Adaptive Compression Offloading and Resource Allocation for Edge Vision Computing

Wenjing Xiao, *Member, IEEE*, Yixue Hao, *Member, IEEE*, Junbin Liang, Long Hu,
Salman A. Alqahtani, and Min Chen, *Fellow, IEEE*

***Abstract*—The rapid progress in edge computing (EC) and 5G wireless communication technology has opened up novel opportunities for intelligent applications driven by deep neural networks (DNNs). In particular, machine vision tasks are widely used in mobile/edge computing scenarios. However, the real-time and dense data transmission involved in vision inference services impose significant communication burdens on wireless networks. Thus, this paper investigates the general vision services strategy with cognitive computing network and proposes a communication-efficient edge inference deployment architecture for vision analysis tasks. In this framework, users dynamically perceive the inference data in local, and then compress and offload them to edge servers to perform inference. Specifically, we present a collaborative optimization model of compression ratio and network bandwidth to generate the reliable compression offloading and resource allocation scheme. For this model, the offloading scheme carefully considers the constraints imposed by delay and resources and maximizes the success probability of vision inference tasks. To improve the vision inference performance in the edge network, we further propose a flexible data compression algorithm for images or video frames, which can preserve the more important visual information under a fixed compression rate to reduce the inference accuracy loss from compression. This algorithm first perceives the importance of visual information at different pixel positions, and then compresses different visual regions to varying degrees according to their importance, enabling content-aware adaptive vision data coding. Experimental results show that our proposed offloading model and compression strategy outperform other algorithms, achieving significant communication improvements and performance gains.**

***Index Terms*—Edge computing, computer vision service, compression offloading, resource allocation, data compression.**

I. Introduction

IN RECENT times, deep learning has demonstrated remarkable performance in various computer vision tasks, enabling machine equipment such as unmanned aerial vehicles [1] and autonomous driving systems to analyze visual data in real-time and achieve environment understanding at the pixel level [2]. However, due to limited computational power and storage space, running deep learning applications on Internet of Thing (IoT) devices remains challenging. With the deployment of 5G networks featuring mobile edge computing, a promising solution is offloading DNN inference tasks to edge servers near users. In this paradigm, IoT devices transmit captured video frames or images for computation processing at executable edge servers. Once the inference is completed, the results are seamlessly delivered back to the users. However, with the rapid growth of sensors and service requests, bandwidth limitations have become one of the most pressing challenges hindering the development of edge vision computing [3], [4]. Especially for vision applications, the transmission of numerous high-resolution image/video frames imposes a significant communication burden on wireless networks. The compression offloading schemes are to compress the original data before vision task offloading and then transmit the compressed data to the edge server, thus effectively alleviating the communication cost [5], [6]. While task offloading with a high compression rate can minimize channel resource demands, it often leads to noticeable accuracy losses of inference services. Conversely, offloading at a low compression rate requires more network resources, and may even cause transmission failures due to excessive latency. Thus, exploring a suitable compression offloading and resource allocation scheme is crucial for balancing the communication efficiency and inference performance in the edge vision computing network.

Moreover, extensive researches in computer vision have proved that visual features at different positions have different contributions to the final model accuracy. However, current data compression strategies usually encode different pixel regions of an image/video frame with the same compression level. Such compression strategies lack discriminative insight into the importance of different visual regions, which limits

Manuscript received 1 September 2023; revised 6 March 2024 and 11 April 2024; accepted 7 May 2024. Date of publication 14 May 2024; date of current version 6 December 2024. This work was supported by Distinguished Scientist Fellowship Program (DSFP2024), King Saud University, Riyadh, Saudi Arabia. This work was supported in part by State Key Laboratory of Featured Metal Materials and Life-cycle Safety for Composite Structures under Grant AA23073019, and in part by the Guangxi Key R&D Plan Project under Grant AB23049006. The associate editor coordinating the review of this article and approving it for publication was D. B. Rawat. *(Corresponding authors: Wenjing Xiao; Min Chen.)*

Wenjing Xiao and Junbin Liang are with the Guangxi Key Laboratory of Multimedia Communications and Network Technology, School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China (e-mail: wenjingx@ieee.org; liangjb@gxu.edu.cn).

Yixue Hao and Long Hu are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yixuehao@hust.edu.cn; hulong@hust.edu.cn).

Salman A. Alqahtani is with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11574, Saudi Arabia (e-mail: salmanq@ksu.edu.sa).

Min Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with Pazhou Laboratory, Guangzhou 510330, China (e-mail: minchen@ieee.org).

Digital Object Identifier 10.1109/TCCN.2024.3400820

the optimization space for trade-offs between compression ratio and inference performance. In general, the existing data compression-driven EC systems encounter two primary problems: (1) How to jointly optimize compression offloading and network resources while considering network delay requirements and computing resource constraints to ensure the quality of inference service? (2) How to retain key visual information as much as possible during data compression, so as to effectively alleviate the decline of inference accuracy while reliably compressing visual data?

To tackle these problems, in this paper, we investigate the general vision services strategy for edge computing and propose a communication-efficient edge inference deployment architecture with adaptive compression offloading. First, it is important to consider the interaction between compression ratio and resource allocation strategy in data compression-based task offloading. This motivates us to propose a collaborative optimization model of compression ratio and network resource to generate reliable vision task offloading schemes. Therefore, our offloading decision model will jointly optimize the compression ratio and resource allocation scheme of edge intelligent networks. Second, existing vision compression algorithms employ a uniform-level compression scheme, neglecting the discriminative importance of visual features across different pixel regions. To address this limitation, we propose an adaptive image compression algorithm that takes into account the differential impact of individual pixel locations on inference accuracy. By incorporating this insight, our algorithm enables varying degrees of compression for different visual regions based on their respective importance levels. In summary, the main contributions are as follows:

1) We propose a communication-efficient edge inference framework for vision analytics tasks, which dynamically compresses high-resolution data and then transmits compressed data to the edge server for performing inference.
2) Considering the contradiction between limited resources and numerous inference requests, we build a collaborative optimization model of compression rate and resource allocation, where the goal is to maximize the success probability of vision inference tasks on edge computing in wireless networks.
3) This paper proposes a spatial-attention aware image compression strategy, which utilizes the priors of the deep model to perceive the importance of each visual region in an image, enabling adaptive region-wise data compression tailored to their importance.
4) Extensive experiments are conducted to verify the effectiveness and superiority of our optimization algorithm and compression strategy. Results demonstrate that them can improve the success probability of inference tasks by at least 15.1%, and save network bandwidth by about 10% with the fixed inference accuracy.

The remainder of the article is organized as follows: Related works about video analytic and edge inference are briefly reviewed in Section II. We introduce the compression offloading framework for wireless EC networks in Section III. Then, Section IV describes the collaborative optimization model and problem solution. In Section V, we propose the spatial-attention aware compression strategy. Section VI evaluates the performance of the proposed architecture and analyzes experimental results. Finally, Section VII concludes the paper.

II. Related Work

With the rapid growth of sensor devices and service requests, bandwidth limitation has become one of the important challenges for edge vision computing. As a result, the concept of leveraging data compression technology in edge computing has gained significant attention in recent years [7], [8]. In the following, we summarize the related literatures in two categories: (1) the emerging compression offloading for edge computing, and (2) the data compression strategy optimization.

The edge inference scheme driven by data compression first compresses the original data using compression algorithms, and then transmits compressed data to edge nodes to perform inference tasks [9], [10]. Such image lossy compression methods are based on the sensitivity of different visual features of human beings, and achieve a larger compression ratio with a certain loss of information. For deep models, the more complete the data information, the higher the accuracy of inference. Therefore, some recent works aim to design better compression algorithms. In the work presented in [11], the authors established a quantitative relationship between compression ratio and inference accuracy. This relationship model is employed to make compression strategy decisions that strike a balance between accuracy and compression ratio. Another approach proposed in literature [12], [13] quantified the performance change caused by each pixel during compression according to the loss gradient of DNNs. By considering the maximum compression ratio that each pixel can tolerate, an individualized compression strategy is formulated. In addition, several other works studied the DNN-driven semantic compression for edge computing framework, which utilizes encoder, decoder, and channel layers to build an end-to-end semantic communication network. The encoder in local encodes the data into the abstract semantic information. The decoder in edge servers restores semantic information for performing inference [5], [14].

While data compression techniques can reduce communication latency, different compression levels result in varying energy consumption in edge systems. It is unwise to adopt a fixed compression ratio scheme in an EC systems. Thus, in order to improve resource efficiency while maintaining inference quality, optimizing the compression offloading and making resource allocation decisions are crucial. Ren et al. [15] investigated video compression offloading in mobile EC systems and made optimal decisions to execute tasks on local, remote, or mixed-mode servers with the goal of minimizing overall latency. However, their work lacked the optimization of the compression ratio since the task itself involved data compression. In literature [16], the actual limitations of maximum transmission power, wireless access bandwidth, backhaul capacity and computing resources were considered. They jointly optimized compression ratio and

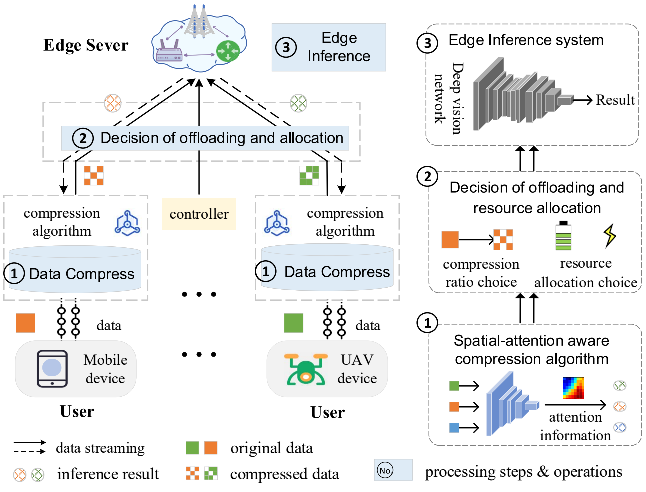


Fig. 1. Architecture of data compression-driven edge computing vision inference system.

task offloading, and formulated them as a Mixed-Integer Nonlinear Programming optimization problem [17] to solve. Furthermore, Wang et al. [18] proposed a data compression-driven multi-user mobile edge computing system to minimize the total energy consumption of the system by optimizing the transmission bandwidth and data compression ratio strategies.

