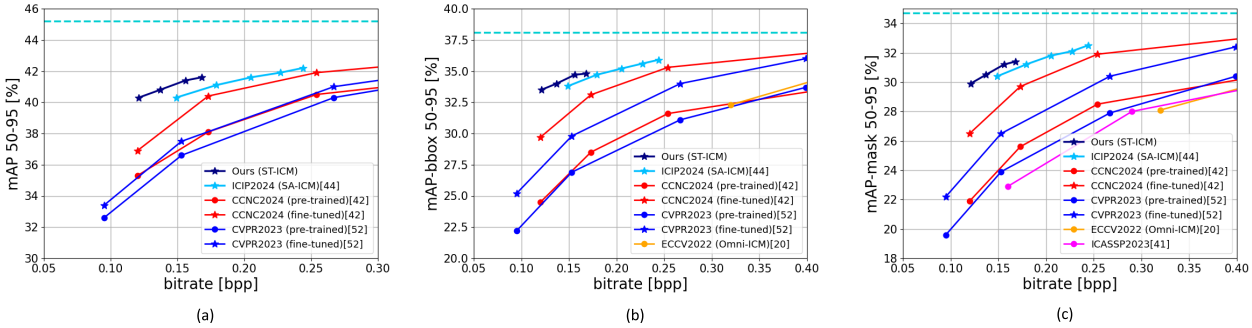


**FIGURE 5.** Compression performance in image recognition accuracy. (a): YOLOv5 as an object detection model. (b): Mask-RCNN as an object detection model. (c): Mask-RCNN as an instances segmentation model.

Fig. 5 presents the image compression performance of the proposed methods for image recognition models. The left and center graphs in Fig. 5 show the Rate-mAP curves of the compression methods when YOLOv5 and Mask R-CNN are used as detection models, respectively. The light blue dotted line in the figure represents the recognition accuracy when uncompressed images are input to the recognition model. The navy and light blue curves represent the image compression performance of ST-ICM and SA-ICM. The red and orange curves correspond to the performance of the conventional ICM methods, while the blue curve represents the compression performance of the LIC-TCM, designed for human vision. The points represented by circles indicate recognition accuracy when the detection models are applied without fine-tuning, whereas the points represented by stars show recognition accuracy when fine-tuning is performed using compressed images. A comparison between the light blue dotted line and the navy curve reveals a reduction in recognition accuracy of 3.6 percentage points with YOLOv5 and 3.3 percentage points with Mask R-CNN. Nevertheless, our compression method demonstrates that high recognition accuracy can be maintained even at low bit rates. These results confirm that the proposed ICM methods are effective image compression methods for object detection models.

The right graph in Fig. 5 illustrates the image compression performance of the proposed methods for the instance segmentation model. Mask R-CNN is used as the segmentation model. In the figure, the light blue dotted line indicates the recognition accuracy achieved when uncompressed images are fed into the recognition model. The navy and light blue curves denote the compression performance of ST-ICM and SA-ICM, respectively. The red-orange-magenta and blue curves represent the performance of the conventional ICM method and the image compression method for human perception, respectively. This graph demonstrates that the proposed ICM methods outperform other ICM approaches in compression performance for the segmentation model.

These Rate-mAP curves validate the effectiveness of the proposed method for both object detection and instance segmentation models. From these graphs, we can confirm that the compression performance of ST-ICM outperforms SA-ICM, especially at low bitrates. In addition, these experimental results are summarized in Table 1. The numerical values in the table represent the BD-rate-mAP [%]. The BD-rate (Bjøntegaard delta rate) is an evaluation metric used to compare the compression performance between two RD-curves. While this metric is commonly used to compare RD-curves defined defined in terms of bitrate and PSNR, in this paper, we apply it to compare Rate-mAP curves, which are defined defined in terms of bitrate and mAP. To clearly distinguish this usage, we refer to the metric as BD-rate-mAP throughout this paper. The BD-rate-mAP values shown in Table 1 indicate the results of comparing each compression method with the LIC model designed for human perception (LIC-TCM [52]). From the results in this table, it can be confirmed that the proposed method outperforms the other compression methods. Furthermore, its versatility is confirmed by its successful application to multiple recognition models.

