applied sciences

Article

End-to-End Latency Optimization for Resilient Distributed Convolutional Neural Network Inference in Resource-Constrained Unmanned Aerial Vehicle Swarms

Jeongho Kim [1], Joonho Seon [1], Soohyun Kim [1], Seongwoo Lee [1], Jinwook Kim [1], Byungsun Hwang [1], Youngghyu Sun [2] and Jinyoung Kim [1,*]

[1] Department of Electronic Convergence Engineering, Kwangwoon University, Seoul 01897, Republic of Korea; jh980828@kw.ac.kr (J.K.); dimlight13@kw.ac.kr (J.S.); kimsoogus@kw.ac.kr (S.K.); swoo1467@kw.ac.kr (S.L.); yoonlight12@kw.ac.kr (J.K.); hbsun1225@kw.ac.kr (B.H.)

[2] Research and Development Department, SMART EVER, Co., Ltd., Seoul 01886, Republic of Korea; ygsun@smartever.net

* Correspondence: jinyoung@kw.ac.kr; Tel.: +82-2-940-5567

Abstract: An unmanned aerial vehicle (UAV) swarm has emerged as a powerful tool for mission execution in a variety of applications supported by deep neural networks (DNNs). In the context of UAV swarms, conventional methods for efficient data processing involve transmitting data to cloud and edge servers. However, these methods often face limitations in adapting to real-time applications due to the low latency of cloud-based approaches and weak mobility of edge-based approaches. In this paper, a new system called deep reinforcement learning-based resilient layer distribution (DRL-RLD) for distributed inference is designed to minimize end-to-end latency in UAV swarm, considering the resource constraints of UAVs. The proposed system dynamically allocates CNN layers based on UAV-to-UAV and UAV-to-ground communication links to minimize end-to-end latency. It can also enhance resilience to maintain mission continuity by reallocating layers when inoperable UAVs occur. The performance of the proposed system was verified through simulations in terms of latency compared to the comparison baselines, and its robustness was demonstrated in the presence of inoperable UAVs.

Keywords: resource-constrained UAV swarm; distributed inference; resilient UAV system; deep reinforcement learning; end-to-end latency optimization.

Citation: Kim, J.; Seon, J.; Kim, S.; Lee, S.; Kim, J.; Hwang, B.; Sun, Y.; Kim, J. End-to-End Latency Optimization for Resilient Distributed Convolutional Neural Network Inference in Resource-Constrained Unmanned Aerial Vehicle Swarms. *Appl. Sci.* **2024**, *14*, 10832. https://doi.org/10.3390/app142310832

Academic Editors: Yuliang Tang and Minghui Liwang

Received: 25 October 2024
Revised: 16 November 2024
Accepted: 21 November 2024
Published: 22 November 2024

1. Introduction

In recent decades, unmanned aerial vehicles (UAVs) have been employed in various applications [1–7] thanks to their flexibility, maneuverability, wide coverage, and real-time data collection [8]. When equipped with various sensors, UAVs can monitor target objects and transmit information in real-time during missions [9]. In the initial stages of UAV applications, missions were conducted with a single UAV, but this approach may be challenging to apply to large-scale operations, as the malfunction of a single UAV could result in the abortion of the mission. Unlike a single UAV, a UAV swarm can provide several advantages, such as reducing operation time, decreasing mission failure rate, and multitasking capability through cooperation and parallel processing [10,11]. Furthermore, the UAV swarm is suitable for large-scale operations due to their wide coverage and enhanced cost-effectiveness [12–14].

The use of convolutional neural networks (CNNs) in a variety of UAV missions has been the main objective of numerous studies [15,16]. DNN architectures have become more complicated as UAV missions in a complex plan become more demanding. Naturally, the memory usage and computational requirements of models have increased, which may pose challenges when deploying CNN models on resource-constrained UAVs [17]. Several solutions have been proposed to address the discrepancy between the substantial resource requirements of CNN models and the restricted resource supply of UAVs. Cloud-based intelligence has addressed these challenges by transferring raw UAV data to a cloud server for CNN inference [18–20]. This approach offers computing resources from powerful cloud servers. However, it can suffer from high transmission latency due to the long distance between the UAV and the cloud server. Moreover, unstable network conditions in battlefield or disaster areas can disrupt data transmission, thereby limiting the feasibility of cloud-based intelligence. Edge-based intelligence has implemented CNN inference on edge servers near UAVs to circumvent the resource requirements of CNNs [21–23]. However, this approach may constrain the mobility of UAV swarms because they need to remain within the operational range of edge servers to maintain connectivity. Additionally, server overload may occur when multiple UAVs simultaneously offload to a single edge server.

