Enabling Sustainable and Unmanned Facial Detection and Recognition Services With Adaptive Edge Resource

Zhengzhe Xiang, *Member, IEEE*, Xizi Xue, Zengwei Zheng, Honghao Gao, *Senior Member, IEEE*, Yuanyi Chen, and Schahram Dustdar, *Fellow, IEEE*

***Abstract*—Facial recognition techniques are used extensively in areas like online payments, education, and social media. Traditionally, these applications relied on powerful cloud-based systems, but advancements in edge computing have changed this, enabling fast and reliable local processing in complex and extreme environments. However, new challenges arise in availability and durability insurance to make the system run 24/7 with acceptable performance. This paper proposes a novel solution to these challenging settings. First, we use edge devices for local data processing, reducing the need for cloud communication and enhancing user privacy. Second, we implement an adaptive control strategy to improve energy management in these devices. Lastly, we establish a solar-powered energy system to facilitate long-term device operation. The experiments show our approach strikes a balance between performance, quality, and durability, enabling facial recognition systems to work energy-efficiently in complex environments. Meanwhile, considering the limited resources of devices in extreme cases, we also proposed a learning-based approach to accelerate the solution generation.**

***Index Terms*—Edge computing, service management, resource allocation, energy harvesting.**

I. Introduction

THE INDUSTRIAL revolution has undergone a transformative journey, evolving from Industry 1.0 to Industry 4.0, marking a shift from mechanized production to the fusion of automation and informatization. Central to this evolution has been the persistent pursuit of enhancing production efficiency and quality within the industrial sector. However, as consumer demands diversify and emphasize production flexibility and sustainability, there is a growing urgency for the industry to meet these evolving needs. This demand has catalyzed a continuous pursuit of innovation and improvement, with Industry 5.0 emerging as a prominent concept garnering widespread attention [1].

In the vision of Industry 5.0, close collaboration among workers, robots, and automation systems is paramount to achieving personalized and highly efficient production processes. This collaborative model not only enhances production efficiency but also fosters collective learning and innovation. To materialize this vision, Industry 5.0 relies heavily on the integration of advanced technologies such as the Internet of Things (IoT), artificial intelligence (AI), and edge computing [2], [3]. IoT technology enables real-time data collection and exchange among production equipment, AI supports intelligent data analysis and decision-making, and edge computing ensures real-time data processing and security. By synergizing these technologies collaboratively, a robust technological foundation can be laid for the implementation of Industry 5.0.

Among the various technologies, face recognition stands out as pivotal in Industry 5.0. For instance, cameras with face recognition capabilities bolster security management in factories by enabling swift identity verification. Additionally, precise facial recognition allows systems to assign tasks based on workers' skills and experience, significantly enhancing production line efficiency. In consumer electronics, facial recognition simplifies transactions, while in education, it verifies student identities and prevents cheating [4]. These applications showcase the versatility and importance of facial recognition in modern industries.

Traditionally, facial recognition relied on cloud-based processing, which presented challenges such as data transmission delays and overreliance on central servers. However, the advent of edge computing has revolutionized this paradigm, bringing processing tasks closer to data sources and end-users, thereby enhancing system responsiveness and reliability [5]. This shift enables the deployment of facial recognition models on edge devices, offering a more efficient and flexible solution tailored to Industry 5.0.

While wired connections suffice for routine tasks, complex environments reliant on wireless networks and battery power

Manuscript received 7 May 2024; revised 26 June 2024 and 4 August 2024; accepted 11 August 2024. Date of publication 19 August 2024; date of current version 14 August 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62102350 and Grant 62072402; in part by the National Key Research and Development Program of China under Grant 2022YFF0902500; and in part by the Supercomputing Center of Hangzhou City University. *(Corresponding authors: Honghao Gao; Yuanyi Chen.)*

Zhengzhe Xiang, Zengwei Zheng, and Yuanyi Chen are with the School of Computer and Computing Science, Hangzhou City University, Hangzhou 310015, China (e-mail: xiangzz@hzcu.edu.cn; zhengzw@hzcu.edu.cn; chenyuanyi@hzcu.edu.cn).

Xizi Xue is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: xuexizi@zju.edu.cn).

Honghao Gao is with the School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China, and also with the College of Future Industry, Gachon University, Seongnam 1342, South Korea (e-mail: gaohonghao@shu.edu.cn).

Schahram Dustdar is with the Distributed Systems Group, TU Wien, 1040 Vienna, Austria, and also with the Department of Engineering, Universitat Pompeu Fabra, 08002 Barcelona, Spain (e-mail: dustdar@dsg.tuwien.ac.at).

Digital Object Identifier 10.1109/TCE.2024.3445435

pose challenges. Developing portable, energy-efficient edge facial recognition systems for such environments is essential. This necessitates a focus on sustainability, aligning with Industry 5.0's core principle of environmental protection and resource conservation.

In practical applications, facial recognition systems in complex environments prioritize system availability and durability over strict real-time performance. These systems must exhibit exceptional durability for stable performance during extended operations, ensuring continuous and stable operation in demanding contexts.

Therefore, it is crucial to fully consider the different requirements of various application scenarios and flexibly use suitable algorithms and technologies to balance key metrics such as real-time performance, availability, and durability, providing efficient and reliable solutions for facial recognition applications in complex environments. We aim to achieve this by addressing the following contributions:

1) *Utilizing MEC Paradigm*: By processing data locally on edge devices, we can increase data density, reduce transmission time and costs, improve response times.

2) *Implementing Adaptive Work-Sleep Strategies*: Adjusting device work and sleep frequencies smartly can maintain system performance, reduce energy consumption, and extend battery life, crucial for stable and long-term operation in complex environments.

3) *Establishing Solar-Electricity-Based Energy Prototypes*: For systems in areas with limited access to traditional power, such as remote locations, we propose using solar power. By managing energy collection, storage, and release, devices can operate continuously and stably, optimizing energy usage and extending device lifespans.

Through a series of comparative and single-factor experiments, we have demonstrated that our approach significantly enhances the availability of the system and equipment without much compromising in real-time performance and revealed the factors that may effect the results of our proposed approach.

II. Related Work

A. Energy Efficiency in Edge Computing

Recent advancements in edge computing have greatly improved energy efficiency across various applications. Researchers have focused on optimizing energy usage in edge devices, such as autonomous drones and IoT devices, resulting in reduced energy consumption and improved performance. Navardi et al. introduced the E2EdgeAI approach for energy-efficient deployment of vision-based DNNs on drones [6]. Liu et al. proposed techniques like temporal sparsity to reduce computations in Tiny Machine Learning networks on edge devices [7]. Jiang et al. developed an energy efficiency improvement technology for numerical control machines using an ant colony algorithm [8]. Chen et al. designed a system energy consumption model, which takes into account the runtime, switching, and computing energy consumption of all participating servers and IoT devices [9]. Dai et al. used Deep Reinforcement Learning for energy-efficient computation in 5G networks [10]. Mao et al. studied the tradeoff between Energy Efficiency and delay in a multi-user wireless powered MEC system, formulating a stochastic optimization problem to investigate the EE-delay tradeoff [11]. Lakew et al. investigated integrating renewable energy sources into edge computing [12]. These efforts highlight ongoing innovations in energy efficiency in edge computing.

B. Service Performance Management

Achieving real-time performance in facial recognition systems while ensuring system durability requires a balanced approach. Su et al. emphasized the need for balancing processing speed and energy consumption [13]. Wen et al. conducted studies on algorithm effectiveness in edge-based facial recognition systems [14]. Khazaei et al. developed performance models for cloud-based microservices platforms [15]. Mahmoudi and Khazaei contributed analytical models for serverless computing platforms [16]. Wen et al. introduced a Service Path Performance Monitoring Scheme for network performance management [17]. Singh et al. focused on QoS optimization in IoT-smart agriculture [18]. Alizadeh and Tabassum used deep learning for power control with QoS guarantees [19]. Zhang et al. proposed a multi-target detection neural network that can improve the performance of vulnerability detection services [20]. Yang et al. optimized the service deployment of AI robots by improving data freshness utility [21]. An et al. enhanced visual coding through collaborative perception, making services more adaptable in complex scenes [22]. These studies collectively contribute to improving service performance management, especially in intelligent service systems.

However, the aforementioned works either only consider reducing system energy consumption by improving network structure or building energy optimization models, etc., or focus on improving system durability by improving service performance in specific scenarios. Our work takes the energy utilization efficiency, system real-time and durability of edge computing into account from multiple perspectives: on the one hand, we take advantage of edge computing's local processing of data, which reduces the time and cost of data transmission; on the other hand, we analyze the characteristics of the model and incorporate them into the energy consumption model, dynamically adjusting the device's operating frequency in order to maintain the system's performance. In addition, we built a solar-based energy prototype system to optimize energy usage and extend device lifetime by managing energy collection, storage, and release.

III. Problem Description and Formulation

A. System Model

Face recognition typically involves two primary stages [23]:

1) Face Detection and Alignment: Typically, deep learning methods like RetinaFace [24] or CenterFace [25] are used to detect faces in an image. This involves identifying facial features like eyes, noses, and mouths. Sometimes, image processing techniques are used to align faces by adjusting their size, rotation, and scaling.

TABLE I
COMPARISON OF RELATED WORKS

Work	Portability	Renewability	Timeliness	Energy-Efficiency	Dynamic	Prototype	Sync/Async
Navardi et al. [6]	√	×	√	√	×	√	S
Liu et al. [7]	√	×	×	√	×	×	S
Jiang et al. [8]	×	×	√	√	×	×	S
Chen et al. [9]	×	×	√	√	×	×	S
Dai et al. [10]	×	×	√	√	√	×	S
Mao et al. [11]	√	√	√	√	√	×	S
Lakew et al. [12]	√	√	√	√	√	×	S
Su et al. [13]	×	×	√	√	×	×	S
Wen et al. [14]	×	×	×	√	×	×	S
Khazaei et al. [15]	×	×	×	×	×	√	S
Mahmoudi et al. [16]	×	×	×	×	×	√	S
Wen et al. [17]	×	×	×	×	×	√	S
Singh et al. [18]	×	√	√	√	×	√	S
Alizadeh et al. [19]	×	×	×	√	×	√	S
Zhang et al. [20]	×	×	×	×	×	√	S
Yang et al. [21]	×	×	×	×	×	×	S
An et al. [22]	×	×	×	×	×	√	S
Ours	√	√	√	√	√	√	A

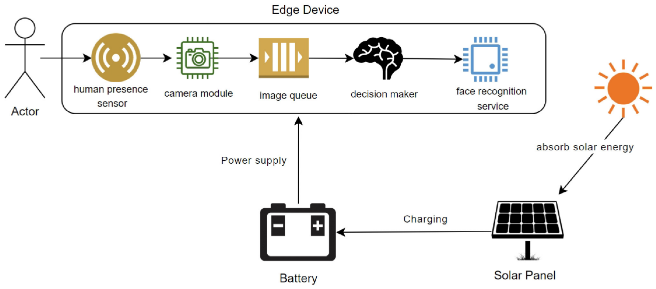


Fig. 1. Workflow of the renewable energy based edge system.