**TABLE 1.** BD-rate-mAP (%) comparison across tasks and recognition models.

| method | Detection | | Segmentation |
|---|---|---|---|
| | yolov5 | mask r-cnn | mask r-cnn |
| CVPR2023 (pre-trained) | 0.00 | 0.00 | 0.00 |
| CVPR2023 (fine-tuned) | -11.24 | -30.80 | -29.20 |
| ECCV2022 (Omni-ICM) | - | -3.26 | 15.54 |
| ICASSP2023 | - | - | 9.86 |
| CCNC2024 (pre-trained) | -6.81 | -5.90 | -8.04 |
| CCNC2024 (fine-tuned) | -33.42 | -49.98 | -49.22 |
| ICIP2024 (SA-ICM) | -41.83 | -63.25 | -62.58 |
| Ours (ST-ICM) | **-53.43** | **-68.34** | **-66.65** |

### D. PERFORMANCE OF ST-ICM APPLICATION

As discussed in Section III-D, applying ST-ICM as an image compression method for specific object recognition can further enhance compression performance. To evaluate this approach, we develop an image compression model specifically for person recognition by restricting the target object to ‘‘person.’’ During the creation of the binary mask


**FIGURE 6.** Examples of decoded images by each compression model. (a): original image, (b): decoded image with SA-ICM, (c): decoded image with ST-ICM, (d): decoded image with ST-ICM_person.

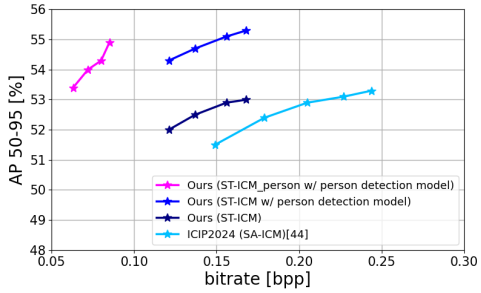


**FIGURE 7.** Image compression performance for person detection models.

used in training, the prompt "person" is fed into the Grounding-DINO text encoder. The generated mask is then incorporated into the loss function (4) to train the LIC model. For the $\lambda$ parameter in equation (4), four values—0.03, 0.04, 0.05, and 0.06—are substituted. As with the training of other proposed compression models, the model is initially trained with $\lambda = 0.05$. Subsequently, it is fine-tuned for the other $\lambda$ values to optimize compression model.

The performance of the image compression models (ST-ICM_person) for person detection is evaluated as follows. First, an example of a decoded image is presented in Fig. 6. Compared to the original ST-ICM, the images encoded by ST-ICM_person contain fewer non-person textures. This enables the image to be represented with less information while preserving essential details for person recognition. In the evaluation process, YOLOv5 is used as the recognition model and COCO-val dataset is utilized as the dataset, as in the ST-ICM compression performance measurement. To evaluate the person recognition accuracy of images decoded by ST-ICM_person, the recognition model is fine-tuned specifically for person detection. Fig. 7 illustrates the relationship between bitrate and person detection accuracy (AP). In this figure, the magenta and blue curves represent the

**TABLE 2.** Compression performance comparison between ST-ICM applied for specific object recognition and the original ST-ICM.

| method | BD-rate-mAP (%) |
|---|---|
| SA-ICM w/ normal detection model | 0.00 |
| ST-ICM w/ normal detection model | -25.11 |
| ST-ICM w/ person detection model | -53.51 |
| ST-ICM_person w/ person detection model | **-74.67** |

compression performance of ST-ICM_person and ST-ICM for yolov5 tuned toward person detection, respectively. The navy curve and the light blue curve represent the compression performance of ST-ICM and SA-ICM, and these person recognition accuracies are measured by applying the COCO 80 class detection model. The graph in Fig. 7 demonstrates that ST-ICM_person outperforms both ST-ICM and SA-ICM as an image compression method for person recognition. Limiting the detection target to "person" facilitates training of the detection model and improves detection accuracy.

In addition, the comparison results of these methods are summarized in Table 2. As in Table 1, BD-rate-mAP is used as the evaluation metric for the comparison. In Table 2, SA-ICM serves as the baseline for evaluating the compression performance of each method. From this table, it can be observed that ST-ICM_person achieves superior image compression performance for person detection compared to the other methods. These results confirm that ST-ICM can be effectively applied as an image compression method tailored for specific object recognition.