To address problems related to transmission latency and UAV mobility constraints, recent works [24–26] have focused on jointly executing inference for complex CNN models on resource-constrained devices without relying on cloud or edge servers. This approach, called device–device collaborative edge intelligence (CoEI), performs distributed inference by partitioning the CNN model into segments (e.g., layers, multiplication tasks) and assigning the segments to devices. The device–device CoEI is suitable for swarms consisting of multiple UAVs and enables CNN inference through the collective power of the swarm, even when the resources of UAVs are limited. In [24], a pipeline-based collaborative method has been proposed to conduct CNN inference on a resource-constrained UAV swarm. However, this method was tested with a small number of UAVs, which can lead to concerns about scalability in a larger UAV swarm. In [25], an optimal UAV-based layer distribution method was proposed to minimize the end-to-end latency of the distributed inference by allocating partitioned CNN layers. However, this method does not consider UAV-to-ground (U2G) communication link needed to transmit results from the UAV network to ground devices. Incorporating the U2G link can more effectively optimize overall end-to-end latency. Dhuheir et al. [26] proposed a method for distributed inference by optimizing layer assignment using deep reinforcement learning (DRL). However, this method may overlook scenarios where a UAV becomes inoperable due to internal or external factors. The absence of a resilience mechanism for inoperable UAVs may lead to mission failure.

In mission-critical applications, such as disaster response, search and rescue, and surveillance, minimizing the end-to-end latency and ensuring system resilience are essential. The aforementioned methods for distributed CNN inference may fail to perform their mission due to the lack of mechanisms for ensuring system resilience. Therefore, the proposed system aims to enhance mission continuity by prioritizing latency minimization and system resilience. In this paper, a DRL-based resilient layer distribution (DRL-RLD) system is proposed to enable resilient distributed inference in UAV swarm. The proposed system is designed to minimize end-to-end latency through distributed inference, considering the UAV's position, available resources, and operability. The proposed system also incorporates both UAV-to-UAV and UAV-to-ground communication links, ensuring that final outputs reach ground devices with minimal latency. The proposed system is designed to ensure network resilience by dynamically reallocating CNN layers in real time when UAV malfunctions occur. This mechanism can ensure mission continuity. In addition, the robustness and scalability of the proposed system are demonstrated by evaluating its performance across various environmental conditions and swarm sizes. The main contributions of this paper are summarized as follows:

- The proposed system minimizes the end-to-end latency of distributed CNN inference by considering both U2U and U2G communication links. Therefore, the proposed system can optimize the entire process from image acquisition to transmitting the final prediction to the ground device.
- The DRL-RLD system is proposed to optimize resilient distributed inference in resource-constrained UAV swarm. The resilience of the UAV network can be enhanced through dynamic reallocation of layers even when UAVs become inoperable.

This dynamic layer reallocation ensures the continuous operation of the network and enables adaptive real-time mission execution.

- Extensive simulations have been conducted with various CNN models, available resources, and UAV configurations to evaluate the performance of the proposed system. In particular, the proposed system has been simulated with a larger number of UAVs than the settings in the previous research, with practical conditions of UAV malfunction. Therefore, it has the potential to offer greater scalability and generalization performance compared to conventional methods.

The remainder of the paper is structured as follows: The system model is described in Section 2. The optimization problem is formulated in Section 3. The proposed deep reinforcement learning model for layer distribution is detailed in Section 4. The simulation results are presented in a variety of scenarios in Section 5. The conclusions and future directions are summarized in Section 6.

2. System Model

The system model is depicted in Figure 1, where UAV swarms are configured to form a wireless ad hoc network. The network consists of N UAVs, which are employed in target area monitoring, intermediate data transmission, and classification tasks (e.g., human detection, forest fire detection, highway accident detection). At each time step t, requests are received, which are defined as units of classification tasks to be processed by UAV swarm. RQ^t is the total number of requests at time step t. Each UAV $i \in \{1, \ldots, N\}$ operates under two significant constraints: a maximum memory usage m_i^{max} and a maximum computational capacity c_i^{max}. The position of the UAV is represented by 3D coordinates (x, y, z).