Overall, ensuring the connectivity of edge intelligence is crucial for future communication systems. Existing researches have addressed the challenges of high-traffic network environments through edge inference strategies based on data compression. However, these approaches lack the ability to differentiate between important and unimportant area information of vision data. Moreover, different compression ratio settings can impact network communication performance and model inference accuracy. Therefore, there is a need to jointly optimize the data compression ratio and network resource allocation scheme in order to establish an efficient edge vision inference architecture. This architecture should consider factors such as edge service quality, model inference accuracy, and communication status while exploring content-aware data compression strategies.

III. System Architecture

In edge inference architecture for DNN vision services, numerous high-resolution data are uploaded to edge servers and then return prediction results with negligible size. In this case, the uplink is often overloaded with necessity to optimize its transmission process. Thus, we investigate the data compression-driven edge vision system, as shown in Fig. 1.

Specifically, users collect vision data through local sensor devices and send inference task requests to the controller. This controller is responsible for dynamically determining the appropriate compression rate and resource allocation scheme according to current network status and inference requests. These decisions are then broadcasted to each user. Subsequently, each user follows the compression offloading decision to compress the original vision data and uploads the compressed data to edge servers for inference. Our system mainly includes the collaborative optimization model of compression rate and resource allocation (Co-CRRA), and the spatial attention-aware image compression strategy (SpaT-IMCO):

(1) Co-CRRA optimization model: Due to resource limitations in wireless networks, low compression rates for inference offloading still require large channel resources, potentially leading to the transmission failures caused by excessive service delays. Conversely, while high compression rates can meet resource requirements, they may result in excessive loss of image information so as to seriously affect inference performance. Hence, this paper presents a trade-off between the offloading compression rate and resource allocation scheme with the goal of maximizing the success probability of vision inference tasks to ensure high accuracy of reasoning services. To solve this non-convex optimization model, we jointly optimize the offloading compression rate and resource allocation, which decomposes the original problem into two convex sub-problems for two-stage iterative computation.

(2) SpaT-IMCO compression strategy: Each image/video frame has key visual features, which are more important for its recognition performance. Inspired by the literature [19], we can use the prior knowledge of DNN models to obtain the importance of visual features in different regions. To adapt to the terminal device, a lightweight network module is designed to get the importance of different pixel positions in a image. At a specific compression rate, our compression algorithm executes non-uniform compression across different regions, prioritizing high-resolution preservation in critical image regions and blurring less important features.

IV. Collaborative Optimization Model

In this section, we introduce the network framework and user requests of the edge inference system driven by compression offloading, and model the optimization model of compression offloading and resource allocation. When solving this optimization model, the compression offloading and channel allocation problems will be decoupled into two sub-problems. Therefore, the controller of edge networks determines the optimal data compression rate and wireless resources allocation scheme under satisfying inference requirements of the users and resource constraints of the system. Tab. I shows the main symbols and their meanings of the system.

A. Edge Inference Network

Our system considers a wireless edge computing network consisting of an edge server $\mathcal{E}$ and multiple users $\mathcal{U}s = \{u_i \mid i = 1, 2, \ldots, I\}$. All local users connect to edge servers through shared wireless mediums, where the frequency spectrum is divided into non-overlapping bandwidth channels to avoid mutual interference between different devices. On communication links between the user transmitter (U-TX) and the edge server receiver (E-RX), following Shannon information theory, the data transfer rate of u_i can be calculated as follows:

$$V_i = b_i log_2\left(1 + \frac{h_i p_i}{N_0 b_i}\right) \tag{1}$$

TABLE I
MAIN SYMBOLS AND THEIR MEANINGS OF THE SYSTEM

Symbols	Meanings
$\mathcal{E}$	Edge server in wireless base station
$\mathcal{U}$	User sets for edge services network
I	Number of users
b_i	Bandwidth size of user u_i
p_i	Transmit power allocated to user u_i
h_i	Channel gain between user u_i and edge server
N_0	Noise power spectrum in wireless base station
V_i	Transmission rate of user u_i
D_i^0	Original data size in the inference request of user u_i
o_i	Data compression rate of user u_i
t_i	Transmission delay of user u_i
η_i	Probability of successful execution of user u_i after offloading
Φ_i	Total probability of successful execution of user u_i
t_0	Transmission constraints of offloading data in EC
$B_{\min}$	The minimum bandwidth allocated to users in EC system
$B_{\max}$	The maximum bandwidth allocated to users in EC system
$P_{\min}$	The minimum transmit power allocated to users in EC system
$P_{\max}$	The maximum transmit power allocated to users in EC system

where b_i and p_i denote the bandwidth and transmit power of user u_i respectively, h_i is the channel gain between user u_i and the edge server, and N_0 represents the noise power spectral density. Additionally, let D_i^0 denote the amount of original data obtained from the sensor device of user u_i, and o_i represents the corresponding data compression rate.

Therefore, the actual amount of data transmitted by u_i can be expressed as $D_i = D_i^0 \times (1 - o_i)$, and the transmission delay of u_i is $t_i = \frac{D_i}{V_i}$. In practical scenarios such as vehicular networks, the requested inference tasks are delay-sensitive with strict transmission delay constraints. Let t_0 denote the maximum allowable delay for successful inference. The success probability of u_i transmitting tasks can be expressed as $P(t_i \leq t_0)$. The calculation of $P(t_i \leq t_0)$ is as follows:

$$\begin{aligned} P(t_i \leq t_0) &= P\left(\frac{D_i}{V_i} \leq t_0\right) = P\left(\frac{(1-o_i)D_i^0}{b_i log_2(1+\frac{h_i p_i}{N_0 b_i})} \leq t_0\right) \\ &= P\left(\frac{2^{\alpha_i(1-o_i)}-1}{\beta_i} \leq h_i\right) \\ &= 2G\left(\frac{2^{a_i(1-o_i)}-1}{\beta_i \delta}\right) \end{aligned} \tag{2}$$

where t_i represents the actual transmission delay, $\alpha_i = \frac{D_i^0}{b_i t_0}$, $\beta_i = \frac{p_i}{N_0 b_i}$. Moreover, h_i obeys the normal distribution $N(0, \delta^2)$, where δ represents the variance of the channel gain. The G function in eq. (2) is the tail distribution of the standard normal distribution function [20].

Thus, in order to formulate the influence of data compression and resource allocation on the quality of inference services, we define a metric: the success probability of inference tasks Φ_i, and its formula is calculated as follows:

$$\Phi_i = \eta(o_i) \times P(t_i \leq t_0) \tag{3}$$

where $\eta(o_i)$ is the success probability of inference tasks in the case of successful transmission, and this value is related to the compression rate o_i.

Consequently, the overall system's success probability of inference tasks is expressed as $\mathbf{\Phi} = \sum_{i=1}^{\mathcal{U}} \Phi_i$. As indicated by eq. (3), this success probability is a trade-off decision between communication efficiency and inference quality.

B. Problem Formulation and Solution

According to eq. (2) and eq. (3), $\mathbf{\Phi}$ is influenced by the transmit power, network bandwidth and data compression rate. Our model aims to optimize these three variables with the goal of maximizing the success probability of inference tasks. Let $B_{\min}$ and $P_{\min}$ represent the minimum bandwidth and minimum transmit power allocated to each user respectively. $B_{\max}$ and $P_{\max}$ denote the maximum total bandwidth and maximum total transmit power of the system respectively. Mathematically, the problem P* is expressed as follows:

$$\mathrm{P}* : \max_{\boldsymbol{o},\boldsymbol{b},\boldsymbol{p}} \sum_{i=1}^{\mathcal{U}} \Phi_i \tag{4}$$

$$\text{subject to } C1 : \sum_{i=1}^{U} b_i \leq B_{\max} \tag{5}$$

$$C2 : \sum_{i=1}^{U} x_i P_i \leq P_{\max} \tag{6}$$

$$C3 : b_i \geq B_{\min}, \forall i \in \mathcal{U} \tag{7}$$

$$C4 : P_i \geq P_{\min}, \forall i \in \mathcal{U} \tag{8}$$

$$C5 : 0 < o_i < 1, \forall i \in \mathcal{U}. \tag{9}$$

where constraint $C1$ means that the total bandwidth does not exceed a given threshold. Constraint $C2$ means that the total transmit power of all users cannot exceed a given $P_{\max}$ to ensure the total energy supply of the system is limited. Constraints $C3$ and $C4$ denote the minimum bandwidth and minimum transmit power constraints respectively, while constraint $C5$ is to control the range of compression ratio.

However, since the objective function is not a concave function, its optimal solution cannot be obtained directly [21], [22]. To solve the problem P*, we decouple it into two subproblems for iterative solution. Firstly, we compute the optimal compression ratio for each user with fixed resource allocation variables. Then, we solve the resource allocation variable according to the obtained compression ratio. The first and second steps are performed iteratively until a convergence criterion is met.