2) Face Matching: Aligned face images are then matched to determine identity. Models trained for face recognition encode unique facial features. These features are compared with stored face features in a database using similarity or distance thresholds. A match identifies the face, while a non-match labels it as "unknown."

As is shown in Fig. 1, our system is modeled similarly: it starts with a human presence sensor detecting motion using signals like infrared, ultrasonic, or microwave emissions. Upon detection, a camera captures images, triggered by a positive signal, conserving energy by operating reactively instead of continuously. Captured images are stored sequentially in a queue $\mathcal{Q}$ with capacity $L^\star$ based on generation time. A processing unit (PU), like a Raspberry Pi, retrieves and processes images from $\mathcal{Q}$ based on suitable resources signaled by x. PU's maximum resource use is denoted by $\mu^\star$. Power is supplied by a rechargeable battery, part of an integrated photovoltaic system [26] with a capacity of $B^\star$, ensuring continuous operation with renewable solar energy conversion [27]. Since human presence varies over time, we adopt a dynamic approach inspired by studies like [28], dividing time into discrete slots with duration Δ matching picture arrivals.

To develop a comprehensive model, we detail each component's behavior. At each time slot t, the decision-maker sets enable signal x^t, determining PU activation.

Case I. When the decision-maker generates $x^t = 0$, the PU remains idle during this t-th time slot. Suppose C^t represents the quantity of newly captured images during this time slot, L^t denotes the number of images stored in queue $\mathcal{Q}$. These images are stored in the queue $\mathcal{Q}$, necessitating an update to the value of L^t using function

$$L^{t+1} = \min\left(L^t + C^t, L^\star\right) \tag{1}$$

If P_{idle} symbolizes the PU's operational power when the system is idle, the energy consumption is estimated with

$$E_{u,idle}^t = P_{idle}\Delta \tag{2}$$

while the battery energy income is represented by

$$E_c^t = P_I^t\Delta \tag{3}$$

where P_I^t is indicative of the supplied solar charging power.

Consequently, the battery capacity is updated with

$$B^{t+1} = \min\left(B^t + E_c^t - E_{u,idle}^t, B^*\right) \tag{4}$$

with B^t represents the remaining battery capacity.

Case II. Otherwise, when the decision-maker generates $x^t = 1$, the PU will be waken and start to process the stored images in this t-th time slot. Since the task is detecting and recognizing the faces in captured images, the workflow will be arranged based on the property of this task.

1) Face Detection Phase: In the initial phase, the PU scans the first p^t images in the queue, detecting and recording the faces count in each of these p^t images. Although we assert the scanning of p^t images, the actual number of scanned images is given by

$$\hat{p}^t = \min\left(L^t - Q^{t-1}, p^t\right) \tag{5}$$

where Q^{t-1} denotes the count of images scanned up to the previous time slot. Assuming that the time taken to detect and count the faces in an image utilizing the full resources of the PU is T_S, we estimate the total time that the PU spends detecting and counting the images during time slot t:

$$\Delta_S^t = \rho \frac{\mu^\star}{(\mu^t)^\beta} \cdot T_S \cdot \hat{p}^t \tag{6}$$

with constraint

$$0 \le \mu^t \le \mu^\star \tag{7}$$

where the parameter ρ and β are introduced to describe the resource-aware inverse relationship (in most cases, researchers would like to set ρ and β with 1), and the value of Q should be updated with

$$Q^t = \min\left(Q^{t-1} + p^t, L^t\right) \tag{8}$$

given that a maximum of L^t images can be scanned.

2) Face Recognition Phase: Following the detection and recording of faces in $\hat{p}^t$ images, the PU transitions to a new phase where it is tasked with loading m^t images. Similar to the previous phase, even though the decision-maker stipulates that the PU should process m^t images, we have

$$\hat{m}^t = \min\left(m^t, Q^t\right) = \frac{m^t + Q^t - |m^t - Q^t|}{2} \tag{9}$$

that determines the actual number of images loaded and processed. Assuming that the time required to identify the owner of a face from an image, when utilizing the full resources of the PU, is T_R, and we can use

$$\Delta_R^t = \sum_{i=1}^{\hat{m}^t} \rho \frac{\mu^\star}{(\mu^t)^\beta} T_R \cdot N_i = \rho \frac{\mu^\star}{(\mu^t)^\beta} T_R \sum_{i=1}^{\hat{m}^t} N_i \tag{10}$$

to calculate the time cost of this phase. Here, N_i represents the count of faces in the i-th image. Then the value of Q must be updated with

$$Q^t = Q^t - \hat{m}^t \tag{11}$$

Concurrently, the human presence sensor may be activated due to passersby, triggering the camera to start capturing images. Thus, the queue $\mathcal{Q}$ may need to be updated with

$$L^{t+1} = \min\left(L^t - \hat{m}^t + C^t, L^\star\right) \tag{12}$$

The min operation in this expression implies that images may be dropped when the queue reaches its maximum capacity $L^\star$, and it will be obvious that

$$0 \le L^t \le L^\star \tag{13}$$

Since the PU keeps working in this time slot if x^t is 1, then the energy consumption will increase according to the efforts it makes. Given that processing power is known to be proportionate to μ^t, we can express the run power as:

$$P_W^t = \eta \cdot \left(\mu^t\right)^\alpha \tag{14}$$

with $\alpha \ge 1$ (usually researchers would like to set α with 3). Let's designate the time required to deploy the processing module with the PU's full resources as T_P, When the resource allocated to PU is μ^t, the time cost can be approximated as:

$$\Delta_P^t = \rho \frac{\mu^\star}{(\mu^t)^\beta} T_P \tag{15}$$

Consequently, the actual energy consumption is:

$$\begin{aligned} E_u^t &= E_{u,idle}^t + E_{u,work}^t \\ &= P_{idle}\Delta + P_W^t\left(\Delta_S^t + \Delta_R^t + \Delta_P^t\right) \end{aligned} \tag{16}$$

under the constraint:

$$\Delta_S^t + \Delta_R^t + \Delta_P^t \le \Delta \tag{17}$$

When considering the charging power P_I^t, the battery capacity can be updated as follows:

$$B^{t+1} = \min\left(B^t + \left(E_c^t - E_u^t\right), B^\star\right) \tag{18}$$

The minimum function is used here to prevent overcharging of the battery. Notably, a battery's availability significantly diminishes when its remaining energy is low. Therefore, it's crucial to maintain the battery's residual energy above a safe level. To ensure this, we impose the following requirement:

$$B^{t+1} \ge \kappa B^\star \tag{19}$$

where κ is the threshold for safe running.

B. Long-Term Performance Optimization

As defined by the problem statement, the decision-maker's role is to ascertain the values of x^t, μ^t, p^t, and m^t. In other words, it determines the timing and method for processing the persistent images captured by the onboard camera. Given the system's need for stability in a complex environment, we initially focus on the average energy consumption as the objective, with the aim to minimize it:

$$\Psi(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{m}) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(E_u^t x^t + E_{u,idle}^t\left(1 - x^t\right)\right)$$

Here, $\boldsymbol{x}$, $\boldsymbol{\mu}$, $\boldsymbol{p}$, and $\boldsymbol{m}$ record all values of these decision variables from $t = 1$ to T. Considering the importance of prompt image processing, we manage this by applying a constraint to the capacity of Q over time slots:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\{L^{t+1}\} \le L_{opt} \tag{20}$$

With this groundwork, we can formulate the target problem:

$$\begin{aligned} P_1 :& \min_{\boldsymbol{x},\boldsymbol{p},\boldsymbol{m},\boldsymbol{\mu}} \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(x^t E_{u,work}^t + E_{u,idle}^t\right) \\ & s.t.\ p^t, m^t \in \mathbb{N}, x^t \in \{0, 1\} \\ & \quad (7),\ (13),\ (17),\ (19),\ (20) \\ & with\ L^1 = 0, Q^1 = 0, B^1 = B^\star \end{aligned}$$

It's evident that P_1's objective comes from $\Psi(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{m})$, and $\boldsymbol{\mu}$ is the sole continuous variable in this problem. The expected energy consumption is intrinsically linked to the value of x^t. When x^t is 0, the energy consumption is solely for system maintenance; and when x^t is 1, the energy consumption is externally dependent on the actions the system takes for image processing. Hence, here we mainly focus on the situation where x^t is determined and defer the question about "where x^t comes from". To scrutinize the target problem, we need to examine the structure of its objective and constraints. Given the presence of long-term average items, the Lyapunov-based optimization framework is particularly suitable. We introduce a virtual queue $Y(t)$ (with $Y(1) = 0$) to store the constraint violation of (20). The value of this queue can be iteratively updated as:

$$Y(t+1) = \max\{Y(t) + L^{t+1} - L_{opt}, 0\} \tag{21}$$

Then the *drift-plus-penalty minimization* [29] approach can be adopted to minimize the average energy consumption as well as keep the Q in an unstuffed state. The original problem now changes to a per-slot-optimization problem:

$$\begin{aligned} P_2: \min_{x^t,p^t,m^t,\mu^t} \quad & Y(t)L^{t+1} + V \cdot x^t E^t_{u,work} \\ s.t. \quad & p^t, m^t \in \mathbb{N}, x^t \in \{0,1\},\ (17),\ (19),\ (20) \end{aligned}$$

IV. Problem Analysis and Experiments

In this section, we analyze the structure of the formulated problem and devise a precise solution. Additionally, we seek to identify whether approximate yet less complex methodologies exist to provide acceptable solutions. The goal is to employ an optimization algorithm in scenarios where the time slot is short. In problem P_2, it is evident that the optimal objective $Y(t)L^{t+1} = Y(t) \cdot \min(L^t + C^t, L^\star)$ remains constant and is unaffected by the variables μ^t, p^t, and m^t when x^t is set to 0. Therefore, the key factor lies in determining the optimal value of the objective when x^t is set to 1. In this case, the new problem is formulated with:

$$\min_{p^t,m^t,\mu^t} \quad Y(t)L^{t+1} + V\frac{\eta\rho\mu^\star}{(\mu^t)^{(\beta-\alpha)}}\left(T_S \cdot p^t + T_R\sum_{i=1}^{m^t} N_i + T_P\right)$$

$$\rho\frac{\mu^\star}{(\mu^t)^\beta}\left(T_S \cdot p^t + T_R\sum_{i=1}^{m^t} N_i + T_P\right) \leq \Delta \tag{22}$$

$$B^t + \left(E^t_c - E^t_u\right) \geq \kappa B^\star \tag{23}$$

$$0 \leq \mu^t \leq \mu^\star \tag{24}$$

$$0 \leq L^t \leq L^\star \tag{25}$$

$$0 \leq p^t \leq L^t - Q^t \tag{26}$$

$$0 \leq m^t \leq Q^{t-1} + p^t \tag{27}$$

$$p^t, m^t \in \mathbb{N} \tag{28}$$

It is worth noting here we use $\min(L^t - \hat{m}^t + C^t, L^\star)$ to represent L^{t+1}, which is different from that in $x^t = 0$.