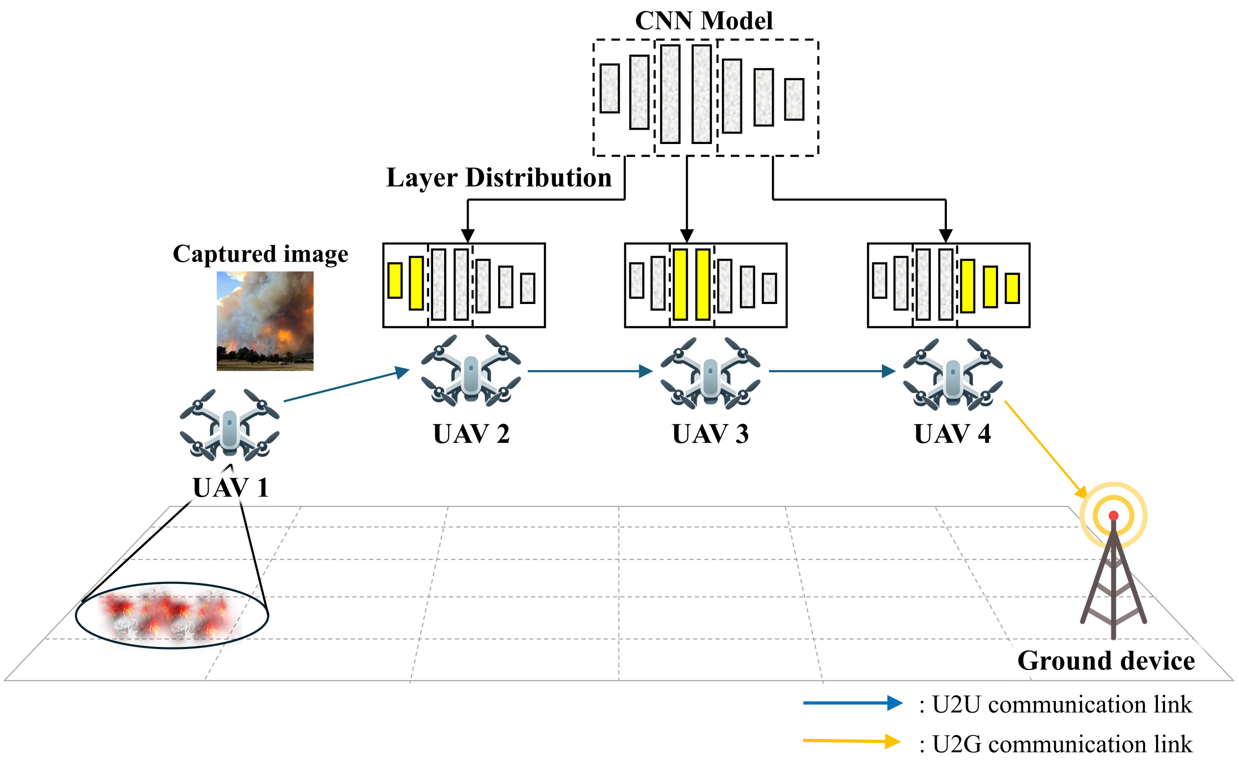


Figure 1. The proposed system model for resilient distributed CNN inference.

The proposed system is operated under the following assumptions: First, UAVs can explore the entire mission area, but not all cells are of equal importance. For instance, the system can intensively monitor hot cells that are established to be of high importance (e.g., areas where fires occurred). Second, CNN layers are assigned to each UAV by the

DRL model, and the requests are processed in accordance with the layer assigned to each UAV. Herein, the DRL model distributes layers to minimize end-to-end latency. The end-to-end latency is defined as the sum of transmission and computational latency. Here, transmission latency represents the time required to transmit intermediate data between UAVs or between a UAV and a ground device, and computational latency represents the time required to process data within each UAV. This end-to-end latency can be affected by several factors, including the distribution of CNN layers, the distance between UAVs, and the available computing and memory resources of each UAV. Third, the proposed system is designed to ensure mission continuity, even if certain UAVs become inoperable due to hardware failures or cyber-attacks. The DRL-RLD model leverages current state information that includes an inoperable state parameter for each UAV. When this parameter is set to be zero, it indicates that a specific UAV is no longer operational. In such scenarios, the DRL-RLD model reallocates layers based on the current state of the remaining UAVs, including available memory, computational capacity, and location.