1) Solve Compression Ratios: For the solution of the data compression rate, it is necessary to first determine the relationship between the success probability of the inference task and different compress ratios, that is $\eta(o_i)$ function. However, because of the inexplicability of deep neural networks, it is difficult to derive the close-form expression of $\eta(o_i)$. In this article, we leverage both statistical data and function approximation models to find this functional relationship. Specially, we first generate compressed images under different compression ratios based on ImageNet data [23]. Then, we take these images as inputs of the DNN model and compute their corresponding inference performance. Here, multiple groups of experimental data regarding image and model performance under compression rates have been obtained. Inspired by [24], we empirically find that the relationship of $\eta(o_i)$ can be fitted as a compound exponential function by observing above these statistical data, expressed as $\zeta_1 e^{\zeta_2 o_i} + \zeta_3 e^{\zeta_4 o_i}$. Next, we use the numerical method to learn the parameter weights, and the optimal fitting parameter solution is computed as [0.6, −0.3, 0.8, −0.4], which is adopted in this paper. Besides, following the approximation suggested in [25], we specify the *G* function as $G(x) \approx 1/2e^{\frac{-x^2}{2}}$ in our optimization model. So far, assuming that the bandwidth and transmit power allocation scheme has been determined, the $\boldsymbol{b} = \{b_i\}$ and $\boldsymbol{p} = \{p_i\}$ variables in eq. (2) become constants. Therefore, the original problem P* can be transformed into a simplified problem P1, only focusing on optimizing the compression ratio o_i without constraints related to resource allocation:

$$\begin{aligned} \text{P1} :\ & \max_{\boldsymbol{o}} \sum_{i=1}^{\mathcal{U}} \Phi_i = \sum_{i=1}^{\mathcal{U}} \gamma_i \times \eta(o_i) \\ & \text{subject to } \ C5. \end{aligned} \tag{10}$$

where $\gamma(o_i|b_i, p_i)$ is equal to $\exp\{-\frac{1}{2}[\frac{2^{\alpha_i(1-o_i)}-1}{\beta_i \delta}]^2\}$, that is $P(t_i \le t_0)$. From problem P1, it can be seen that when the bandwidth and transmit power have been determined, the compression rates between different users are independent. Therefore, maximizing the sum of Φ_i for all users in problem P1 can be simplified to maximize the individual Φ_i for each user. In this case, we can use the enumeration method to explore the solution of $o_i \in (0, 1)$, to find the optimal image compression rate for each user under fixed resource schemes.

2) Solve Bandwidth and Transmit Power: Given the known image compression rate, we can solve the optimal allocation scheme of bandwidth and transmit power for each user. Since o_i has been determined, the $\eta(o_i)$ can be regarded as a constant, denoted as φ_i. Consequently, the original problem P* can be transformed into problem P2, which aims to allocate bandwidth and transmit power in edge network resources:

$$\begin{aligned} \text{P2} :\ & \min_{\boldsymbol{b},\boldsymbol{p}} \sum_{i=1}^{\mathcal{U}} -\varphi_i \times \gamma(b_i, p_i|o_i) \\ & \text{subject to } \ C1 \sim C4. \end{aligned} \tag{11}$$

To address problem P2, the non-convex constraints *C*4 in eq. (11) are transformed into convex constraints by introducing slack variables. We define the following slack variables: $\boldsymbol{f} = [f_1, f_2, \ldots, f_U]$, $\boldsymbol{l} = [l_1, l_2, \ldots, l_U]$, $\boldsymbol{y} = [y_1, y_2, \ldots, y_U]$, $\boldsymbol{m} = [m_1, m_2, \ldots, m_U]$ and $\boldsymbol{q} = [q_1, q_2, \ldots, q_U]$. To this end, problem P2 can be reformulated as problem P2.1:

$$\text{P2.1} :\ \min_{\boldsymbol{b},\boldsymbol{p},\boldsymbol{f},\boldsymbol{l},\boldsymbol{y},\boldsymbol{m},\boldsymbol{q}} \sum_{i=1}^{\mathcal{U}} -\varphi_i \times f_i \tag{12}$$

$$\text{subject to } \ C1 \sim C3 \tag{13}$$

$$C6 : f_i \le e^{l_i}, \forall i \in \mathcal{U} \tag{14}$$

$$C7 : l_i \le -\frac{1}{2} y_i^2, \forall i \in \mathcal{U} \tag{15}$$

$$C8 : y_i \ge \frac{N_0 b_i m_i}{\delta p_i}, \forall i \in \mathcal{U} \tag{16}$$

$$C9 : m_i \ge 2^{q_i} - 1, \forall i \in \mathcal{U} \tag{17}$$

$$C10 : q_i \ge \frac{D_i^0(1-\sigma)}{b_i t_i}, \forall i \in \mathcal{U}. \tag{18}$$

Based on the above formula, except for constraint *C*6 and constraint *C*8, other constraints have become convex constraints. Next we transform them.

For constraint *C*6, Successive Convex Approximation (SCA) algorithm [26] is utilized to convert it into a convex constraint. Specially, we perform a first-order Taylor expansion on the function e^{l_i} at the point $e^{l_i^j}$, which is $f_i \le e^{l_i^j} + (l_i - l_i^j) e^{l_i^j}$, where the superscript *j* denotes the variable value computed after the *j*-th iteration.

For constraint *C*8, we introduce a slack variable $\boldsymbol{z} = [z_1, z_2, \ldots, z_U]$. This allows us to reformulate constraint $C8$ into two separate constraints: $z_i \ge b_i m_i$ and $y_i p_i \ge \frac{N_0 z_i}{\delta_i}$. Moreover, simplifying this expression, $b_i m_i$ is expanded as $\frac{1}{4}((b_i + m_i)^2 - (b_i - m_i)^2)$. Then, we perform a first-order Taylor expansion of $(b_i - m_i)^2$ at (b_i^j, m_i^j), and transform it into a new constraint form using SCA algorithm:

$$\begin{aligned} z_i \ge & \frac{1}{4}\Big((b_i + m_i)^2 + (b_i^j - m_i^j)^2\Big) \\ & - \frac{1}{2}(b_i - m_i)\Big(b_i^j - m_i^j\Big) \end{aligned} \tag{19}$$

Similarly, $y_i p_i$ is rewritten as $\frac{1}{4}((y_i + p_i)^2 - (y_i - p_i)^2)$, and let $(y_i + p_i)^2$ and $(y_i - p_i)^2$ be imposed the first-order Taylor expansion at the point (y_i^j, p_i^j), calculated as follows:

$$\begin{aligned} \frac{4N_0 z_i}{\delta_i} \le\ & 2(y_i + p_i) * \Big(y_i^j + p_i^j\Big) - \Big(y_i^j + p_i^j\Big)^2 \\ & - 2(y_i - p_i) * \Big(y_i^j - p_i^j\Big) + \Big(y_i^j + p_i^j\Big)^2 \end{aligned} \tag{20}$$

To this end, P2.1 is further expressed as follows:

$$\text{P2.2} :\ \min_{\boldsymbol{b},\boldsymbol{p},\boldsymbol{f},\boldsymbol{l},\boldsymbol{y},\boldsymbol{m},\boldsymbol{q}} \sum_{i=1}^{\mathcal{U}} -\varphi_i \times f_i \tag{21}$$

$$\text{subject to } \ C1 \sim C3, C7, C9, C10 \tag{22}$$

$$C11 : f_i \le e^{l_i^j} + \Big(l_i - l_i^j\Big) e^{l_i^j}, \forall i \in \mathcal{U} \tag{23}$$

$$C12 : \text{eq. (19)} \tag{24}$$

$$C13 : \text{eq. (20)} \tag{25}$$

So far, all constraints of problem P2 have been transformed into convex constraints in problem P2.2, which can be solved by the dual method [27]. Specifically, we initialize l_i^j, b_i^j, m_i^j, x_i^j and p_i^j, and then iteratively solve until the objective function converges. In this way, the optimal compression rate and resource allocation scheme of our edge inference system can be obtained, and the pseudo code for the solution of our Co-CRRA algorithm is presented in Algorithm 1.

3) Algorithm Complexity Analysis: The iteration number of outer-layer of Algorithm 1 denote as L_1, and that of compression rates and resource allocation subproblem are L_2 and L_3. Under fixed resource allocation, the complexity of using the enumeration method is $O({L_2}^U)$ for solving the compression ratios subproblem (P1). With fixed compression