## V. CONCLUSION

We propose two novel ICM methods: SA-ICM and ST-ICM. Building on the hypothesis that object contour information is crucial for image recognition models, we developed a compression method that accurately preserves these contours during compression process. By incorporating Segment Anything into SA-ICM and Grounded-SAM into ST-ICM during training, both methods can generate decoded images with high versatility across various recognition models. Moreover, experiments have shown that ST-ICM further enhances compression performance by selectively focusing on objects to be recognized. We believe these proposed ICM methods can be effectively utilized in edge-cloud systems, as they are not tailored to a specific recognition model and do not increase the encoder's computational load.

On the other hand, there are still some challenges. Compressed images from both SA-ICM and ST-ICM tend to retain unnecessary information for object detection tasks. Since object detection involves estimating the position and size of target objects, excessive shape details may be redundant. Therefore, future research should explore more efficient image representation methods for object detection models, aiming to achieve lower bit rates without compromising detection performance. In such exploration, it is necessary to reconsider the method for generating mask images used in the training process. The mask images applied during the

training of ST-ICM are created by a segmentation model and capture the shapes of objects. However, capturing precise object shapes is not a requirement for object detection tasks. Therefore, for training LIC models tailored to object detection, it is considered effective to generate mask images using a detection model. For example, a mask image can be created by setting the detected bounding box regions to white and the rest to black, and apply it to the calculation of the loss function (4). This design of the loss function enables the construction of an LIC model that reconstructs regions containing objects and discards the rest. Moreover, since the compression method designed for the detection model shares similar characteristics with ST-ICM, these methods can be integrated to develop a scalable image compression framework. These future research directions are valuable for designing image compression methods that do not degrade recognition accuracy, by facilitating a better understanding of the image information required for each recognition task.

## REFERENCES

[1] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. 18–34, Feb. 1992.

[2] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. Consum. Electron.*, vol. 46, no. 4, pp. 1103–1127, Nov. 2000.

[3] J. Alakuijala, R. V. Asseldonk, S. Boukortt, M. Bruse, I. Comşa, M. Firsching, T. Fischbacher, E. Kliuchnikov, S. Gomez, R. Obryk, K. Potempa, A. Rhatushnyak, J. Sneyers, Z. Szabadka, L. Vandevenne, L. Versari, and J. Wassenberg, "JPEG XL next-generation image compression architecture and coding tools," *Proc. SPIE*, vol. 11137, pp. 112–124, Sep. 2019.

[4] J. Ascenso, E. Alshina, and T. Ebrahimi, "The JPEG AI standard: Providing efficient human and machine visual data consumption," *IEEE MultimediaMag.*, vol. 30, no. 1, pp. 100–111, Jan. 2023.

[5] *ITU-T and ISO/IEC JTC 1, Advanced Video Coding for Generic Audiovisual Services*, Standard ISO/IEC 14496–10, 2010.

[6] C. Horne, T. Naveen, A. Tabatabai, R. O. Eifrig, and A. Luthra, "Study of the characteristics of the MPEG2 4:2:2 profile-application of MPEG2 in studio environment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 251–272, Jun. 1996.

[7] *High Efficiency Video Coding*, Standard ISO/IEC 23008-2, 2013.

[8] *Versatile Video Coding*, Standard ISO/IEC 23090-3, 2020.

[9] A. Horé and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Istanbul, Turkey, Aug. 2010, pp. 2366–2369.

[10] N. Le, H. Zhang, F. Cricri, R. Ghaznavi-Youvalari, and E. Rahtu, "Image coding for machines: An end-to-end learned approach," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Toronto, ON, Canada, Jun. 2021, pp. 1590–1594.

[11] N. Le, H. Zhang, F. Cricri, R. Ghaznavi-Youvalari, H. R. Tavakoli, and E. Rahtu, "Learned image coding for machines: A content-adaptive approach," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Shenzhen, China, Jul. 2021, pp. 1–6.

[12] S. Wang, Z. Wang, S. Wang, and Y. Ye, "Deep image compression towards machine vision: A unified optimization framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 6, pp. 2979–2989, Jun. 2022.

[13] R. Feng, J. Liu, X. Jin, X. Pan, H. Sun, and Z. Chen, "Prompt-ICM: A unified framework towards image coding for machines with task-driven prompts," 2023, *arXiv:2305.02578*.

[14] A. Alkhateeb, A. Gnutti, F. Guerrini, R. Leonardi, J. Ascenso, and F. Pereira, "JPEG AI compressed domain face detection," in *Proc. IEEE 26th Int. Workshop Multimedia Signal Process. (MMSP)*, West Lafayette, IN, USA, Oct. 2024, pp. 1–6.