Depending on the mission of the UAV swarm, an appropriate CNN model can be selected. Each UAV in the swarm contains a copy of the trained model and performs tasks associated with the activated layers for distributed inference. The proposed distributed system is designed with the pipeline, incorporating convolutional and fully connected layers (FCN) with rectified linear unit (ReLU) activations, and is compatible with CNN architectures that do not incorporate residual blocks. The CNN model consists of L layers. For each $j \in \{1, \ldots, L\}$, K_j represents the size of the intermediate output generated at layer j and transmitted to the subsequent layer $j+1$ of the CNN model.

Each CNN layer has memory requirements m_j and computational load requirements c_j. The memory requirements of layers m_j are determined by the number of weights W_j of the current layer j and the number of bits b dedicated to storing the weight, and it is represented as $m_j = W_j \cdot b$. The computational requirements c_j are the number of multiplications needed to execute the layer j [27]. Therefore, the computational load requirement of the convolutional layers is calculated as follows:

$$c_{j\_\text{conv}} = k_h \cdot k_w \cdot Ch_{\text{in}} \cdot F_h \cdot F_w \cdot Ch_{\text{out}}, \tag{1}$$

where k_h and k_w are the height and width of the filter, respectively, and Ch_{in} is the number of input channels at layer j, F_h and F_w are the height and width of the output feature map, respectively, and Ch_{out} is the number of output channels at layer j. The computational load calculations of the fully connected layer j are the number of neurons n_j of layer j multiplied by the number of neurons n_{j-1} of the layer $j-1$, and it is represented as $c_{j\_\text{fcn}} = n_{j-1} \cdot n_j$.

Communications Model for U2U and U2G Link

A communication model is used to implement U2U and U2G communication links. The elevation angle and environmental factors determine the probability of line of sight (LoS) between two UAVs, or UAV and ground device. The probability of LoS, denoted as P_{LoS}, can be modeled by

$$P_{\text{LoS}} = \frac{1}{1 + a\exp(-b(\theta - a))}, \tag{2}$$

$$\theta = \frac{180}{\pi} \sin^{-1}\left(\frac{h}{d}\right), \tag{3}$$

where environmental factors a and b are set to be 4.88 and 0.43, respectively, under the assumption of a rural environment with few obstacles and wide-open space [28]. θ denotes the elevation angle between the UAV and the ground station or another UAV. In addition, d denotes the Euclidean distance between the UAV and the ground station or another UAV. The probability of non-line of sight (NLoS) P_{NLoS} is $P_{\text{NLoS}} = 1 - P_{\text{LoS}}$. The path loss PL is calculated by

$$PL = P_{\text{LoS}} \cdot PL_{\text{LoS}} + P_{\text{NLoS}} \cdot PL_{\text{NLoS}}, \tag{4}$$

where PL_{LoS} and PL_{NLoS} are the path losses under NoS and NLoS channel conditions, respectively. PL_{LoS} and PL_{NLoS} are calculated by

$$PL_{\mathrm{LoS}} = 20\log_{10}(d) + 20\log_{10}(f) + 20\log_{10}\left(\frac{4\pi}{c}\right), \tag{5}$$

$$PL_{\mathrm{NLoS}} = PL_{\mathrm{LoS}} + \eta_{\mathrm{NLoS}}, \tag{6}$$

where f denotes the carrier frequency, c is the speed of light, and ν_{NLoS} is an additional attenuation factor that is set to 20 dB, assuming a rural environment [28]. The path loss PL affects the quality of a communication link, and the data transmission rate ρ can be calculated by the Shannon formula,

$$\rho = B\log_2\left(1 + \frac{P_t}{PL \cdot N_0 \cdot B}\right), \tag{7}$$

where P_t represents the transmission power, B is the bandwidth, and N_0 is the noise power spectral density.

3. Problem Formulation

An optimization problem is formulated for minimizing the end-to-end latency in making the final prediction through distributed inference within a resource-constrained UAV swarm. Based on an optimization discussed in [25], this new optimization problem considers the UAVs availability and both the U2G and U2U links. The proposed optimization problem depends on three decision variables $\delta_{t,r,i,j}$, $\psi_{t,i,q}$, and $o_{t,i}$, which are defined as follows:

$$\delta_{t,r,i,j} = \begin{cases} 1, & \text{if UAV } i \text{ executes layer } j \text{ of request } r \text{ at time } t \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

$$\psi_{t,i,q} = \begin{cases} 1, & \text{if UAV } i \text{ exists at cell } q \text{ at time } t \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

$$o_{t,i} = \begin{cases} 1, & \text{if UAV } i \text{ is operable at time } t \\ 0, & \text{if UAV } i \text{ is inoperative at time } t. \end{cases} \tag{10}$$