Algorithm 1 Collaborative Optimization Algorithm for Offloading Compression Rate and Resource Allocation

```
Require: Amount of image data to offload $\{D_i\}_{i=1}^{I}$, Channel gain between users and edge servers $\{h_i\}_{i=1}^{I}$, Noise power spectral density $N_0$, Transmission delay constraints $t_0$; Parameters of compression rate and accuracy $\zeta_1, \zeta_2, \zeta_3, \zeta_4$.
Ensure: Compression rate $\{o_i\}_{i=1}^{I}$ of $u_i$, Wireless transmission bandwidth $\{b_i\}_{i=1}^{I}$, Transmit power ${p_i}_{i=1}^{I}$
Initialize the convergence threshold $\epsilon_1, \epsilon_2$ and the total iterations $n = 1$.
Repeat1
  Initialize compression rate iteration interval $\iota = 0.01$.
  for $u_i \in \mathcal{U}$ then
    Fix user bandwidth $b_i$ and transmit power $p_i$
    Initialize the optimal objective function value $T_{i,opt}^{cp}$ as 0 and the optimal compression rate $o_{i,opt}$
    for $o_k = 1\iota, 2\iota, \cdots, n\iota$ to 1 do
      Calculate the maximum success probability of inference tasks of user $u_i$: $T_{i,tmp}^{cp} \longleftarrow$ eq. (10)
      if $T_{i,tmp}^{cp} \geq T_{i,opt}^{cp}$ do
        $T_{i,opt}^{cp} = T_{i,tmp}^{cp}, o_{i,opt} = o_k$
  Fix compression ratio of each user $\boldsymbol{o}_{opt} = \{o_{i,opt}\}_{i=1}^{I}$
  Initialize the iteration number of the resource scheduling solution $k = 1$
  Initialize slack variables $\boldsymbol{b}_0, \boldsymbol{p}_0, \boldsymbol{f}_0, \boldsymbol{l}_0, \boldsymbol{y}_0, \boldsymbol{m}_0$ and $\boldsymbol{q}_0$
  Repeat2
    Solving problem P2.2 using the dual method
    According to $\boldsymbol{b}_k, \boldsymbol{p}_k, \boldsymbol{f}_k, \boldsymbol{l}_k, \boldsymbol{y}_k, \boldsymbol{m}_k$ and $\boldsymbol{q}_k$, calculate its objective function $T_k^{ra}$
  Until $|T_k^{ra} - T_{k-1}^{ra}| \leq \epsilon_2$
  According to the solved parameter $\boldsymbol{o}_{opt}, \boldsymbol{b}_k, \boldsymbol{p}_k, \boldsymbol{f}_k, \boldsymbol{l}_k, \boldsymbol{y}_k, \boldsymbol{m}_k, \boldsymbol{q}_k$, calculate the overall objective function $T_n^{all} \longleftarrow$ eq. (4)
Until $|T_n^{all} - T_{n-1}^{all}| \leq \epsilon_1$
```

ratios, the complexity of solving resource allocation subproblem (P2) is $O(L_3 3.5U)$, where dual decomposition are used for solving [16]. With the above analysis, the total complexity of Co-CRRA is $O(L_1 L_2 U + L_1 L_2 3.5U)$.

V. Spatial Attention-Aware Compression Strategy

To alleviate the loss of inference accuracy in image compression scenarios, we present a novel compression approach called SpaT-IMCO (Spatial Attention-aware Image Compression). The core idea of SpaT-IMCO is to use the deep vision model to evaluate the importance of visual features at different regions in an image, so as to retain the content of key areas as much as possible and compress less important information. To be specific, this compression strategy consists of two main components. Firstly, a spatial feature importance evaluation network is introduced, which employs a lightweight network structure to evaluate the importance of each region within the image. Subsequently, based on the spatial importance evaluation results, the image is compressed non-uniformly across different regions. The details will be introduced next.

A. Spatial Feature Importance Evaluation Network

In a DNN with loss function $\mathcal{L}$, the input consists of a N-pixel feature map represented by the pixel vector $x = \{x_1, x_2, \ldots, x_N\}$. The gradient of an individual input pixel x_i is denoted as $\rho_i = \bigtriangledown_{x_i}\mathcal{L} = \frac{\partial \mathcal{L}}{\partial x_i}$. By the definition of above gradient, when a small perturbation $\triangle x$ is added to the input image, the DNN loss changes as $\triangle \mathcal{L} = \rho_i \triangle x$. In fact, the essence of the image compression process is essentially similar to adding noise $\triangle x_i$ to the original image. Specifically, for each pixel x_i, let e_i represent the compression noise (which is equivalent to $\triangle x_i$) applied to the pixel x_i, and the corresponding loss change is $\triangle \mathcal{L}(x_i) = \rho_i \triangle x = \rho_i e_i$. To this end, the total loss change caused by the entire image compression is $\triangle \mathcal{L}_x = \sum_{i=1}^{N} \rho_i e_i$. This means that if the gradient value ρ_i of pixel x_i is large, even if e_i is small, it also will cause a significant increase in loss and decrease in inference accuracy. It can be seen that the value size of ρ_i reflects the importance of the corresponding pixel. If ρ is known in advance, we can preserve more detail of the image region with large ρ and impose a high compression rate on that with small ρ during data compression.

Here, this paper builds a spatial feature attention evaluation network to dynamically estimate the importance of image pixels. The evaluation network takes the original image as the input, and generates an attention weight matrix as the output which reflects the importance of each pixel. Let $Q = \{q_1, q_2, \ldots, q_N\}$ denote the attention weight matrix, where q_i represents the importance of pixel x_i at its spatial location. A higher attention weight value indicates that the image feature at that location has a stronger impact on the performance of model inference. The q_i corresponds one-to-one with the above-mentioned ρ_i. However, due to the absence of a dedicated dataset for training the evaluation network, we leverage vision DNN models trained on other tasks to create such datasets. The primary challenge lies in obtaining the Q matrix of each image, which serves as the label for the dataset.

Therefore, we employ the discriminative localization matrices of the gradient-weighted class activation mapping (Grad-CAM) [28] to generate the Q matrix. Specifically, let c represent the true category of the image, $\mathcal{A}_k$ denote the feature map of layer k, and $\mathcal{F}_k$ represent the global pooling output of $\mathcal{A}_k$. As shown in Fig. 2, the first step is to calculate the gradient value $\frac{\partial \mathcal{Y}_c}{\partial \mathcal{A}_k}$ of category c in the probability score $\mathcal{Y}_c$ for the corresponding feature map $\mathcal{A}_k$. Then, we perform global average pooling on these feature gradients at the corresponding spatial locations to obtain the neuron importance weight w_k^c. During the calculation of w_k^c, the back-propagation of activation gradients is a continuous product between the weight matrix and the gradient of the activation function. This calculation process continues until the gradient is propagated to the final feature layer. In fact, w_k^c represents a partial linearization of $\mathcal{A}_k$ that captures the importance of the k-th feature map in the target category. In this way, this w_k^c can be used as the attention weight matrix Q, that is the label of

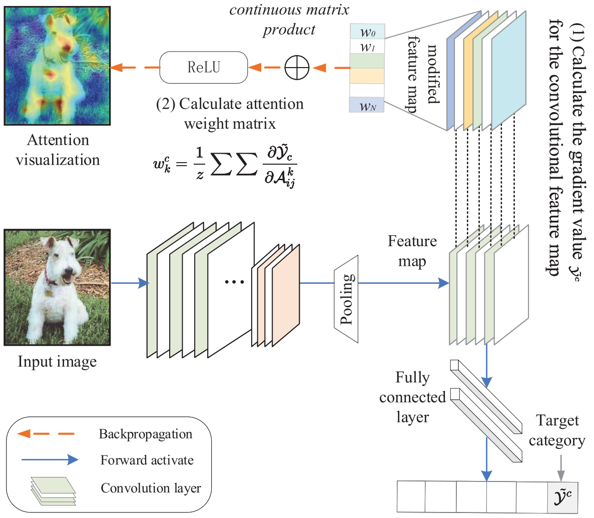
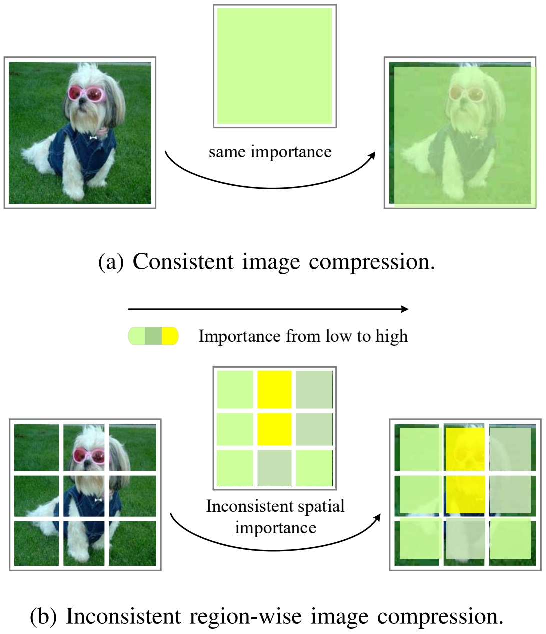

continuous matrix product
w_0
w_1
w_N
modified feature map
(1) Calculate the gradient value $\tilde{y}^c$ for the convolutional feature map
ReLU
(2) Calculate attention weight matrix
$w_k^c = \frac{1}{z}\sum\sum\frac{\partial\tilde{y}_c}{\partial\mathcal{A}_{ij}^k}$
Attention visualization
Input image
Pooling
Feature map
Fully connected layer
Target category
$\tilde{y}^c$
Backpropagation
Forward activate
Convolution layer

Fig. 2. The calculation process of attention weight matrixes for the spatial feature importance evaluation network.

the image. Based on the image datasets and constructed labels, we can efficiently train the spatial feature attention evaluation network. The training of this network is performed beforehand on a cloud server, and the well-trained network is subsequently deployed onto local devices for actual usage. Each input image to the evaluation network produces a corresponding spatial attention weight matrix as output.

Ideally, within a reasonable range, the deeper the evaluation network is, the more reliable the evaluation results become. However, it can be seen that if this model is large, this entire evaluation process will cause user devices to generate intensive computing at the same level as edge inference services. This computation cost cannot be supported by local devices. To adapt to the local computing environment, this paper designs a terminal-computable lightweight spatial feature importance evaluation network. We use depthwise separable convolutions and pointwise convolutions [29] to replace the traditional convolutions. The depthwise convolution operation applies a single convolution filter to each input channel to achieve lightweight filtering. The pointwise convolution operation calculates a linear combination of input channels to construct new features. In comparison, the computation cost of standard convolution is $h_l \cdot w_l \cdot D_l \cdot (d_j \cdot k^2)$, whereas it reduces to $h_l \cdot w_l \cdot D_l \cdot (d_j + k^2)$ when adopting the depthwise separable convolution operation. In addition, since the usage of ReLU activation transformation in low-dimensional space may filter out some valuable information, we insert a linear bottleneck module [29] after the convolution module to mitigate the loss of crucial features. Let θ_Q denote this evaluation network, and the spatial attention weight matrix of the input image x be represented as $Q = \{q_i, q_i = \theta_Q(x_i)\}_{i=1}^N$. The lightweight network only incurs very small additional computational overhead, which is suitable for efficient execution on mobile devices in edge network.