In this problem, it is apparent that the variable m^t has an impact on the structure of the objective and constraints, while p^t and μ^t do not. Therefore, by considering the constraint (27), we can partition the problem into several sub-problems. In every sub-problem, the value of m^t is fixed with one of the elements in $\{0, 1, \ldots, Z_m\}$, where Z_m is the maximum value that m^t can be. It is evident that the value of Z_m can be set as Q^t. However, we can further enhance the runtime efficiency by examining the constraints (22), (23), (24), (26), and (27). By validating the following constraint, we can obtain the supremum of m^t:

$$\begin{aligned} & T_s \cdot \max\left(0, m^t - Q^{t-1}\right) + T_R\sum_{i=1}^{m^t} N_i + T_p \\ & \leq \left(\frac{B^t + P^t_I\Delta - \kappa B^\star - P_{idle}\Delta}{\eta \cdot \rho^{\frac{\alpha}{\beta}} \cdot (\mu^\star)^{\frac{\alpha}{\beta}} \cdot \Delta^{1-\frac{\alpha}{\beta}}}\right)^{\frac{\beta}{\alpha}} \end{aligned} \tag{29}$$

By solving these individual sub-problems, we can determine the optimal value by comparing the new ones obtained from the sub-problems where m^t is not a variable.

Now the present problem can be solved by solving sub-problems with different m^t values in parallel. With the constraint (22), we have:

$$\mu^t \geq \sqrt[\beta]{\frac{\rho \cdot \mu^\star \cdot \left(T_S \cdot p^t + T_R\sum_{i=1}^{m^t} N_i + T_P\right)}{\Delta}} \tag{30}$$

Hence, we can achieve the optimal value when

$$\begin{aligned} p^t &= \max\left(0, m^t - Q^{t-1}\right) \\ \mu^t &= \sqrt[\beta]{\frac{\rho \cdot \mu^\star \cdot \left(T_S \cdot p^t + T_R\sum_{i=1}^{m^t} N_i + T_P\right)}{\Delta}} \end{aligned} \tag{31}$$

With comparisons of those objectives with the above p^t, μ^t under enumerated x^t and m^t, we can obtain a **s**ustainable **un**manned **fa**cial **de**tection and **r**ecognition (SUNFADER) approach in the edge environment.

1) Dataset. We use real-world data from the Sensor Weather Traces dataset available in the UMASS Trace Repository.[1] This dataset records solar intensity (in watts/m^2) and various meteorological indicators in Amherst, Massachusetts, as shown in Fig. 2. The meteorological data is collected every 5 minutes, providing detailed observations of weather variables. Our focus is primarily on analyzing solar intensity data. The figure displays seasonal variations in solar intensity, which are also evident in daily observations. We use this data to estimate the energy absorbed by solar panels, namely charging power. Then, we apply our proposed method to determine if it's viable to arrange processing units for task processing. We use data from April 1, 2012, for constructing the charging curve and conducting experiments, with each experiment lasting a day. Although using data from a longer period could offer more comprehensive results, it would significantly increase the time required for this study.

2) Testbed. In this section, we set up a basic experimental platform, depicted in Fig. 3, to evaluate the performance of our proposed method using a `Raspberry Pi` 4B. Powered by the Broadcom `BCM2711` with an ARM Cortex-A72 processor running at 1.8GHz, it consumes between 2.7W and 6.4W. By executing the command `cpufreq-set -f x`, where `x` represents the desired frequency (e.g., 600000), we can adjust the processor frequency to balance computing

[1] https://traces.cs.umass.edu/index.php/Sensors/Sensors

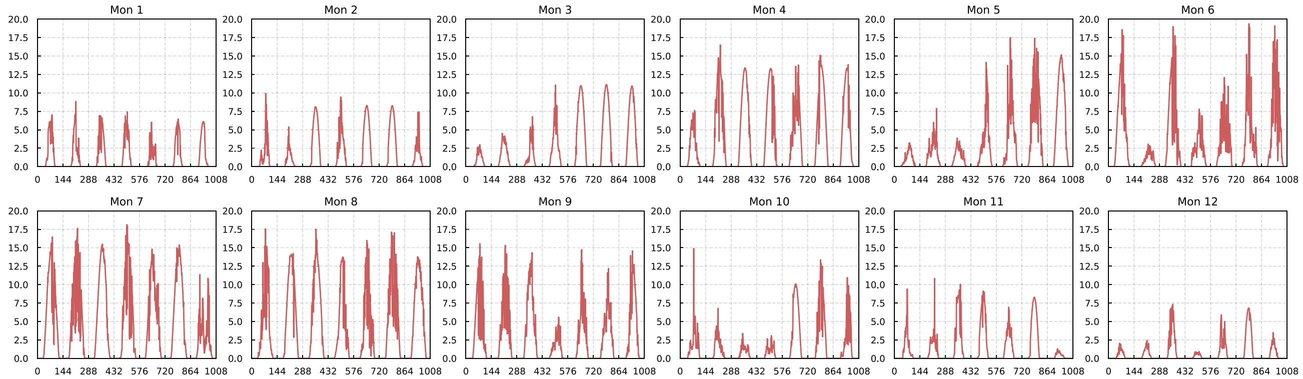
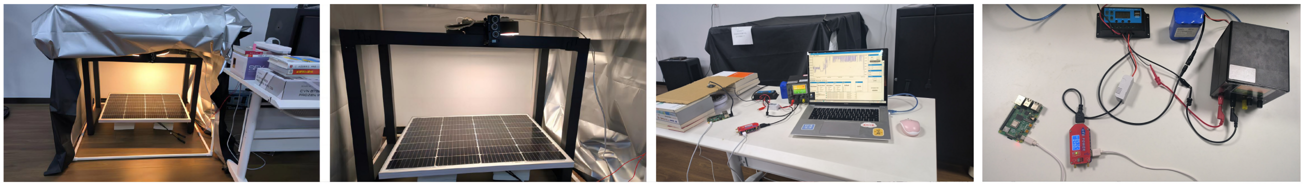


Fig. 2. Solar radiation intensity record for 2012.



(a) Set high solar intensity (b) Set low solar intensity (c) Monitor end-device running state (d) Experimental devices

Fig. 3. The testbed used in our experiment.

speed and power consumption. We utilize a Wuling LF4200-LS deuterium tungsten halogen light to mimic sunlight. Additionally, our setup includes a rechargeable lithium battery (working voltage=12V, battery capacity=5000mAH), a 100W solar panel with a 182mm monocrystalline silicon cell, and a solar charge controller. We assume a uniform distribution for crowd arrivals, with the average arrival rate denoted as λ. Furthermore, the number of faces in each photograph follows a uniform distribution, with an average value of γ.

3) Baselines. As our algorithm is the first attempt to solve the energy-efficient face detection and recognition problem in an edge computing environment, the existing algorithm is not specifically designed for this application and is not directly applicable. Hence, we have selected a few classic yet representative methods as benchmarks to compare with the proposed methods in this article:[2]

- STRICT. This method introduces the ideas like [30] to adopt a more radical strategy in which the long-term average for a given problem is substituted with a per-slot average. According to this approach, the queue length should never exceed L_{opt} in any given time slot. Concurrently, with this constraint in place, the energy consumption of the same time slot should be optimized.

- ACTIVE. This method necessitates the processing unit to handle all images in each time slot immediately [31] after collection, implying a continual consumption of the queue Q when it is not empty. This approach provides a mechanism for instantaneous processing, similar to those typical face detection and recognition applications.

- LAZY. This method implies a more passive role [32] for the PU, as it only initiates the processing of images once the queue Q is full.