[15] L. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video coding for machines: A paradigm of collaborative compression and intelligent analytics," *IEEE Trans. Image Process.*, vol. 29, pp. 8680–8695, 2020.

[16] K. Fischer, F. Brand, C. Herglotz, and A. Kaup, "Video coding for machines with feature-based rate-distortion optimization," in *Proc. IEEE 22nd Int. Workshop Multimedia Signal Process. (MMSP)*, Tampere, Finland, Sep. 2020, pp. 1–6.

[17] T. Shindo, T. Watanabe, K. Yamada, and H. Watanabe, "Accuracy improvement of object detection in VVC coded video using YOLO-v7 features," in *Proc. IEEE Int. Conf. Artif. Intell. Eng. Technol. (IICAIET)*, Sep. 2023, pp. 247–251.

[18] W. Yang, H. Huang, Y. Hu, L.-Y. Duan, and J. Liu, "Video coding for machines: Compact visual representation compression for intelligent collaborative analytics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 7, pp. 5174–5191, Jul. 2024.

[19] W. Gao, S. Liu, X. Xu, M. Rafie, Y. Zhang, and I. Curcio, "Video coding for machines: Compact visual representation compression for intelligent collaborative analytics," 2021, *arXiv:2105.12653*.

[20] R. Feng et al., "Image coding for machines with omnipotent feature learning," in *Proc. Eur. Conf. Comput. Vis.*, Cham, Switzerland: Springer, Oct. 2022, pp. 510–528.

[21] Y. Wang, C. Yang, S. Lan, L. Zhu, and Y. Zhang, "End-edge-cloud collaborative computing for deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 4, pp. 2647–2683, 4th Quart., 2024.

[22] J. Yao, S. Zhang, Y. Yao, F. Wang, J. Ma, J. Zhang, Y. Chu, L. Ji, K. Jia, T. Shen, A. Wu, F. Zhang, Z. Tan, K. Kuang, C. Wu, F. Wu, J. Zhou, and H. Yang, "Edge-cloud polarization and collaboration: A comprehensive survey for AI," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6866–6886, Jul. 2022.

[23] P. Lou, S. Liu, J. Hu, R. Li, Z. Xiao, and J. Yan, "Intelligent machine tool based on edge-cloud collaboration," *IEEE Access*, vol. 8, pp. 139953–139965, 2020.

[24] H. Gu, L. Zhao, Z. Han, G. Zheng, and S. Song, "AI-enhanced cloud-edge-terminal collaborative network: Survey, applications, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 2, pp. 1322–1385, 2nd Quart., 2024.

[25] C. Ding, A. Zhou, Y. Liu, R. N. Chang, C.-H. Hsu, and S. Wang, "A cloud-edge collaboration framework for cognitive service," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1489–1499, Jul. 2022.

[26] J. Balle, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent.*, May 2018, pp. 1–10.

[27] D. Minnen, J. Ballé, and G. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–10.

[28] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7936–7945.

[29] I. V. Bajic, W. Lin, and Y. Tian, "Collaborative intelligence: Challenges and opportunities," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Toronto, ON, Canada, Jun. 2021, pp. 8493–8497.

[30] J. I. Ahonen, N. Le, H. Zhang, F. Cricri, and E. Rahtu, "Region of interest enabled learned image coding for machines," in *Proc. IEEE 25th Int. Workshop Multimedia Signal Process. (MMSP)*, Poitiers, France, Sep. 2023, pp. 1–6.

[31] B. Peng, T. Lin, D. Jin, Z. Pan, and J. Lei, "Saliency map-guided end-to-end image coding for machines," *IEEE Signal Process. Lett.*, vol. 31, pp. 1755–1759, 2024.

[32] Y. Qi, R. Feng, Z. Zhang, and Z. Chen, "Image coding for machines based on non-uniform importance allocation," in *Proc. IEEE Int. Conf. Vis. Commun. Image Process. (VCIP)*, vol. 31, Dec. 2023, pp. 1–5.

[33] Z. Zhang, L. Yu, Q. Zhang, J. Mei, and T. Guan, "Towards efficient learned image coding for machines via saliency-driven rate allocation," in *Proc. IEEE Int. Conf. Vis. Commun. Image Process. (VCIP)*, Dec. 2023, pp. 1–5.

[34] H. Choi and I. V. Bajic, "High efficiency compression for object detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, Apr. 2018, pp. 1792–1796.