(a) Consistent image compression.

(b) Inconsistent region-wise image compression.

Fig. 3. Comparison of traditional unified image compression and spatial attention aware region-wise image compression.

B. Spatial Attention Aware Image Compression Strategy

Based on the spatial attention matrix Q and the given compression ratio o_{img}, our image compression strategy could adaptively select an appropriate compression rate for different regions of the input image. Fig. 3 shows the comparison of traditional unified image compression strategy and our region-wise compression strategy. The existing compression strategy uses the same compression rate to consistently compress each region of the image, which clearly lacks attention to regions with important content. And SpaT-IMCO strategy enables imposing different compression degrees for different visual regions based on their respective importance levels.

Accordingly, this paper establishes L different compression levels, and each level corresponds to a unique compression rate $\{s_l\}_{l=1}^L$. For image regions with low importance, it is preferred to compress them with a higher compression rate. Conversely, for some regions with high importance, we select the low compression rate to preserve key feature information as much as possible. The image or video frame is divided into multiple sub regions, each having the same size and number of pixels. Let $\{R_r\}_{r=1}^{|R|}$ denote the set of all sub regions, where $|R|$ is the number of regions. When encoding images, we use each image region rather than the entire image as the compression basic unit, each with individual compression rate.

The specific compression steps for our strategy are as follows: First, we calculate the average importance of each region based on the Q matrix, expressed as $q_{R_r} = \frac{1}{M}\sum q_i$. Among that, M is the number of pixel points in each region, and q_i is the attention weight of pixel x_i in the R_r region. Correspondingly, $Q_R = \{q_{R_r}\}_{r=1}^{|R|}$ represents the weight matrices of image sub-regions. Under the requirements of the overall image compression rate o_{img}, each region R_r selects a compression level s_{R_r} according to its corresponding attention weight q_{R_r}. The decision-making goal of image region compression rate is to make the average compression level s_{tar} of all sub-regions closest to the compression level

s_{img} of the entire image. This target compression level s_{img} is determined by the given o_{img}.

Here we take $L = 3$ as an example to illustrate our approach. There are three compression levels, denoted as s_1, s_2, s_3, which correspond to three compression ratios o_1, o_2, o_3, where $s_1 > s_2 > s_3$. The larger s_i, the greater o_i, and the more visual information loss. Assume that the number of regions selected compression levels s_1, s_2, s_3 are z_1, z_2, z_3, respectively, where $z_1 + z_2 + z_3 = |R|$. Thus, the overall compression level of the image is formulated as $s_{tar} = \frac{1}{|R|}\sum_{l=1}^{L}(s_l \cdot z_l)$, where $L = 3$. To solve the selection of $\{z_1, \ldots, z_L\}$, we define the objective function of region-wise compression using the square loss as follows:

$$\text{P3}: \min_{\boldsymbol{z}} \left(s_{img} - \frac{1}{|R|}\sum_{l=1}^{L} s_l \cdot z_l\right)^2 \tag{26}$$

$$\text{subject to} \quad z_1 + z_2 + z_3 = |R|, z_l \in \mathbb{R}. \tag{27}$$

It is evident that P3 is a typical integer programming problem. To determine the values of $z = \{z_1, \ldots, z_L\}$, we employ the branch and bound algorithm for solving. In detail, we sort the importance weight sequence Q_R in ascending order to obtain $\hat{Q}_R$, and subsequently use the corresponding region indices in the z sequence to calculate the compression levels and ratios for each sub-region. In a word, the spatial attention aware image compression strategy is summarized in Algorithm 2. Firstly, the attention weight matrix Q is calculated through the evaluation network θ_Q, and these image regions are compressed with different compression ratios.

Major complexity of SpaT-IMCO algorithm lies in solving optimal z_{opt}, which is $O(3.5L\log(1/\tau_z))$. Besides, relaxing the integer constraints and calculating optimization function require $2L$ computation iterations. The total computational complexity is $O(3.5L\log(1/\tau_z)+2L)$. In summary, Co-CRRA obtain optimal data compression rates and resource allocation schemes with the goal of maximizing the success probability of inference tasks, according to user request and network state; Then, under the given compression rate, SpaT-IMCO preserves key visual features as much as possible to reduce the accuracy drop caused by data compression.

Algorithm 2 Spatial-Attention Aware Compression Strategy

```
Require: Image region compression level of $L$ layer: $\{s_l\}_{l=1}^{L}$, Spatial feature importance evaluation network $\theta_Q$, Input images to offload $x$, Given overall picture compression ratio $s_{img}$, Number of pixels per region $M$
Ensure: Compression level set for $|R|$ image regions $\{s_{R_r}\}_{r=1}^{|R|}$
The image data $x$ is input to $\theta_Q$ to get the spatial attention matrix $Q = \{q_i\}_{i=1}^{N}$
for each image region $r = 1$ to $|R|$ then
    Calculate the average attention weight of each region $q_{R_r}$ according to each image region has $M$ pixels:
        $q_{R_r} = \frac{1}{M}\sum_{i=1}^{M} q_i, q_i = \theta_Q(x_i)$
Obtain the attention weights of all regions $Q_R = \{q_{R_r}\}_{r=1}^{|R|}$
Under relaxing the integer constraints of $z_l$, Computing the optimal solution $\hat{z}_l$ to the linear programming problem of P3
Let the relaxed solution of $\hat{z}_l^U$ be the upper bound solution $\hat{z}_l$, and the rounded solution of $\hat{z}_l^D$ be the lower bound solution $\hat{z}_l$
Select the largest variable in the fraction to branch, and calculate all feasible integer solution sets $\tau_z$
Initialize $\mathrm{P}_{opt} = +\infty$ and $z_{opt} = 0$
 for $z^* \in \tau_z$ then
    According to $z^*$, calculate the optimization function $P^* = (s_{img} - \frac{1}{|R|}\sum_{l=1}^{L} s_l \cdot z_l^*)^2 \longleftarrow$ eq. (26)
    if $P^* < \mathrm{P}_{opt}$ do
        $z_{opt} = z^*, \mathrm{P}_{opt} = P^*$
Obtain the optimal set of solution $z_{opt}$
Calculate the ascending sequence $\tilde{Q}_R$ of $Q_R$
Match the index of $\tilde{Q}_R$ according to $z_{opt}$, and determine $s_{R_r}$ of each $R_r$
```

VI. Experiment and Results

In this section, extensive experiments are designed to verify the effectiveness of the proposed architecture. First, we introduce the setting of simulation experiment platforms and vision inference tasks. Then, we compare the success probability of inference tasks under different algorithms to evaluate the performance of the Co-CRRA offloading algorithm. Finally, we analyze the effects of SpaT-IMCO compression strategy in different deep learning models.

A. Experiment Setup

1) Simulation Parameter Settings: In the simulation experiment, multiple users in the base station area are randomly distributed, and the parameters are mainly set according to the research [7], [30]. The inference delay constraint t_0 ranges from 50 to 100ms, while the noise power spectral density N_0 varies from -174dBm/Hz to -204dBW/Hz. The maximum network bandwidth $B_{\max}$ ranges from 15Mbps to 30Mbps, with a minimum bandwidth $B_{\min}$ of 5Mbps. The maximum transmit power $P_{\max}$ is set between 100mW and 200mW, and the minimum transmit power $P_{\min}$ is 10mW. The data compression ratio o_i ranges from 0 to 1, and the data size D_i^0 for inference services is between 10MB and 20MB. The poisson process is used to simulate the task arrival model. At each time slot, controller schedules these tasks in the request pool, so different devices may be offloaded simultaneously. Our edge inference mode is to directly transmit the compressed data to the edge server, and then perform inference tasks at the edge node. In the experiment, we take the image classification inference task as an example of vision inference to evaluate the effectiveness of the proposed Co-CRRA in improving the quality of edge inference service. Additionally, we analyze the adaptability of SpaT-IMCO to different deep learning models.