The proposed optimization problem can be an integer linear programming (ILP) optimization, and it is formulated as

$$\min_{\psi_{t,i,q},\delta_{t,r,i,j}} \sum_{t=1}^{T}\sum_{r=1}^{RQ_t}\sum_{i=1}^{N}\sum_{\substack{k=1\\k\neq i}}^{N}\sum_{j=1}^{L-1} \delta_{t,r,i,j}\cdot o_{t,k}\cdot \delta_{t,r,k,j+1}\cdot \frac{K_{r,j}}{\rho_{i,k}} + \sum_{i=1}^{N} o_{t,i}\cdot t_i^{(p)} + t_l, \tag{11}$$

Constraints:

$$\sum_{r=1}^{RQ_t}\sum_{j=1}^{L} \delta_{t,r,i,j}\cdot m_j \le m_i^{\max}, \tag{11a}$$

$$\sum_{r=1}^{RQ_t}\sum_{j=1}^{L} \delta_{t,r,i,j}\cdot c_j \le c_i^{\max}, \tag{11b}$$

$$\sum_{i=1}^{N} \delta_{t,r,i,j} = \begin{cases} 1, & \text{if } j \le L \\ 0, & \text{otherwise,} \end{cases} \tag{11c}$$

$$\sum_{q=1}^{C} \psi_{t,i,q} = 1, \tag{11d}$$

$$\sum_{i=1}^{N} \psi_{t,i,q} \leq 1, \tag{11e}$$

$$\delta_{t,r,i,j} \leq o_{t,i}, \tag{11f}$$

and where

$$t_l = \sum_{i=1}^{N} o_{t,i} \cdot \delta_{t,r,i,L} \cdot \frac{K_{r,L}}{\rho_{i,G}}, \tag{12}$$

$$t_i^{(p)} = \sum_{t=1}^{T} \sum_{r=1}^{RQ_t} \sum_{j=1}^{L} \delta_{t,r,i,j} \cdot \frac{c_j}{c_i^{\max}}. \tag{13}$$

The proposed optimization problem is divided into three parts to systematically address the latency components associated with distributed CNN inference in a UAV swarm. The first part focuses on inter-UAV transmission latency for transmitting intermediate data between layers, which is crucial for efficient distributed processing. The second part addresses the latency of transmitting the final output to the ground station. Lastly, the third part addresses the computational latency required to process the layers assigned to each UAV.

1. Inter-UAV transmission latency: the time required to transmit the intermediate data of layer j assigned to UAV i to UAV k can be represented in

$$\frac{K_{r,j}}{\rho_{i,k}}, \tag{14}$$

   where $K_{r,j}$ denotes the intermediate data generated from layer j of request r, and $\rho_{i,k}$ denotes the transmission data rate between the UAV i and the UAV k.
2. Final output transmission latency to ground devices: in Equation (12), t_l is the time required to transmit the output of the last layer to the ground device G, and $K_{r,L}$ is the size of the final output for classification tasks.
3. Computational latency of individual UAVs: In Equation (13), $t_i^{(p)}$ is the total time required to compute all processing tasks assigned to UAV i. This time is defined as the ratio of the computational requirement c_j of the layer j to the computational capacity c_i^{max} of the UAV i.

The constraint in Equations (11a) and (11b) ensures that each UAV i does not exceed its memory usage and computational capacity while processing its assigned CNN layer, respectively. The constraint in Equation (11c) guarantees that each layer is assigned to only one UAV. To prevent collisions, the constraint in Equation (11d) ensures that only one UAV visits each cell at each time step. The constraint in Equation (11e) restricts each UAV to visiting only one cell at any given time t. Lastly, the constraint in Equation (11f) ensures that no layers are assigned to inoperable UAVs.

The proposed ILP optimization problem can be formulated as a combinatorial problem that includes multiple binary variables and complex constraints. As the size of the proposed problem which denotes the number of UAVs, CNN layers, and requests increases, the combinations that can be searched for expand exponentially. The increase in the search space may lead to a combinatorial explosion. This means that the time required to find the optimal solution increases exponentially, resulting in a highly time-intensive task. This complexity poses significant challenges when implementing the proposed system in real-time applications. Recently, DRL has demonstrated impressive performance in managing dynamic and realistic systems [29–31]. DRL-based systems can learn optimal solutions by analyzing environmental states and predicting the best actions based on previous rewards. Therefore, a DRL-based system is adopted to efficiently solve the proposed optimization problem and ensure optimal performance.