A. Performance Comparison

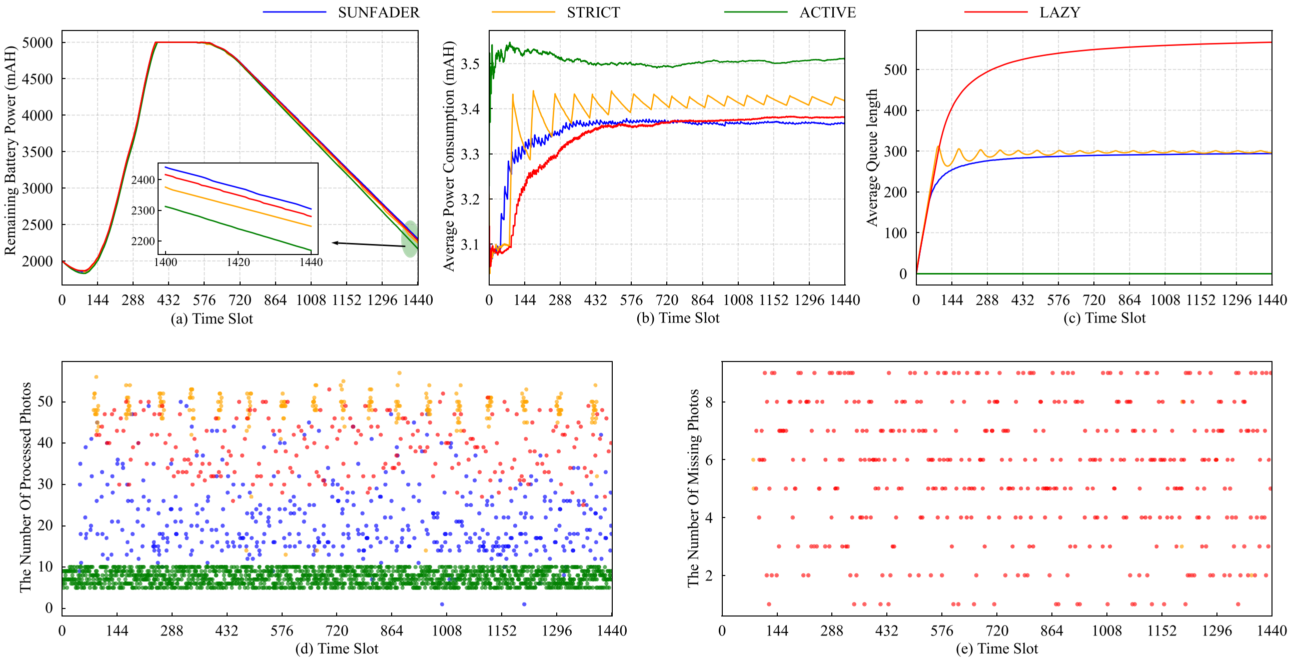
To analyze the performance of our SUNFADER method and the baselines, we conducted experiments on our simple experimental platform using the following settings: Δ=1min, T=1440, $L^{\star}$=600, L_{opt}=300, $\eta = 1.51 \times 10^{-16}$, α=1.36, $\rho = 0.94$, $\beta = 1$, $\mu^{\star}$=1.8GHz, $B^{\star}$=5000mAH, κ=0.3, V=1000, P_{idle}=0.54W, T_p=0.47s, T_s=0.19s, T_r=0.45s. The values of α, β, η and ρ were determined by fitting the performance curve of the device, whereas T_p, T_s, and T_r were obtained by measuring the mean actual time costs.

Let's dive into the comparative analysis of various algorithms in real-world settings. Fig. 4(a) clearly shows that the ACTIVE algorithm has the highest energy consumption among the four, mainly due to its immediate initiation of facial recognition and identity verification processes without queuing photos. This frequent activation pattern leads to a steady rise in energy usage, as depicted in Fig. 4(b), where the ACTIVE method consistently consumes most energy.

Analyzing Fig. 4(a) alongside Fig. 4(b), we notice the LAZY method starts with lower energy consumption and maintains higher remaining battery levels initially. This is because it delays facial recognition until the photo queue nears capacity at the start. However, this delay causes significant data loss, as seen in Fig. 4(e). As the queue fills up later, the system faces increased processing pressure, leading to higher energy consumption over time, as shown in Fig. 4(c).

The STRICT method exhibits regular energy consumption and queue length fluctuations, visible in Fig. 4(b) and

[2] The code is released at github.com/xuexizi/SunFader.



(a) Time Slot (b) Time Slot (c) Time Slot

(d) Time Slot (e) Time Slot

Fig. 4. Experiment results on the prototype.

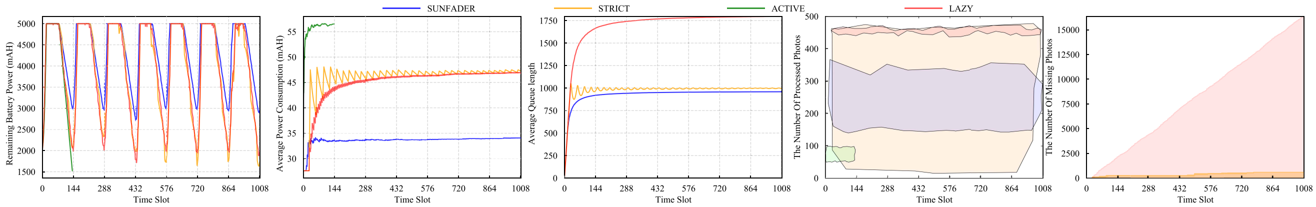


Fig. 5. Simulating experiment results of the four algorithms.

Fig. 4(c). Its rigid queue length limit and lack of preemptive photo processing result in concentrated processing bursts, maintaining a stable but high energy usage pattern, as shown in Fig. 4(d).

In contrast, our proposed method maintains higher battery levels for longer periods and gently optimizes photo queue length within set constraints. Unlike other methods with fixed processing numbers, our method SUNFADER dynamically adjusts based on available resources, as highlighted in Fig. 4(d), providing flexibility and efficiency in decision-making.

B. Single-Factor Investigation on System Settings

Furthermore, to ensure the long-term stability of our system, we selected a time interval of $\Delta = 10$ minutes for the simulation experiment mentioned in the paper. This resulted in a total of 1008 time slots, corresponding to a week, where $T = 6 \times 24 \times 7$. The settings of the single-factor experiments are shown in Table II.

The results are illustrated in Fig. 5. In Fig. 5(1), the ACTIVE algorithm fails after approximately 144 time slots or one day. The remaining battery levels for the STRICT and LAZY methods are initially similar but vary due to solar radiation changes and decision strategies, as seen in Fig. 5(2). The STRICT method's energy consumption stabilizes and converges with the LAZY method after fluctuations. Our method SUNFADER maintains a consistently higher battery level (46.18% higher than STRICT and 119.86% higher than LAZY), with the lowest overall energy consumption and minimal fluctuations, as shown in Fig. 5(2). Fig. 5(c) depicts the average queue length. Only our method and the ACTIVE method adhere to the queue length constraint. In Fig. 5(4), our method SUNFADER processes data steadily, adapting based on available resources, unlike the STRICT method with significant fluctuations and the ACTIVE method processing less data each time. The LAZY method processes the most data but has the lowest processing frequency. Data loss occurs due to algorithms' inability to process data timely, as seen in Fig. 5(5). Our method and the ACTIVE method avoid data loss, while the LAZY and STRICT suffer data loss throughout.

Fig. 6 shows the influence of parameter V on our method's performance. As V increases from 10 to 800, energy consumption decreases, and queue length increases significantly. Further increases in V show negligible impact, indicating optimal V selection for energy control and buffer size management.

1) Average Arrival Rate: Upon comparing Fig. 5 and Fig. 7, it is evident that when the value of λ is small (indicating a low arrival rate and a relatively idle system), the ACTIVE method can also maintain long-term stable operation. From

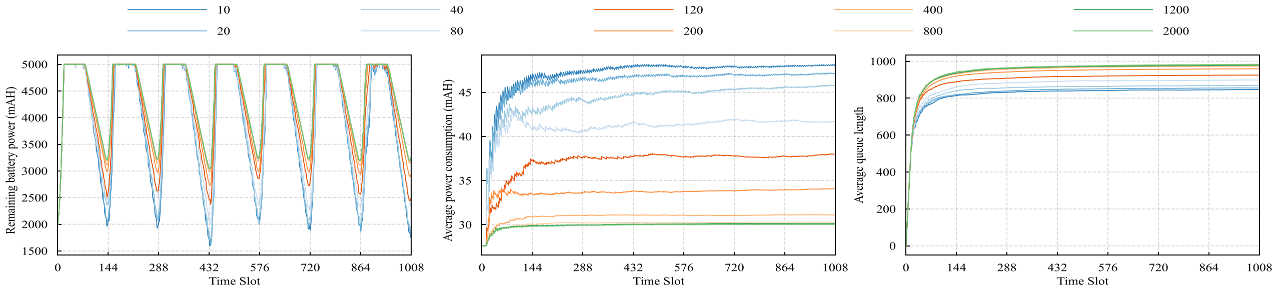


Fig. 6. The impact of the Lyapunov control parameter V.

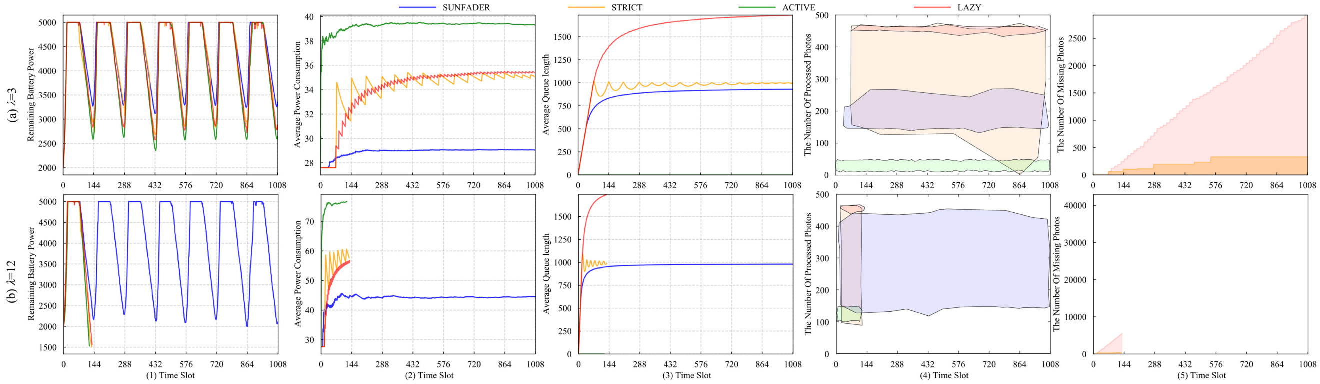


Fig. 7. The average photo arrival rate $\lambda = 3$ and 12.

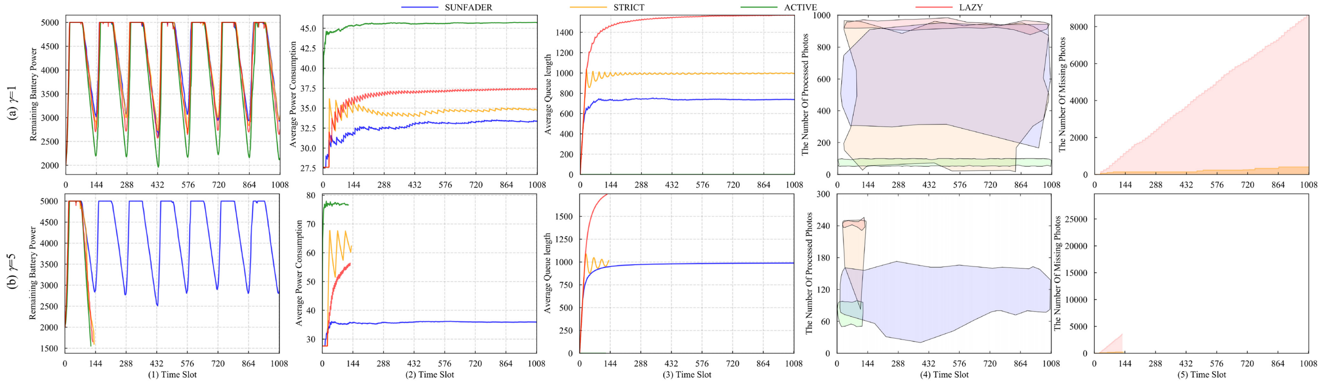


Fig. 8. The average photo arrival rate $\gamma = 1$ and 5.