2) Spatial Feature Importance Evaluation Network Settings: This paper uses the pytorch framework to simulate the vision inference environment for achieving the compression offloading and resource allocation strategy. To

TABLE II
THE SPATIAL FEATURE IMPORTANCE EVALUATION NETWORK

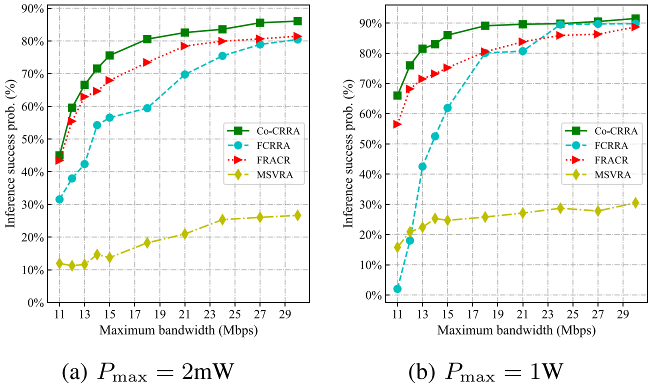
Input Size	Operation	Expansion factor	Channel nums	Bottleneck nums	Step
224×224	Conv2d dw	-	3	1	2
112×112	Bottleneck	1	32	1	1
112×112	Bottleneck	6	16	2	2
56×56	Bottleneck	6	24	3	2
28×28	Bottleneck	6	32	4	2
14×14	Bottleneck	6	64	3	1
14×14	Bottleneck	6	96	3	2
7×7	Bottleneck	6	160	1	1
7×7	Conv2d 1×1	-	320	1	1
7×7	Avgpool 7×7	-	1280	1	-
1×1	Conv2d 1×1	-	1280	-	-

be able to run the spatial feature importance evaluation network on local devices, we select MobileNet [29] to build it. The specific backbone network structure is shown in Tab. II. For the data compression process, we divide each image into image region blocks with a size of 8 × 8 pixels, and then perform our compression strategy on the image blocks with different compression ratios. This paper mainly uses two key indicators, the success probability of inference tasks and image classification accuracy to evaluate different algorithms. Notably, the number of parameters and computation complexity of this evaluation network are 3.5MB and 318M MAdds, respectively. And we take NVIDIA Jetson TX2 (JTX2) as mobile devices, the execution time of this network ranges from 68ms to 73ms by several experimental tests, which is used in Section VI-D. It can be seen that this evaluation network is of low complexity and can support the execution on mobile devices to obtain Q which guides inconsistent compression of images.

B. Analysis of Compression Offloading Algorithm

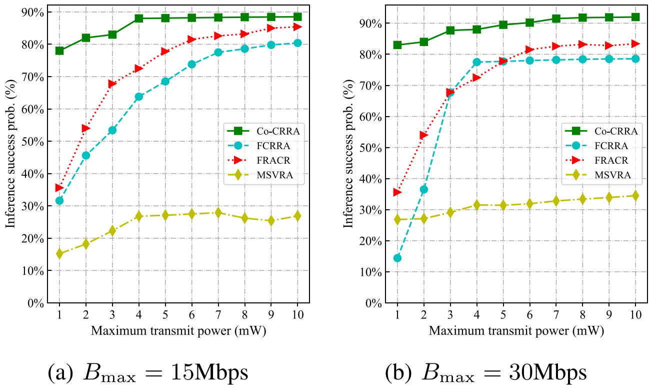
A typical compression offloading algorithm encodes the image through the compression algorithm and transmits it to edge servers for inference. In the experiment, three baseline algorithms are selected to compare with our offloading algorithm: the maximizing the system velocity for resource allocation (MSVRA), the fixed compression ratio for resource allocation (FCRRA) and the fixed resource allocation for compression ratio decision (FRACR).

1) Relationship Between the Success Probability and Maximum Bandwidth: Following the above settings, Fig. 4(a) and Fig. 4(b) respectively show the influence of the maximum bandwidth value on the success probability of inference tasks under $P_{\max}$=2 mW and $P_{\max}$=1 W. Experimental results show that with the gradual increase of the maximum bandwidth value, the success probability of DNN inference tasks also gradually increases, and finally converges to a certain threshold. Under $P_{\max} = 2$mW and $P_{\max} = 1$W, the success probability can reach about 87% and 91%, respectively. Compared with the three algorithms MSVRA, FCRRA and FRACR, the average success probability increases by 25.1% when $P_{\max} = 2$mW, and by 32.1% when $P_{\max} = 1$W, respectively. Notably, above performance gain ratio of the average success probability is the average value of performance gain ratios of Co-CRRA compared with each of the three benchmark algorithms. Without special notes, the average performance gain afterward is calculated in the same way. Since the limitation of the maximum bandwidth, it is necessary to compress data with smaller compression rates for improving the success probability of the vision model inference and reducing the transmission delay of the edge system. Especially when $B_{\max}$ is relatively small and resource competition is more intense, there are greater requirements for the data compression ratio and resource allocation decision. Co-CRRA algorithm in this paper has significant performance gain, where MSVRA has the worst performance. This is because MSVRA only focuses on the resource allocation optimization and lacks the consideration for compression decisions.



(a) $P_{\max} = 2$mW (b) $P_{\max} = 1$W

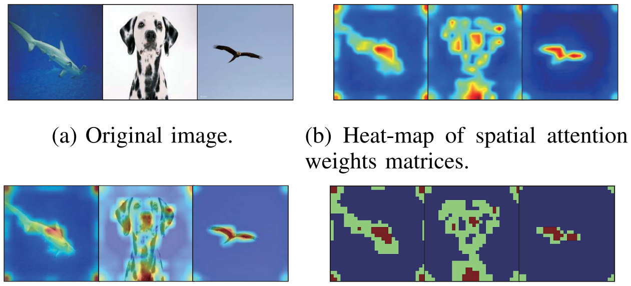
Fig. 4. The influence of the maximum network bandwidth on the average success probability of inference tasks.



(a) $B_{\max} = 15$Mbps (b) $B_{\max} = 30$Mbps

Fig. 5. The influence of the maximum transmit power on the average success probability of inference tasks.

2) Relationship Between the Success Probability and Maximum Transmit Power: In addition, we analyze the impact of the maximum transmit power. Fig. 5(a) and Fig. 5(b) show the changes in the success probability of inference tasks with the change of the maximum transmit power under $B_{\max} = 15$MHz and $B_{\max} = 30$MHz respectively. In the transmit power change from 1mW to 10mW, the success probability value can still be maintained between 80%-90%. Higher signal transmission power means that the greater the radiation energy, the farther the communication distance will



(a) Original image.

(b) Heat-map of spatial attention weights matrices.

(c) Images with the heat-map of spatial attention weights.

(d) Heat-map of the compression level of region-wise image.

Fig. 6. An example of spatial attention weight information.

be. To this end, the number of success inference tasks increases as data transmission rate in communication increases. The highest inference success probabilities for $B_{\max} = 15\text{Mbps}$ and $B_{\max} = 30\text{Mbps}$ can reach about 88.5% and 92.0%, respectively. Compared with the three baseline algorithms, in the two cases of $B_{\max}$, the system performance can be improved by 13.5%∼61.7% and 16.8%∼57.7% respectively, both of which increase by an average of about 32%. It can be observed that compared with the three baseline algorithms, our algorithm has notable decision-making advantages. Compared with the maximum bandwidth, the maximum transmit power has a greater impact on the inference success probability. Overall, the experimental results show that the Co-CRRA algorithm is suitable for edge inference scenarios with limited network resources.

C. Performance Analysis of Algorithm Parameters

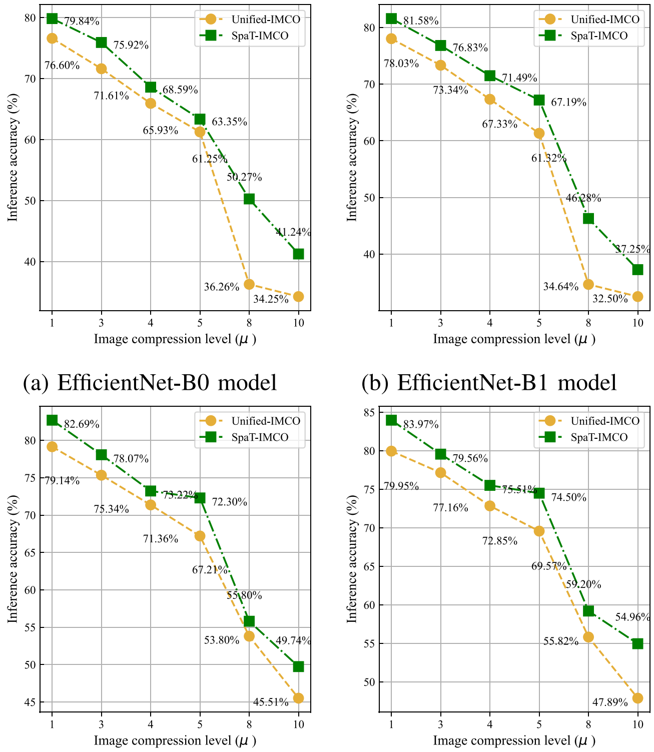
1) Image Spatial Attention Evaluation: In this experiment, we select the well-trained vision Transformer model to generate the attention weight of the ImageNet dataset [23] according to the strategy shown in Fig. 2. Specifically, we extract the multi-head attention weight matrices for each block within the vision network model, denoted as $\mathcal{J}_b \in R^{m \times N \times N}$. Here, b refers to the number of layers, m indicates the number of attention heads, and N represents the number of Tokens. These matrices are merged using a maximum value strategy, resulting in $\max(\mathcal{J}_b) \in R^{N \times N}$. By successively multiplying the feature maps of each block, we calculate the final spatial attention matrix Q' for the original image.

Fig. 6 shows an illustration of the generated spatial attention weight information. We utilize the image data and their respective attention weights as the training dataset to train our spatial feature importance evaluation network. Fig. 6(a) is the original images. Fig. 6(b) displays the heat-map corresponding to the attention matrix of the image shown in Fig. 6(a), and Fig. 6(c) is the image with the heat-map of spatial attention weights. Next, we compute upper and lower bounds for each importance level in our compression strategy based on the range of attention weight values. According to the pseudocode of SpaT-IMCO algorithm (see Algorithm 2) to determine the compression level of each sub-region of the image, that is $\{s_{R_r}\}_{r=1}^{|R|}$. Finally, here we take $L = 3$ as an example to calculate the importance level of each region of the original image. Fig. 6(d) is a heat map of the corresponding importance level after the image is divided into each region. Among them, red indicates the greatest weight level, where the lowest compression level will be set; blue has the least importance, which will compress the area with a greater compression ratio.