4. Deep Reinforcement Learning Model for Layer Distribution

In this section, the DRL-RLD model depicted in in Figure 2 is proposed to address the optimization problem. The DRL-RLD model is trained offline and performs layer distribution tasks during UAV missions. The DRL-RLD model can dynamically distribute layers to ensure optimal performance depending on environmental changes and UAV availability. The DRL-RLD model is built using a Markov decision process (MDP) framework, represented as (S, A, P, R, γ), where S represents a state space, A represents an action space, P represents transition probabilities between states, R represents rewards obtained after each action, and γ represents a discount factor. At each time step t, an agent observes a current state s_t and selects an action a_t based on observed information. Depending on the chosen action, the agent obtains a reward r_t, and learns optimal policy π^* through feedback based on the reward. The agent's goal is to identify optimal policy π^*, which maximizes value function $V^{\pi^*}(s)$ for all states. The value function $V^{\pi}(s)$ represents expected cumulative reward when following policy π starting from state s, and is mathematically expressed as

$$V^{\pi}(s) = \mathbb{E}_{a_t,s_{t+1}}\left(\sum_{k=0}^{\infty} \gamma^{k-t} r_k \mid s_t = s\right), \tag{15}$$

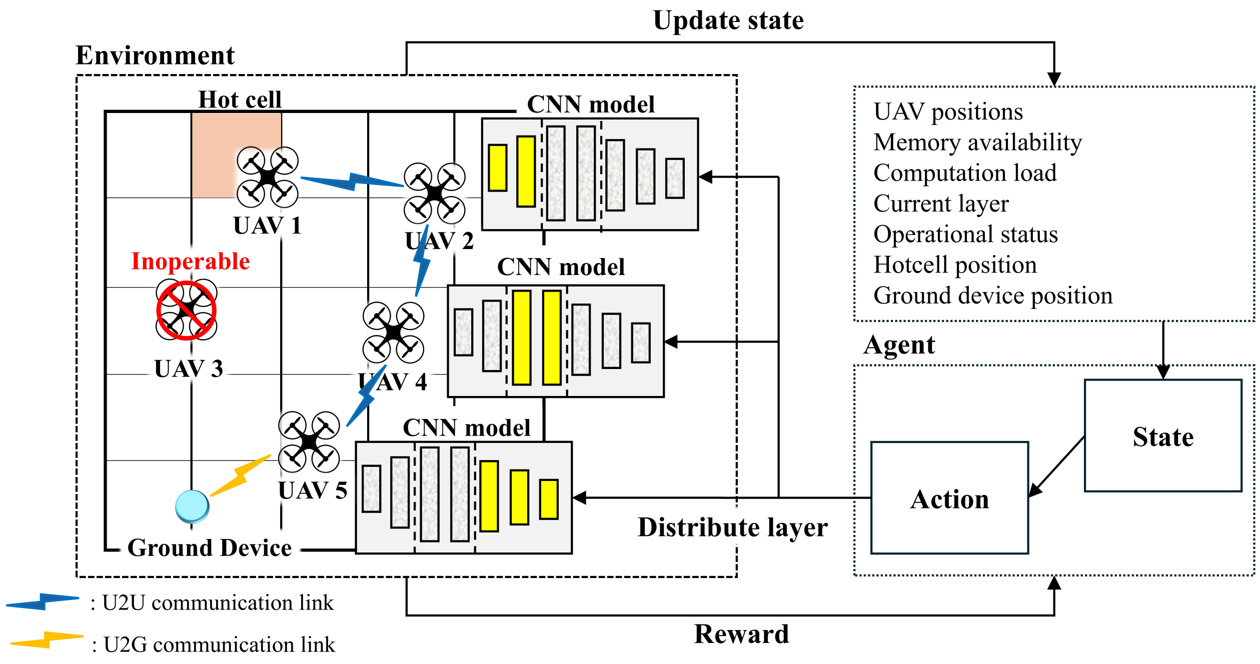


Figure 2. The proposed DRL-RLD model for layer distribution.

4.1. Design for DRL-RLD Model

The state space S consists of the following components to provide the agent with comprehensive state information that accurately reflects the environment: the position of the UAVs in the grid, the current memory usage of the UAV i, and the computational capacity of the UAV i. The CNN layers currently assigned to UAV i, the operational status of UAV i, the position of the ground device in the grid, and the positions of hot cells in the grid.