Fig. 7(a), we observe that after running for a week, the STRICT method reaches an equilibrium state in terms of average system energy consumption. Furthermore, the converged average energy consumption of the STRICT method remains similar to that of the LAZY method, but is 21.65% higher than our proposed method. Additionally, after 300 time slots, the ACTIVE method's average system energy consumption reaches a nearly equilibrium state, but its final converged energy consumption is approximately 34.71% higher than our proposed method.

When the value of λ increases, indicating a higher arrival rate and a busier system, we can observe from Fig. 7(b) that, except for our proposed method, the other methods fail to maintain long-term stable operation. After running for almost a day, the STRICT, ACTIVE, and LAZY enter a sleep state due to the battery falling below the safe threshold.

2) Average Face Number: The parameter γ reflects the degree of crowd aggregation. When the crowd is relatively sparse, or when the number of faces in each photo decreases, we can observe from Fig. 8(a) that all four algorithms can operate for an extended period. However, from Fig. 8(a)(2), it can be seen that the STRICT method exhibits less fluctuation in average energy consumption, and the converged average energy consumption is 7.45% lower than the LAZY method. In comparison, our proposed method converges at a slower rate, taking approximately 500 time slots to approach convergence. By comparing Fig. 5(3) and Fig. 8(a)(3), we also observe a decrease in average queue length for both the LAZY method and our proposed method when the crowd is sparse, while the STRICT method shows little change. In Fig. 8(a)(4), due to the sparse crowd, the processing frequency of all methods significantly decreases, but the number of photos

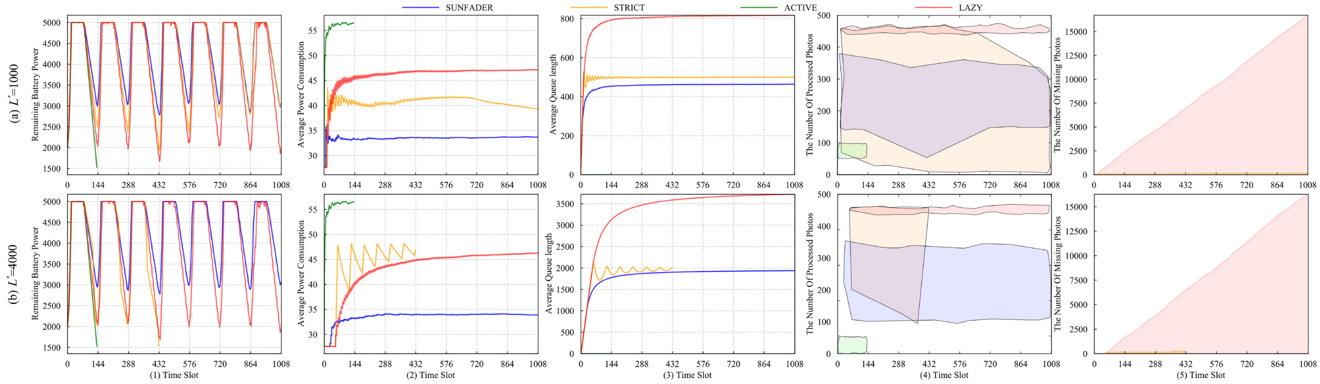


Fig. 9. Acceptable Maximum Queue Length $L^{\star} = 1000$ and 4000.

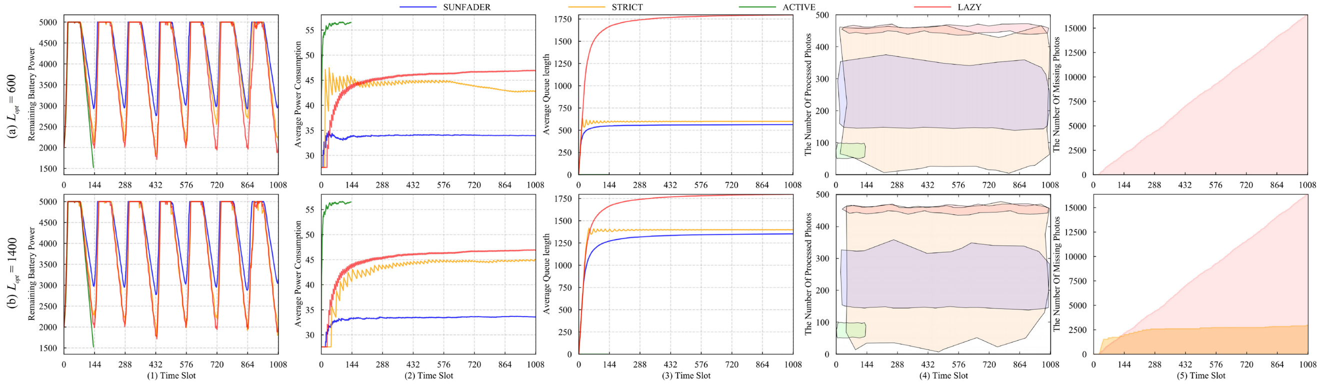


Fig. 10. Optimal Queue Length $L_{opt} = 600$ and 1400.

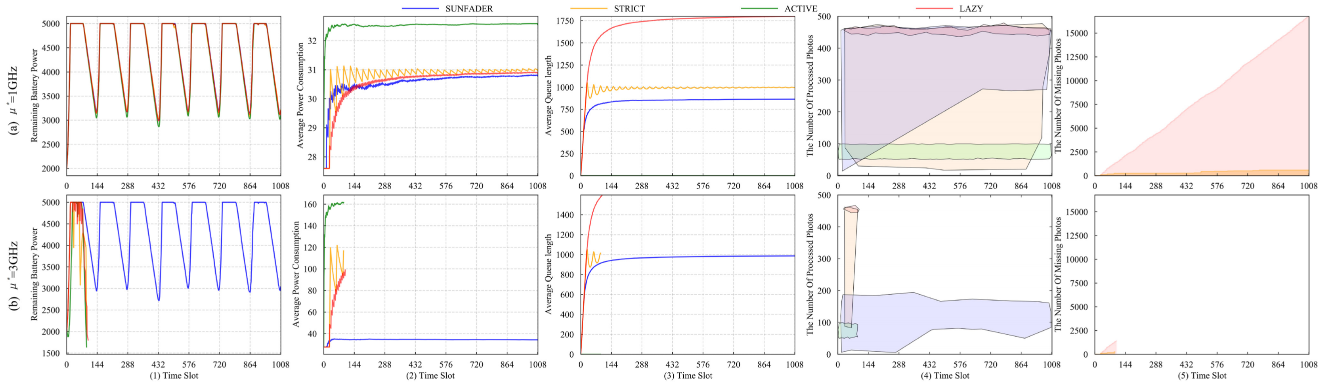


Fig. 11. Maximum Resource of PU $\mu^{\star} = 1GHz$ and $3GHz$.

processed per instance increases significantly. At this point, the processing frequency and the number of processed photos are quite similar for the STRICT method and our method SUNFADER.

On the other hand, when the crowd aggregation level is high, or when the average number of faces in the photos captured by the system increases, only our proposed method can operate continuously. Additionally, by examining Fig. 7(b) and Fig. 8(b), we can observe that compared to the increased number of faces in the photos due to dense crowds, increasing the photo capture frequency leads to a higher number of photos processed for our method SUNFADER, resulting in a greater increase in average energy consumption.

3) Acceptable Maximum Queue Length: The maximum length of the queue, denoted as $L^{\star}$, is a measure that reflects the system's tolerance for worst-case scenarios during task processing. It has an impact on the rate of data loss. In Fig. 9(a) and Fig. 9 (b), we compare the effects of reducing and increasing $L^{\star}$, respectively. When $L^{\star}$ is set to 1000, we can observe in Fig. 9(a)(1) that the remaining battery level of the STRICT method suddenly increases in the later stages of system operation. Additionally, from Fig. 9(a)(2), we can see a decreasing trend in average energy consumption of the STRICT method on the last day. When $L^{\star}$ is set to 4000, in Fig. 9(b), the STRICT method consumes more energy during the queue length constraint process due to the increased length

Algorithm 1: SUNFADER

```
Input: the solar charging power supply $P_I^t$, the number of incoming images $C^t$
Output: enable signal $\boldsymbol{x}$, quantity of images scanned $\boldsymbol{p}$, quantity of images recognized $\boldsymbol{m}$, allocated resources $\boldsymbol{\mu}$
initialize battery capacity $B^1 \leftarrow B^\star$, image queue length $L^1 \leftarrow 0$, quantity of images scanned $Q^1 \leftarrow 0$, virtual queue $Y(1) \leftarrow 0$
for $t = 1$ to $T$ do
    $P_2 = \text{MAX}$, $x^t = 1$, $m^t = 0$, $p^t = 0$, $\mu^t = 0$
    for $\tilde{m^t} = 1$ to $Q^t$ do
        if $\tilde{m^t}$ conforms to constraint (29) then
            $\tilde{p^t} = \max(0, \tilde{m^t} - Q^{t-1})$
            Solve $\tilde{\mu^t}$ with Equation (31)
            $\tilde{P_2} \leftarrow (\tilde{m^t}, \tilde{p^t}, \tilde{\mu^t})$
            if $\tilde{P_2} < P_2$ then
                $P_2 = \tilde{P_2}$, $m^t = \tilde{m^t}$, $p^t = \tilde{p^t}$, $\mu^t = \tilde{\mu^t}$
            end
        end
    end
    $\tilde{P_2} = Y(t) \cdot \min(L^t + C^t, L^\star)$
    if $\tilde{P_2} < P_2$ then
        $x^t = 0$, $m^t = 0$, $p^t = 0$, $\mu^t = 0$
    end
    Place new photos into the queue $\mathcal{Q}$
    if $x^t = 0$ then
        Using Equation (1), (4), (21) update $(L^{t+1}, B^{t+1}, Y(t+1))$
    else
        Using Equation (8), (11), (12), (18), (21) update $(Q^t, L^{t+1}, B^{t+1}, Y(t+1))$
    end
end
return the decision $\Psi(\boldsymbol{x}, \boldsymbol{\mu}, \boldsymbol{p}, \boldsymbol{m})$
```

TABLE II
EXPERIMENTAL PARAMETERS

λ	8/min	γ	3	$\mu^\star$	1.8GHz
$L^\star$	2000	L_{opt}	1000	T_s	0.19s
V	200	T	1008	T_r	0.45s
η	3.8×10^{-29}	Δ	10min	T_p	0.47s
κ	0.3	$B^\star$	5000mAH	P_{idle}	0.54W
α	3	β	1	ρ	1

of the photo storage queue. As a result, it enters a sleep state at 430-th time slot due to the battery level falling below the threshold. In contrast, our method maintains relatively stable when $L^\star$ changes.