2) Comparison of Different Compression Strategies: As a basic image compression algorithm, JPEG first converts the RGB color space into a brightness-color-difference model, and then transforms the spatial domain information into the frequency domain through the discrete cosine transform operation. JPEG is a lossy compression method, and the main loss step is data quantization. It uses the quantization matrix K_Y, K_{Cb} and K_{Cr} to calculate the characteristics of the brightness space and the color difference space respectively, formulated as $\tilde{I} = \text{round}(\frac{I}{K}), K \in \{K_Y, K_{Cb}, K_{Cr}\}$. This quantization operation can distinguish important data and unimportant data, where unimportant data will be converted to zero to achieve compressed storage. During the actual compression process, a coefficient μ is often multiplied on the quantization matrix, that is $\tilde{K} = \mu * K$. By adjusting the size of μ to make more or less data become 0, so as to control the degree of image compression. The flow of our compression algorithm is basically the same as that of JPEG. The difference is that JPEG sets a uniform μ for the entire image (termed as Unified-IMCO), while the SpaT-IMCO algorithm sets an individual μ for each image region. In the experiment, we construct a dataset of compressed images in jpg format with μ ranging from 1 to 10, and compare the inference accuracy of the proposed SpaT-IMCO algorithm with the benchmark Unified compression algorithm at multiple compression levels.

In order to evaluate the performance of SpaT-IMCO strategy, this paper uses two deep learning models, the EfficientNet network based on CNNs and the DeiT network based on Transformer, to analyze the inference accuracy of compressed images on ImageNet dataset. Fig. 7 shows the comparison results of the compression algorithms under the networks of EfficientNet-B0, EfficientNet-B1, EfficientNet-B2 and EfficientNet-B4, where the model sizes are 5.3 MB, 7.8 MB, 9.2 MB and 12 MB respectively. For EfficientNet-B0 and EfficientNet-B1, compared to the baseline algorithm, the inference accuracy of the two models can be improved by 0.66%∼14.0% and 0.16%∼11.6% respectively. Our algorithm reduces the accuracy loss by 14.0% and 11.6% under $\mu = 8$, respectively. Except for $\mu = 3$ and 4, the accuracy improvement under other quantization coefficient settings all exceeds 3.0%. In addition, the average performance of EfficientNet-B2 and EfficientNet-B4 under different compression levels can be improved by 1.9% and 3.1% respectively.

Fig. 8 shows comparison results of the compression algorithms under the networks of Transformer-based DeiT-Tiny and DeiT-Small, whose model sizes are 5.7 MB and 22.1 MB, respectively. At different compression levels, the inference accuracy of these two models can be improved by 0.4∼9.6 points and 0.6∼14.8 points respectively. Under the same compression quantization factor, our proposed compression algorithm can mitigate the inference accuracy loss caused by data compression. Especially on quantization coefficients $\mu = 8$ and $\mu = 10$ with high compression rates, the



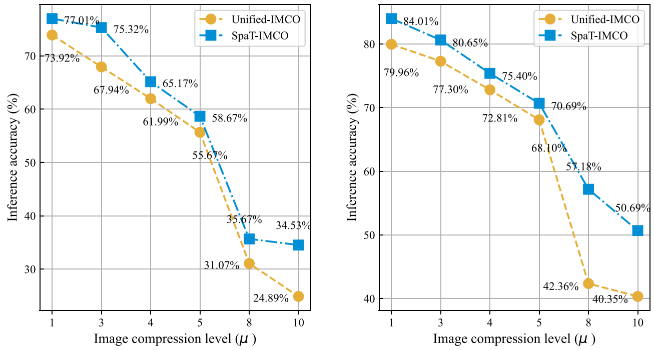
(a) EfficientNet-B0 model

(b) EfficientNet-B1 model

(c) EfficientNet-B2 model

(d) EfficientNet-B4 model

Fig. 7. Comparison of SpaT-IMCO algorithms with traditional Unified compression algorithms on EfficientNet Networks.



(a) DeiT-Tiny model

(b) DeiT-Small model

Fig. 8. Comparison of SpaT-IMCO algorithms with traditional Unified compression algorithms on DeiT Networks.

compression algorithm can significantly improve inference accuracy, where the accuracy gains are respectively 14.8% and 38.7% on DeiT-Tiny, and 35.0% and 25.6% on DeiT-Small. Under all quantization coefficients, DeiT-Tiny model improves the accuracy by an average of 4.6% and 9.6%, and DeiT-Small model improves the accuracy by an average of 14.8% and 10.3%, respectively. Overall, these results demonstrate that the SpaT-IMCO algorithm can achieve higher inference accuracy at the same compression level compared to the Unified compression algorithm.

In the above experiments, we discuss the performance of image compression algorithms on inference accuracy. Both the Unified-IMCO algorithm and the SpaT-IMCO algorithm utilize the compression quantization coefficient μ to adjust the compression rate. Therefore, this section further analyzes the relationship between the quantization coefficient μ and the compression ratio, as well as the image size, as illustrated in Fig. 9. The experimental results show that, under the same quantization coefficient, our compression algorithm can generate images with a higher compression ratio and smaller storage capacity than Unified-IMCO algorithm. Correspondingly, this also implies that less communication resources are required in the compression offloading network. Without considering other factors, when the compression ratios are around 0.2 and 0.45, our compression algorithm can save approximately 10% of bandwidth consumption. It is worth noting that this performance improvement solely comes from this novel compression strategy. In a word, these extensive results demonstrate its effectiveness under different compression levels.

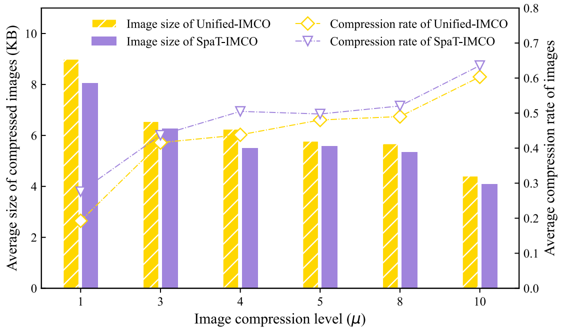
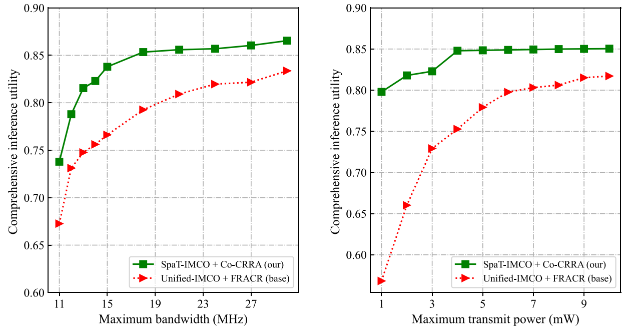


Fig. 9. The relationship between the quantization coefficient μ and the compression ratio, image size.

D. Analysis of End-to-End Performance

Above experiments demonstrate the efficiency of Co-CRRA and SpaT-IMCO algorithms, respectively. In this section, we analyze the inference and delay performance of end-to-end edge service architecture combining these two algorithms. Specially, we use JTX2 as mobile device and laptops as edge servers. We design the simulation system with reference to real experimental data and network conditions.

1) Inference Performance Analysis: We adopt a new weighted metric to comprehensively evaluate inference performance of overall system, incorporating the two metrics of the success probability of inference tasks and inference accuracy, called as inference utility: $\alpha * \sum \Phi_i + (1 - \alpha) * \mathrm{Acc}$. In our case $\alpha = 0.5$ and the inference model takes EfficientNet-B1. Since FRACR algorithm is the most competitive offloading algorithm comparing with ours (in Section VI-B), combination of Unified-IMCO and FRACR is used as baseline of this end-to-end experiment. As shown in Fig. 10, our scheme (SpaT-IMCO + Co-CRRA) improves the comprehensive inference performance by an average of 7.1% and 14.7% under $P_{\max} = 1\text{W}$ and $B_{\max} = 15\text{MHz}$, respectively. In fact, this result is predictable because our image compression strategy and offloading algorithm have proved to outperform both baseline algorithms in previous experiments, respectively.



(a) $P_{\max} = 1$W, the effect on the(b) $B_{\max} = 15$MHz, the effect on

Fig. 10. Comparison of end-to-end inference utility between our scheme and the baseline.

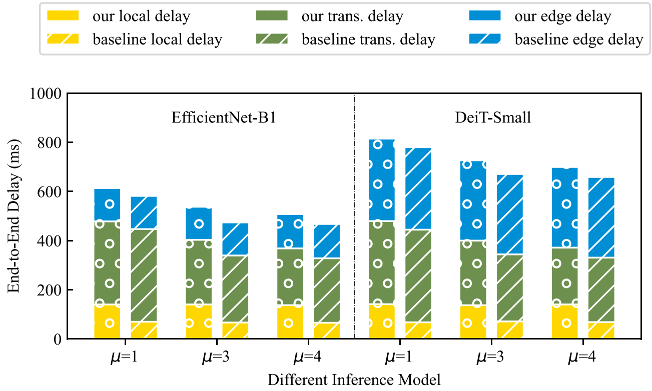


Fig. 11. Comparison of end-to-end system delay between our scheme and the baseline.