The action a is defined as the allocation of CNN layers to the entire UAV swarm. The agent is capable of undertaking actions based on the current state of each UAV, including the computation capacity, memory usage, and operational status.

The reward is designed to provide a higher reward when the end-to-end latency of distributed inference is low and impose penalties when the constraints of the system are violated. This structure can enable the DRL-RLD model to learn how to distribute layers in a way that minimizes end-to-end latency. The penalties are imposed when the constraints below are not satisfied.

- Constraint 1: All layers within a CNN model must be allocated to UAV swarm, and each layer must be allocated to exactly one UAV. This constraint is violated if unassigned layers occur or a layer is assigned to multiple UAVs.
- Constraint 2: for each UAV i, the sum of the memory requirements and computational requirements of the assigned layers must not exceed the UAV's memory usage and computational capacity, respectively.
- Constraint 3: CNN Layers can only be assigned to operational UAVs, and assigning layers to inoperative UAVs violates this constraint.

A penalty mechanism is structured so that if any constraint is violated, the reward for that specific time step is set to be zero. For example, if a UAV exceeds its resource limits as specified in Constraint 2, the corresponding variable is set to zero ($\text{Cons}_2 = 0$), effectively setting the reward to zero. Conversely, when all constraints are satisfied, the variables for each constraint are set to 1, ensuring that rewards are granted only when all conditions are satisfied. This penalty mechanism ensures that the model learns to distribute layers while strictly satisfying the constraints. A reward function applying the penalty mechanism R is defined as:

$$R = \frac{A}{L_{\text{comp}} + L_{\text{tran}}} \cdot \text{Cons}_1 \cdot \text{Cons}_2 \cdot \text{Cons}_3, \tag{16}$$

where Cons_i is a parameter that denotes whether Constraint i is violated, L_{comp} is total computation latency, L_{tran} is total transmission latency, A is a constant, and the value of the reward function increases as L_{comp} and L_{tran} decrease.

4.2. Proximal Policy Optimization Algorithm

Proximal policy optimization (PPO) [32] is one of the policy gradient methods designed to optimize the policy π while maintaining stability during training. The main objective function of PPO is defined as

$$J^{\text{PPO}} = \mathbb{E}_t[\min(\eta * Ad^{\pi}(s_t, a_t), \text{clip}(\eta, 1-\epsilon, 1+\epsilon)Ad^{\pi}(s_t, a_t))], \tag{17}$$

$$\eta = \frac{\pi(a_t \mid s_t)}{\pi_{\text{old}}(a_t \mid s_t)}, \tag{18}$$

where $Ad^{\pi}(s_t, a_t)$ is the advantage function, ν denotes the policy probability ratio, and ϵ denotes the clipping parameter that restricts ν within the range of $[1-\epsilon, 1+\epsilon]$. The clipping ensures that policy updates are not too large, thereby avoiding instability during training. The advantage function $Ad^{\pi}(s_t, a_t)$ provides insight into the expected future rewards for acting a_t in state s_t, and it is defined as

$$Ad^{\pi}(s_t, a_t) = Q(s_t, a_t) - V^{\pi}(s_t), \tag{19}$$

where $Q(s_t, a_t)$ is the action-value function that represents the expected cumulative rewards obtained according to the policy π when performing specific action a_t in state s_t, and $V^{\pi}(s_t)$ is the value function that represents the expected cumulative rewards from state s_t under policy π.

The PPO algorithm iteratively updates the policy π_θ and the value function using experiences stored in the rollout buffer. During each policy update, the probability ratio η is calculated, and the clipping function is applied to prevent large updates. The clipping parameter ϵ can constrain policy updates to ensure training stability without drastic changes in a single step. The training process of the DRL-RLD model is summarized in Algorithm 1.