4) Optimal Queue Length: The optimal length of the queue, L_{opt}, differs from $L^\star$ in that it aims to maintain a relatively optimal state of data processing timeliness. By comparing Fig. 5, Fig. 10(a), and Fig. 10(b), we can observe that changing L_{opt} does not affect the long-term operation of the algorithm. Instead, it only affects the average power consumption of the algorithm. Therefore, whether decreasing or increasing L_{opt}, it only has an impact on our method SUNFADER and the STRICT method. In our method, the average power consumption remains almost unchanged, whereas the power consumption of the STRICT method varies with the decrease or increase of L_{opt}.

When L_{opt} decreases, the average power consumption of the STRICT method quickly converges and becomes lower than that of the LAZY method after running for one day. When L_{opt} increases, the average power consumption of the STRICT method is consistently lower than that of the LAZY method, but it also leads to a significant increase in the number of lost frames in the STRICT method. Furthermore, decreasing or increasing L_{opt} will reduce the fluctuation amplitude of the average queue length in the STRICT method.

5) Maximum Resource of PU: The parameter $\mu^\star$ represents the upper limit of system performance. By setting the maximum operating frequency of the system to 1GHz and 3GHz respectively, we can clearly observe the performance of various algorithms in Fig. 11(a) and Fig. 11(b). It is evident that after reducing the maximum operating frequency, all four algorithms can maintain long-term stable operation. Additionally, the remaining energy in each time slot and the average power consumption of the system tend to converge for all four algorithms. This is due to the smaller difference between the maximum and minimum operating frequencies, resulting in less distinction in the decision-making effectiveness. In this case, the ACTIVE method proves to be the optimal solution.

When the maximum operating frequency of the system increases, it is clearly observed from Fig. 11(b) that the effectiveness of different decision-making approaches significantly improves. Except for our method SUNFADER, which can continue running, the other methods enter sleep mode after approximately 100 time slots.

6) Battery Capacity and Safe Level: By observing Fig. 12(a) and Fig. 12(b), we can draw the following conclusions: selecting an appropriate battery capacity is crucial for the stable operation of the system. When the battery capacity is low, except for our proposed algorithm, the other algorithms fail to ensure stable system operation. However, when a larger battery capacity is chosen, the system using the ACTIVE method can achieve long-term operation as well. It is important to note that a larger battery capacity also increases the weight load on the system, which needs to be carefully considered.

Furthermore, another parameter that we need to consider is the safety energy level of the battery. On one hand, a lower safety energy level may result in frequent battery depletion, reducing the battery's lifespan. On the other hand, a higher safety energy level may cause the system to enter sleep mode prematurely, limiting the system's workload. Therefore, selecting an appropriate safety energy level is crucial. By comparing Fig. 13(a) and Fig. 13(b), we can observe that when κ is set to 0.2, the system enters sleep mode later, allowing the ACTIVE method to continue running. On the other hand, when κ increases to 0.4, the system enters sleep mode earlier. The LAZY method enters sleep mode after running

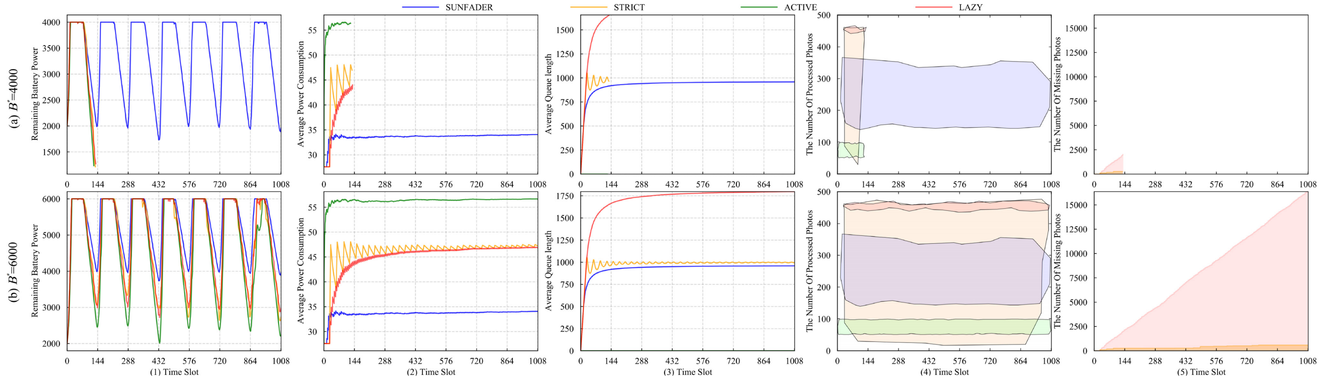


Fig. 12. Battery Capacity $B^\star = 4000mA$ and $6000mA$.

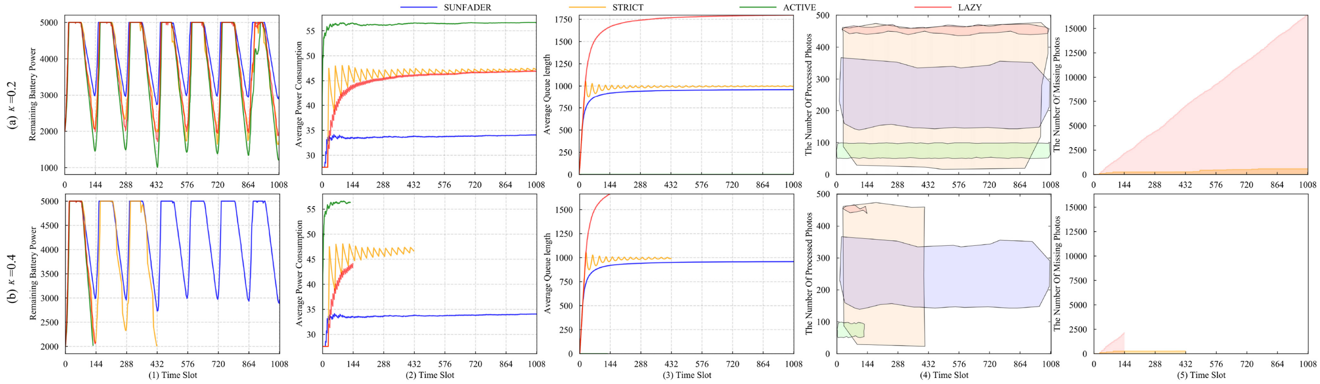


Fig. 13. Battery Safe Level $\kappa = 0.2$ and 0.4.

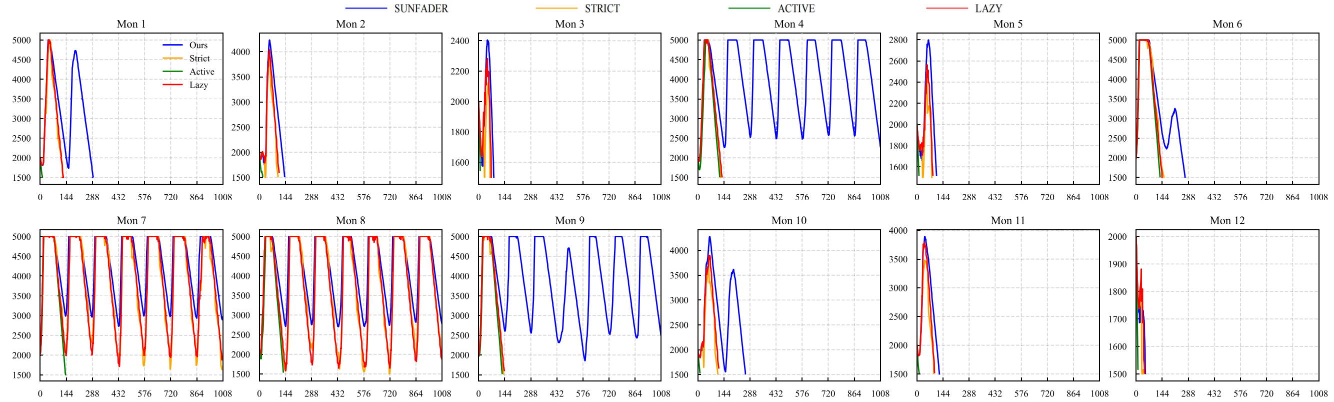


Fig. 14. The change in remaining battery power over time in different months.

for one day, and the STRICT method enters sleep mode after running for three days.

7) Average Charging Power: In Fig. 14, we observe that the operational stability of the monitoring system varies across different months due to the seasonal fluctuations in solar radiation. In months with lower solar radiation intensity, such as January to March and October to December, the system's battery easily drops below the safety threshold. However, our proposed method can still maintain longer operation compared to other methods. In April and September, when solar radiation intensity is enhanced, our method SUNFADER can sustain long-term operation. However, during the unpredictable weather conditions in May and June, none of the four algorithms can maintain prolonged operation. In the peak months of July and August, when solar intensity is at its maximum, our method, along with STRICT and LAZY, can sustain stable operation for a week. On the other hand, ACTIVE algorithm, due to excessive power consumption and failure to consider external environmental factors, can only maintain battery power for a day before dropping below the threshold.