2) Inference Delay Analysis: Besides, we investigate end-to-end latency under different deep learning models and different compression levels. Latency of the overall system mainly includes three parts here: i) local delay, executing the evaluation network and the compression algorithm on mobile devices to compress data; ii) transmission delay, uploading compressed data from local to edge servers; iii) edge delay, executing deep vision models on edge servers and obtaining inference results. Among them, we compute the communication delay by dividing transferred data by the available data rate, which sets as 15Mbps. Fig. 11 shows the breakdown of the end-to-end latency. From this figure, our scheme has a little higher local latency, while the transmission latency of baseline gets a little higher. For the overall delay, ours is almost comparable to the baseline. When the compression level is small, ours is slightly lower. When the compression level is larger, it is reversed. In general, our compressed offloading scheme can achieve higher the success probability of inference tasks and inference accuracy without increasing latency.

VII. Conclusion

This paper investigates the edge vision inference framework driven by data compression. In this framework, vision data is compressed on local device to become smaller, and then is transmitted to edge sever for intelligent inference services. Specifically, we establish the collaborative optimization model of compression offloading and resource allocation. This model maximizes the success probability of inference tasks of the edge vision system, under the constraints of network bandwidth and transmit power. By introducing slack variables and Taylor expansion techniques, we decompose the non-convex optimization model into two sub-problems to solve the compression ratio and resource allocation scheme. In addition, in order to improve the inference performance of compression offloading, we propose a spatial-attention aware image compression strategy. Our compression strategy uses the well-trained vision DNN model to train a lightweight spatial feature importance evaluation network. Based on importance results from this evaluation network, we enable varying degrees of compression for different image regions to adaptively compress vision data. The experimental results demonstrate that our proposed offloading model and compression strategy outperform other algorithms, achieving an average reduction of 11.3% in total system energy consumption.

References

[1] W. Xiao, M. Li, B. Alzahrani, R. Alotaibi, A. Barnawi, and Q. Ai, "A blockchain-based secure crowd monitoring system using UAV swarm," *IEEE Netw.*, vol. 35, no. 1, pp. 108–115, Jan./Feb. 2021.

[2] K. Han et al., "A survey on vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023.

[3] H. J. Park, H. W. Kim, and S. H. Chae, "Deep-learning-based resource allocation for transmit power minimization in uplink NOMA IoT cellular networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 3, pp. 708–721, Jun. 2023.

[4] Y. Hao, L. Hu, and M. Chen, "Joint sensing adaptation and model placement in 6G fabric computing," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 7, pp. 2013–2024, Jul. 2023.

[5] R. Kishore, S. Gurugopinath, S. Muhaidat, P. C. Sofotasios, M. Dianati, and N. Al-Dhahir, "Energy efficiency analysis of collaborative compressive sensing scheme in cognitive radio networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 3, pp. 1056–1068, Sep. 2020.

[6] R. Luo, H. Jin, Q. He, S. Wu, and X. Xia, "Cost-effective edge server network design in mobile edge computing environment," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 4, pp. 839–850, Oct.–Dec. 2022.

[7] J. Liang, H. Xing, F. Wang, and V. K. Lau, "Joint task offloading and cache placement for energy-efficient mobile edge computing systems," *IEEE Wireless Commun.*, vol. 12, no. 4, pp. 694–698, Apr. 2023.

[8] R. Wang, J. Wang, Y. Hao, L. Hu, S. A. Alqahtani, and M. Chen, "C3Meta: A context-aware cloud-edge-end collaboration framework toward green metaverse," *IEEE Wireless Commun.*, vol. 30, no. 5, pp. 144–150, Oct. 2023.

[9] Y. Zheng, T. Zhang, R. Huang, and Y. Wang, "Computing offloading and semantic compression for intelligent computing tasks in MEC systems," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2023, pp. 1–6.

[10] Y. Hao, J. Wang, D. Huo, N. Guizani, L. Hu, and M. Chen, "Digital twin-assisted URLLC-enabled task offloading in mobile edge network via robust combinatorial optimization," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3022–3033, Oct. 2023.

[11] X. Xie and K.-H. Kim, "Source compression with bounded DNN perception loss for IoT edge computer vision," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, 2019, pp. 1–16.

[12] X. Xiao, J. Zhang, W. Wang, J. He, and Q. Zhang, "Dnn-driven compressive offloading for edge-assisted semantic video segmentation," in *Proc. IEEE Conf. Comput. Commun.*, 2022, pp. 1888–1897.

[13] R. Luo, H. Jin, Q. He, S. Wu, and X. Xia, "Enabling balanced data deduplication in mobile edge computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 5, pp. 1420–1431, May 2023.

[14] Y. Matsubara, R. Yang, M. Levorato, and S. Mandt, "Supervised compression for resource-constrained edge computing systems," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 2685–2695.

[15] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.

[16] T. T. Nguyen, V. N. Ha, L. B. Le, and R. Schober, "Joint data compression and computation offloading in hierarchical fog-cloud systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 293–309, Jan. 2020.

[17] G. Pan, H. Zhang, S. Xu, S. Zhang, and X. Chen, "Joint optimization of video-based AI inference tasks in MEC-assisted augmented reality systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 2, pp. 479–493, Apr. 2023.

[18] J.-B. Wang, J. Zhang, C. Ding, H. Zhang, M. Lin, and J. Wang, "Joint optimization of transmission bandwidth allocation and data compression for mobile-edge computing systems," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2245–2249, Oct. 2020.

[19] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 10697–10706.

[20] M. Chiani, D. Dardari, and M. Simon, "New exponential bounds and approximations for the computation of error probability in fading channels," *IEEE Trans. Wireless Commun.*, vol. 2, no. 4, pp. 840–845, Jul. 2003.

[21] M. Chen, A. Liu, N. N. Xiong, H. Song, and V. C. M. Leung, "SGPL: An intelligent game-based secure collaborative communication scheme for metaverse over 5G and beyond networks," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 3, pp. 767–782, Mar. 2024.

[22] H. Jin, R. Luo, Q. He, S. Wu, Z. Zeng, and X. Xia, "Cost-effective data placement in edge storage systems with erasure code," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 1039–1050, Mar./Apr. 2023.

[23] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[24] X. Chen, J.-N. Hwang, D. Meng, K.-H. Lee, R. L. de Queiroz, and F.-M. Yeh, "A quality-of-content-based joint source and channel coding for human detections in a mobile surveillance cloud," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 1, pp. 19–31, Jan. 2017.

[25] X. Xu et al., "Task-oriented and semantic-aware heterogeneous networks for artificial intelligence of things: Performance analysis and optimization," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 228–242, Jan. 2024.

[26] A. Liu, V. K. N. Lau, and B. Kananian, "Stochastic successive convex approximation for non-convex constrained stochastic optimization," *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4189–4203, Aug. 2019.

[27] X. Wang et al., "Wireless powered mobile edge computing networks: A survey," *ACM Comput. Surv.*, vol. 55, no. 13, pp. 1–37, Jul. 2023.

[28] M.-H. Guo et al., "Attention mechanisms in computer vision: A survey," *Comput. Vis. Media*, vol. 8, no. 3, pp. 331–368, 2022.

[29] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[30] A. Hekmati, P. Teymoori, T. D. Todd, D. Zhao, and G. Karakostas, "Optimal mobile computation offloading with hard deadline constraints," *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2160–2173, Sep. 2020.

Wenjing Xiao (Member, IEEE) received the B.S. degree from the School of Computer Science and Technology, North China Electric Power University, Hebei, China, in 2018, and the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2023. She is a Research Assistant Professor with the School of Computer, Electronic and Information, Guangxi University. She has 20+ publications, including eight IEEE Transaction/Journal papers. Her Google Scholar Citations reached 300+ with a H-index of 12. Her research interests include edge computing, deep learning, Internet of Things, blockchain, and wireless sensor network.

Yixue Hao (Member, IEEE) received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 2017, where he is an Associate Professor with the School of Computer Science and Technology. His current research interests include 5G network, Internet of Things, edge computing, edge caching, and cognitive computing. He was selected as a Highly Cited Researcher in 2020.

Junbin Liang received the B.E. and M.S. degrees from Guangxi University, Nanning, China, in 2000 and 2005, respectively, and the Ph.D. degree from Central South University, Changsha, China, in 2010. He was a Visiting Professor with The University of British Columbia from 2019 to 2020. He is currently a Professor with Guangxi University. His research interests include sensor-cloud systems, fog computing, and distributed computing.

Long Hu is an Assistant Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, China. He was a visiting student with the Department of Electrical and Computer Engineering, The University of British Columbia from August 2015 to April 2017. His research includes the Internet of Things, software defined networking, caching, 5G, body area networks, body sensor networks, and mobile cloud computing.

Salman A. Alqahtani is currently a Full Professor with the Department of Computer Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. He also serves as a Senior Consultant in computer communications, integrated solutions, and digital forensics for few development companies, and government sectors in Saudi Arabia. His main research interests include radio resource management for wireless and cellular networks (4G, 5G, the IoT, Industry 4.0, and digital forensics).

Min Chen (Fellow, IEEE) is a Full Professor with the School of Computer Science and Engineering, South China University of Technology. He is also the Director of Embedded and Pervasive Computing Lab, Huazhong University of Science and Technology. He was an Assistant Professor with the School of Computer Science and Engineering, Seoul National University, before he joined HUST. He is the Chair of IEEE Globecom 2022 eHealth Symposium. His Google Scholar Citations reached 42,800+ with an H-index of 96. His top paper was cited 4,400+ times. He received the IEEE Communications Society Fred W. Ellersick Prize in 2017, the IEEE Jack Neubauer Memorial Award in 2019, and the IEEE ComSoc APB Oustanding Paper Award in 2022. He was selected as a Highly Cited Researcher from 2018 to 2023. He is the Founding Chair of IEEE Computer Society Special Technical Communities on Big Data. He is the Chair of IEEE Globecom 2022 eHealth Symposium. He is a Fellow of IET.