[35] J. I. Ahonen, N. Le, H. Zhang, A. Hallapuro, F. Cricri, H. R. Tavakoli, M. M. Hannuksela, and E. Rahtu, "NN-VVC: Versatile video coding boosted by self-supervisedly learned image coding for machines," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2023, pp. 10–19.

[36] L. D. Chamain, F. Racapé, J. Bégaint, A. Pushparaja, and S. Feltman, "End-to-end optimized image compression for machines, a study," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Mar. 2021, pp. 163–172.
[37] H. Choi and I. V. Bajic, "Deep feature compression for collaborative object detection," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 3743–3747.
[38] H. Choi and I. V. Bajic, "Scalable image coding for humans and machines," *IEEE Trans. Image Process.*, vol. 31, pp. 2739–2754, 2022.
[39] N. Yan, C. Gao, D. Liu, H. Li, L. Li, and F. Wu, "SSSIC: Semantics-to-signal scalable image coding with learned structural representations," *IEEE Trans. Image Process.*, vol. 30, pp. 8939–8954, 2021.
[40] K. Fischer, F. Brand, and A. Kaup, "Boosting neural image compression for machines using latent space masking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 4, pp. 3719–3731, Apr. 2022.
[41] B. Li, J. Liang, H. Fu, and J. Han, "ROI-based deep image compression with Swin transformers," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Rhodes Island, Greece, Jun. 2023, pp. 1–5.
[42] T. Shindo, T. Watanabe, K. Yamada, and H. Watanabe, "Image coding for machines with object region learning," in *Proc. IEEE 21st Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2024, pp. 1040–1041.
[43] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 4015–4026.
[44] T. Shindo, K. Yamada, T. Watanabe, and H. Watanabe, "Image coding for machines with edge information learning using segment anything," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2024, pp. 3702–3708.
[45] T. Shindo, T. Watanabe, Y. Tatsumi, and H. Watanabe, "Delta-ICM: Entropy modeling with delta function for learned image compression," 2024, *arXiv:2410.07669*.
[46] T. Shindo, T. Watanabe, Y. Tatsumi, and H. Watanabe, "Scalable image coding for humans and machines using feature fusion network," in *Proc. IEEE 26th Int. Workshop Multimedia Signal Process. (MMSP)*, West Lafayette, IN, USA, Oct. 2024, pp. 1–6.
[47] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded SAM: Assembling open-world models for diverse visual tasks," 2024, *arXiv:2401.14159*.
[48] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *Proc. ECCV*, pp. 1–17, 2024.
[49] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[50] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Montreal, QC, Canada, Oct. 2021, pp. 9992–10002.
[51] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[52] J. Liu, H. Sun, and J. Katto, "Learned image compression with mixed transformer-CNN architectures," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14388–14397.
[53] G. Jocher et al., "Ultralytics/YOLOv5: V7.0-YOLOv5 sota realtime instance segmentation," Zenodo, Geneva, Switzerland, Tech. Rep., 2022.
[54] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2980–2988.
[55] K. Chen et al., "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.

**TAKAHIRO SHINDO** (Graduate Student Member, IEEE) received the B.E. degree from Waseda University, Tokyo, Japan, in 2023, where he is currently pursuing the M.E. degree with the Graduate School of Fundamental Science and Engineering. His research interests include deep learning, image recognition, image and video processing, image compression, and image coding for machines.

**TAIJU WATANABE** received the B.E. degree from Waseda University, Tokyo, Japan, in 2023, where he is currently pursuing the M.E. degree with the Graduate School of Fundamental Science and Engineering. His research interests include deep learning, video prediction, video interpolation, image and video processing.

**YUI TATSUMI** is currently pursuing the B.E. degree with the School of Fundamental Science and Engineering, Waseda University, Tokyo, Japan. Her research interests include deep learning, image and video processing, image compression, and image coding for machines.

**HIROSHI WATANABE** (Life Member, IEEE) received the B.E., M.E., and Ph.D. degrees in engineering from Hokkaido University, in 1980, 1982, and 1985, respectively. He joined Nippon Telegraph and Telephone Corporation (NTT), in 1985, and was engaged in research and development of video coding systems until 2000. He is currently a Professor with the School of Fundamental Science and Engineering, Waseda University. His research interests include deep learning, object recognition, image and video processing, and video coding. He is a member of IEICE, ITE, IPSJ, and IIEEJ. He served as the Chairman of ISO/IEC JTC 1/SC 29, from 1999 to 2006.