As depicted in Fig. 15, the solar intensity is categorized into four distinct levels, within which we conduct a comparative analysis of the number of photos processed per unit of energy across several algorithmic approaches. Evidently, irrespective of the solar intensity level, our method SUNFADER demonstrates the highest energy efficiency, processing the greatest number of photos per unit of energy consumed, significantly outperforming the other three algorithms. ACTIVE algorithm,

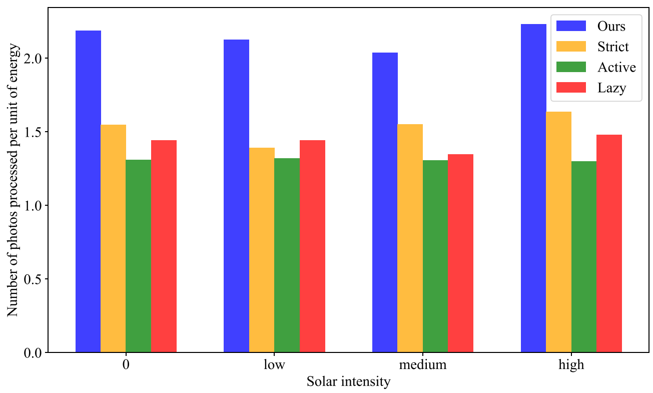


Fig. 15. Energy efficiency at different light intensities.

on the other hand, exhibits the lowest energy efficiency due to the energy expenditure associated with each service activation. In addition, Lazy algorithm, owing to its propensity for image loss, processes a relatively limited number of photos, thereby compromising its overall energy efficiency.

V. Learning-Based Approximate Approach

In the approached proposed above, we can obviously find that our proposed algorithm still takes some time to generate the final strategies. This is primarily due to the fact that our algorithm employs a traversal-based strategy to solve the problem, and not all of these subproblems can be solved using parallel computing techniques on resource-constrained devices. Therefore, optimizing the runtime further in practical scenarios would significantly enhance the generality of our approach.

Looking back at the aforementioned modeling, the solution for μ^t, m^t, and p^t can be generalized as "how to generate appropriate actions based on the observation of the current state at time slot t." In other words, if we define the current state as $\mathcal{S}^t = \{B^\star, \mu^\star, T, Q^\star, L_{opt}\}$, the "solution generation" can be represented as $(\mu^t, m^t, p^t, x^t) = f(\mathcal{S}^t)$. In our previous description, f corresponds to the SunFader method we proposed. If f is a well-trained regression model and its output corresponds to the desired (μ^t, m^t, p^t, x^t), we can offload the computationally expensive part to the training phase, thereby significantly reducing the generation time, especially in scenarios with multiple edge devices.

XGBoost, which stands for eXtreme Gradient Boosting, is an advanced machine learning algorithm known for its efficiency and effectiveness in handling a wide range of predictive modeling tasks [33] with tabular data like this. It is based on the gradient boosting framework, where an ensemble of decision trees is built sequentially to minimize the loss function, with regularization techniques like L1 and L2 to prevent overfitting. XGBoost is designed to be highly scalable, allowing for parallel processing and distributed computing, and it can handle both numerical and categorical data with ease. It also supports custom loss functions and evaluation metrics, making it adaptable for various problems. With built-in cross-validation and the ability to handle missing values, XGBoost is user-friendly and provides valuable insights into feature importance. Its strengths in speed, accuracy, and scalability have made it a popular choice for resource-constrained edge computing environments. The fitting results are illustrated in Table III. Compared with models such as RandomForest, ExtraTree, LightGBM, CatBoost and MLP, XGBoost shows the a good time cost in inference and the best quality in the results.

TABLE III
Performance of Different Learning-Base Models

Model	MAE	Inference Time (s)
XGBoost	-0.002448	0.045435
LightGBM	-0.002488	0.309884
CatBoost	-0.002563	0.028481
RandomForestMSE	-0.002635	0.239409
ExtraTreesMSE	-0.002739	0.264025
LightGBMXT	-0.005357	0.845604
MLP	-0.007521	0.118541

Fig. 16 illustrates the comparison between our trained model and our proposed method on a set of new parameters. Although the number of processed images is not exactly the same, the overall power consumption, energy consumption, and length of the image storage queue remain relatively consistent. Additionally, we compared the execution time of our proposed algorithm with XGBoost, as shown in Fig. 17. We observed that when the average arrival rate of images, λ, is low, our proposed method exhibits lower execution time. However, as λ gradually increases, the decision time of our proposed method experiences exponential growth. Similarly, the capacity of image queue $L^\star$ has a similar effect on decision time. However, no matter how the parameters change, XGBoost consistently maintains an inference time of approximately 0.04s. Therefore, in practical applications, we can choose the appropriate method for deployment based on the specific environment.

VI. System Expansion Discussion

In the previous section, we conducted experimental analyses on the time required for system decision-making, and found that as the size of the population increases, the system decision-making time exhibits an exponential growth trend. Therefore, we proposed using XGBoost to assist in decision-making, and the experiments proved that XGBoost can indeed optimize the algorithm time. However, we only considered optimization in the scenario with a single device. When we are in a more complex environment, we need to increase the number of deployed devices to expand the original system in order to achieve our goal. Therefore, we need to rethink the problem of energy consumption optimization under the deployment of multiple devices.

In the multi-device scenario, assuming the number of devices is N, the series of variables related to system decision-making in the original single-device scenario need to be represented using vectors. Specifically, we use $\boldsymbol{x^t} = (x_1^t, x_2^t, \ldots, x_N^t)$ to represent the service activation status of each device, $\boldsymbol{\mu^t} = (\mu_1^t, \mu_2^t, \ldots, \mu_N^t)$ to represent the PU resources allocated to each device, $\boldsymbol{p^t} = (p_1^t, p_2^t, \ldots, p_N^t)$ to represent the number of photos to be scanned by each device

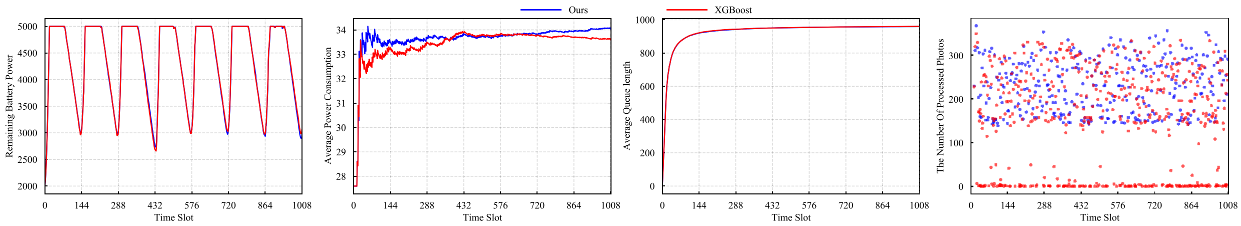
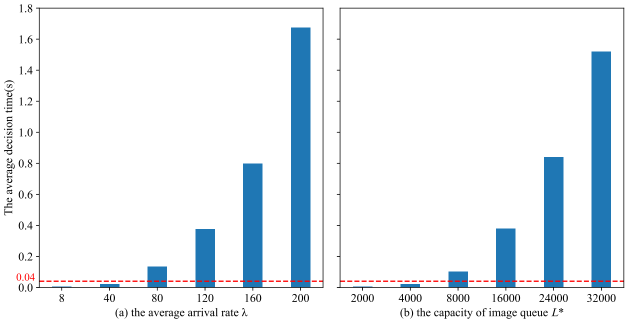


Fig. 16. The comparison in performance of different approaches.



(a) the average arrival rate λ

(b) the capacity of image queue L^*

Fig. 17. The time costs of XGBoost and SunFader.

in time slice t, and $\boldsymbol{m^t} = (m_1^t, m_2^t, \ldots, m_N^t)$ to represent the number of photos to be recognized by each device in time slice t. Then, we can use $\boldsymbol{\mathcal{S}^t = \{B^\star, \mu^\star, T, Q^\star, L_{opt}\}}$ to represent the current state, the "generated solution" can be expressed as $\boldsymbol{(\mu^t, m^t, p^t, x^t) = F(\mathcal{S}^t)}$.

However, through careful analysis of the scenario, we can find that since the devices are not associated with each other, we can decompose the original complex optimization problem into multiple independent sub-problems, and then solve them using the optimization method proposed earlier. Furthermore, by decomposing the multi-device deployment problem in a complex environment into multiple sub-systems for solving, not only can we solve deployment problems of different scales and environments, but we can also configure different time slice management for different devices based on their specific environments, further improving the flexibility of the system.

Therefore, we need to choose the corresponding system deployment and optimization solution based on the complexity of the actual scenario. When the scenario is simple and the population is small, we can directly use the existing method. When the scenario becomes complex and requires the detection of data from multiple regions, we can choose to increase the number of devices and decompose the optimization problem into sub-problems for solving. Furthermore, if the scenario has a large population, and the system decision-making time is relatively long, we need to use tools similar to XGBoost to assist the system decision-making, thereby reducing the decision-making time and further optimizing the energy consumption.

VII. Conclusion and Future Work

In conclusion, this paper has focused on optimizing facial detection and recognition systems in complex environments, with a specific emphasis on improving availability and durability. Our contributions include the development of an unmanned facial detection and recognition system using energy harvesting, the design of an adaptive wake-and-sleep algorithm for maximizing battery life while maintaining real-time performance, and the demonstration of improved system availability without compromising real-time performance.

There are many possibilities to apply our work in real-time scenarios. It can be applied to smart city monitoring, deployed in public places such as squares and stations for real-time monitoring. On the other hand, the system can be integrated into smart home security systems to realize contactless automatic door opening and video recording. In addition, the system can also be applied to the automation of commercial and industrial scenarios such as shopping malls and factories, such as automated import/export management and employee attendance. However, there are some challenges in the implementation process. On the one hand, the system is environmentally adaptive and needs to be optimized and adjusted for each specific application scenario. On the other hand, the application of face recognition technology may cause privacy and security problems, which need to be solved by perfect data security mechanisms and user authorization procedures.

For future work, we plan to explore advanced techniques such as edge-based model compression and acceleration to further optimize our system. These techniques can help reduce the model size and inference latency, thereby improving the overall efficiency and deployment feasibility of our system. Additionally, we aim to investigate the integration of other renewable energy sources, such as solar or wind power, to make the system more self-sustainable and less reliant on grid-based electricity. Furthermore, we intend to utilize unmanned aerial vehicles (UAVs) to collect the processed data, enabling a totally network-independent operation of the system [34]. This approach can help overcome connectivity challenges and expand the system's deployment to remote or infrastructure-limited areas.

Acknowledgment

Z. Xiang and H. Gao designed the method and wrote the paper. X. Xue conducted the experiments and plotting. S. Dustdar and Z. Zheng joined the modeling. Y. Chen provided the hardware.

References

[1] H. Zhang, J. Luo, Y. Tu, R. Wang, D. Wu, and J. Yang, "Microservice deployment mechanism with diversified QoS requirements for smart health system in industry 5.0," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 869–880, Nov. 2023.

[2] S. Anbalagan, G. Raja, S. Gurumoorthy, R. D. Suresh, and K. Ayyakannu, "Blockchain assisted hybrid intrusion detection system in autonomous vehicles for industry 5.0," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 881–889, Nov. 2023.

[3] J.-H. Syu, J. C.-W. Lin, G. Srivastava, and K. Yu, "A comprehensive survey on artificial intelligence empowered edge computing on consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 1023–1034, Nov. 2023.

[4] F. Noorbehbahani, A. Mohammadi, and M. Aminazadeh, "A systematic review of research on cheating in online exams from 2010 to 2021," *Educ. Inf. Technol.*, vol. 27, no. 6, pp. 8413–8460, 2022.

[5] Z. Xiang, Y. Zheng, Z. Zheng, S. Deng, M. Guo, and S. Dustdar, "Cost-effective traffic scheduling and resource allocation for edge service provisioning," *IEEE/ACM Trans. Netw.*, vol. 31, no. 6, pp. 2934–2949, Dec. 2023.

[6] M. Navardi, E. Humes, and T. Mohsenin, "E2edgeai: Energy-efficient edge computing for deployment of vision-based DNNs on autonomous tiny drones," in *Proc. IEEE/ACM 7th Symp. Edge Comput. (SEC)*, 2022, pp. 504–509.

[7] S.-C. Liu, C. Gao, K. Kim, and T. Delbruck, "Energy-efficient activity-driven computing architectures for edge intelligence," in *Proc. Int. Electron Devices Meet. (IEDM)*, 2022, pp. 21.2.1–21.2.4.

[8] H. Jiang, F. Xiong, and Y. Huang, "Energy efficiency improvement scheme based on edge computing," in *Proc. IEEE Int. Conf. Artif. Intell. Comput. Appl. (ICAICA)*, 2021, pp. 1033–1036.

[9] X. Chen, J. Zhang, B. Lin, Z. Chen, K. Wolter, and G. Min, "Energy-efficient offloading for DNN-based smart IoT systems in cloud-edge environments," *IEEE Trans. Parallel Distr. Syst.*, vol. 33, no. 3, pp. 683–697, Mar. 2022.

[10] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Edge intelligence for energy-efficient computation offloading and resource allocation in 5G beyond," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12175–12186, Oct. 2020.

[11] S. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Energy efficiency and delay tradeoff for wireless powered mobile-edge computing systems with multi-access schemes," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1855–1867, Mar. 2020.

[12] D. S. Lakew, A.-T. Tran, N.-N. Dao, and S. Cho, "Intelligent self-optimization for task offloading in LEO-MEC-assisted energy-harvesting-UAV systems," *IEEE Trans. Netw. Sci. Eng.*, early access, Jan. 3, 2024, doi: 10.1109/TNSE.2023.3349321.

[13] N. Su et al., "Joint MU-MIMO precoding and computation optimization for energy efficient industrial IoT with mobile edge computing," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 3, pp. 1472–1485, Sep. 2023.

[14] J. Wen, J. Yang, T. Wang, Y. Li, and Z. Lv, "Energy-efficient task allocation for reliable parallel computation of cluster-based wireless sensor network in edge computing," *Digit. Commun. Netw.*, vol. 9, no. 2, pp. 473–482, 2023.

[15] H. Khazaei, N. Mahmoudi, C. Barna, and M. Litoiu, "Performance modeling of microservice platforms," *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2848–2862, Dec. 2020.

[16] N. Mahmoudi and H. Khazaei, "Performance modeling of metric-based serverless computing platforms," *IEEE Trans. Cloud Comput.*, vol. 11, no. 2, pp. 1899–1910, Jun. 2023.

[17] J. Wen, Y. Chen, K. Jin, and C. Liu, "Revolutionizing network performance: The active and passive service path performance monitoring analysis method," in *Proc. IEEE 10th Int. Conf. Cyber Security Cloud Comput.*, 2023, pp. 108–113.

[18] S. P. Singh et al., "A new QoS optimization in IoT-smart agriculture using rapid adaption based nature-inspired approach," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 5417–5426, Feb. 2024.

[19] M. Alizadeh and H. Tabassum, "Power control with QoS guarantees: A differentiable projection-based unsupervised learning framework," *IEEE Trans. Commun.*, vol. 71, no. 8, pp. 4605–4619, Aug. 2023.

[20] L. Zhang, J. Wang, W. Wang, Z. Jin, Y. Su, and H. Chen, "Smart contract vulnerability detection combined with multi-objective detection," *Comput. Netw.*, vol. 217, Nov. 2022, Art. no. 109289.

[21] Y. Yang et al., "Mixed game-based AoI optimization for combating COVID-19 with AI bots," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 11, pp. 3122–3138, Nov. 2022.

[22] L. An, Z. Yan, W. Wang, J. K. Liu, and K. Yu, "Enhancing visual coding through collaborative perception," *IEEE Trans. Cogn. Develop. Syst.*, vol. 15, no. 4, pp. 1744–1753, Dec. 2023.

[23] I. Adjabi, A. Ouahabi, A. Benzaoui, and A. Taleb-Ahmed, "Past, present, and future of face recognition: A review," *Electronics*, vol. 9, no. 8, p. 1188, 2020.

[24] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5203–5212.

[25] Y. Xu, W. Yan, G. Yang, J. Luo, T. Li, and J. He, "Centerface: Joint face detection and alignment using face as point," *Sci. Program.*, 2020, to be published.

[26] D. Das, B. Singh, and S. Mishra, "Grid interactive solar PV and battery operated air conditioning system: Energy management and power quality improvement," *IEEE Trans. Consum. Electron.*, vol. 69, no. 2, pp. 109–117, May 2023.

[27] R. Wang, D. Ma, M.-J. Li, Q. Sun, H. Zhang, and P. Wang, "Accurate current sharing and voltage regulation in hybrid wind/solar systems: An adaptive dynamic programming approach," *IEEE Trans. Consum. Electron.*, vol. 68, no. 3, pp. 261–272, Aug. 2022.

[28] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016.

[29] H. Hong, J. Jiao, T. Yang, Y. Wang, R. Lu, and Q. Zhang, "Age of incorrect information minimization for semantic-empowered noma system in S-IoT," *IEEE Trans. Wireless Commun.*, vol. 23, no. 6, pp. 6639–6652, Jun. 2024.

[30] Z. Yang, W. Xu, Y. Pan, C. Pan, and M. Chen, "Optimal fairness-aware time and power allocation in wireless powered communication networks," *IEEE Transactions on Communications*, vol. 66, no. 7, pp. 3122–3135, Jul. 2018.

[31] B. Van Houdt, "On the stochastic and asymptotic improvement of first-come first-served and nudge scheduling," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 6, no. 3, pp. 1–22, 2022.

[32] J. Fang, Y. Yu, C. Zhao, and J. Zhou, "Turbotransformers: An efficient GPU serving system for transformer models," in *Proc. 26th ACM SIGPLAN Symp. Princ. Pract. Parallel Program.*, 2021, pp. 389–402.

[33] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, 2016, pp. 785–794.

[34] P. Paikrao, S. Routray, A. Mukherjee, A. R. Khan, and R. Vohnout, "Consumer personalized gesture recognition in UAV based industry 5.0 applications," *IEEE Trans. Consum. Electron.*, vol. 69, no. 4, pp. 842–849, Nov. 2023.

Zhengzhe Xiang (Member, IEEE) received the B.S. and Ph.D. degrees in computer science and technology from Zhejiang University, Hangzhou, China. He was a Visiting Scholar from Shanghai Jiao Tong University, Shanghai, China, in 2022. He is currently an Associate Professor with Hangzhou City University, Hangzhou. His research interests lie in the fields of service computing and edge computing. He serves as a Reviewer for several international journal like the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE TRANSACTIONS ON SERVICES COMPUTING, *IET Communications*, *Digital Communications and Networks*. He also serves as a PC Members of many international conferences.

Xizi Xue is currently pursuing the M.S. degree with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. Her research interests include Internet of Things technology, edge computing, service computing, and reinforcement learning.

Zengwei Zheng received the B.S. and M.Ec. degrees in computer science and western economics from Hangzhou University, China, and the Ph.D. degree in computer science and technology from Zhejiang University, China, in 2005. He is currently a Full Professor with the School of Computer and Computing Science, Hangzhou City University, and the Director of Intelligent Plant Factory, Zhejiang Province Engineering Laboratory, and the Hangzhou Key Laboratory for IoT Technology and Application.

Honghao Gao (Senior Member, IEEE) is currently with the School of Computer Engineering and Science, Shanghai University, China. He is also a Professor with the College of Future Industry, Gachon University, South Korea. He has more than 200 publications, including the IEEE Transactions on Industrial Informatics, the IEEE Transactions on Intelligent Transportation Systems, the IEEE Transactions on Neural Networks and Learning Systems, the IEEE Transactions on Services Computing, and the IEEE Transactions on Fuzzy Systems. His research interests include software intelligence, cloud/edge computing, industrial Internet, and AI4Healthcare. He is a Fellow of the *Institution of Engineering and Technology*, and the British Computer Society, and a member of the *European Academy of Sciences and Arts*.

Yuanyi Chen received the B.Sc. degree from Sichuan University, Chengdu, China, in 2010, the M.Sc. degree from Zhejiang University, Hangzhou, China, in 2013, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2017. He is currently a Full Professor with the Department of Computer Science and Computing, Hangzhou City University, Hangzhou. He has published more than 30 technical papers in major international journals and conference proceedings. His research interest includes the Internet of Things and edge computing.

Schahram Dustdar (Fellow, IEEE) is a Full Professor of computer science and the Head of the TU Wien's Distributed Systems Group. His research interests include distributed systems, edge intelligence, and complex and autonomic software systems. He has received the ACM Distinguished Scientist Award, the Distinguished Speaker Award, and the IBM Faculty Award. He is an Editor-in-Chief of Computing, and an Associate Editor of *ACM Transactions on the Web*, *ACM Transactions on Internet Technology*, the IEEE Transactions on Cloud Computing, and the IEEE Transactions on Services Computing. He is also an Editorial Boards of IEEE Internet Computing and IEEE Computer. He is an Elected Member of Academia Europaea, where he was an Informatics Section Chairman from 2015 to 2022. He is a AAIA Fellow where he is the Current President.