Algorithm 1 DRL-RLD model training algorithm.

```
1:  Initialization
2:  initialize the parameters $\theta$ of the policy network
3:  initialize the parameters $\theta_{\text{old}}$ of the policy network: $\theta_{\text{old}} \leftarrow \theta$
4:  initialize the parameters $\phi$ of the value network
5:  initialize rollout buffer
6:  randomly distributed UAVs in the grid
7:  for each UAV in $C$ do
8:      distributed all UAVs in the grid
9:  end for
10: PPO training
11: for each episode $r \in R$ do
12:     for each timestep $t \in T$ do
13:         randomly render arbitrary UAVs inoperable
14:         select and execute action $a_t$
15:         compute reward $r_t$
16:         store transition in buffer
17:         PPO update
18:     end for
19: end for
```

5. Simulation Results

5.1. Simulation Settings

The simulations are conducted in a scenario where a UAV located in a hot cell captures images and performs distributed inference with a swarm of surrounding UAVs. The UAVs are assumed to perform the mission while maintaining a fixed height $z = 40$ m. The UAV swarm operate in a 100 m $\times$ 100 m area, divided into 25 cells, with one of the 25 cells set as a hot cell. Three CNN models were employed in the simulation: AlexNet [33], VGG16 [34], and YOLO [35]. These models were selected because they are widely used in the object classification or object detection. The AlexNet and VGG16 were trained using the ILSVRC 2014 dataset [36], and YOLO was trained using the PASCAL VOC 2007 dataset [37]. The parameters and depths of these architectures are listed in Table 1. Here, M refers to one million. It is assumed that each UAV has two memory usage levels (low: 1 GB; high: 2 GB) and one computational capacity level (11.2 GFLOPS). The inoperable state parameter is set to be two, and the model is trained in an environment where zero to two UAV becomes inoperable during each episode. Parameters for training the proposed DRL-RLD model are listed in Table 2. The values of the training parameters were determined empirically through extensive trials. The bandwidth and transmission power were configured to assume communication channel within a low-power, resource-constrained environment, as typically encountered in disaster response and surveillance applications [26]. The noise power spectral density was selected based on the standard thermal noise level. The end-to-end latency is employed to evaluate the performance of the proposed DRL-RLD system. This evaluation metric measures the total time required for an image acquired in a hot cell to undergo distributed inference across the layers assigned to the UAV swarm and for the final output to reach the ground device.

Table 1. Comparison of CNN models.

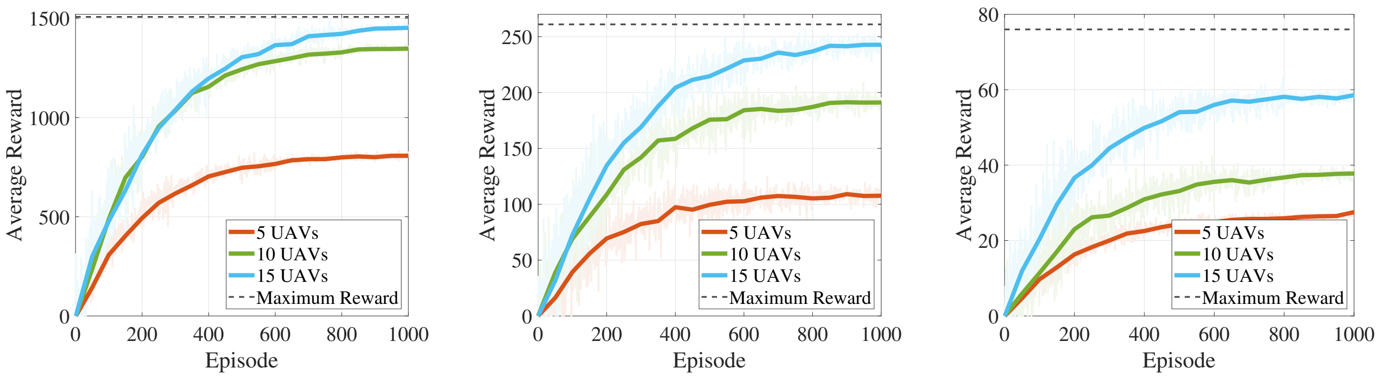
Name	Parameters	Depth
AlexNet [33]	74.7 M	8
VGG16 [34]	169.8 M	16
YOLO [35]	295.5 M	20

Table 2. Simulation parameters of DRL-RLD model.

Name	Parameters
discount factor	0.99
actor learning rate	0.0003
critic learning rate	0.001
clip range	0.2
inoperable state parameter	2
noise power spectral density	4×10^{-21}
transmission power	0.1 W
bandwidth	1 kHz

5.2. Performance Evaluation

The average cumulative rewards per episode are shown in Figure 3, where AlexNet, YOLO, and VGG16 were employed as deployed CNN models for the mission. The maximum reward line in Figure 3 is the value that the UAV swarms are optimally distributed within the mission area. The proposed system is unable to achieve the maximum reward due to a penalty resulting from the UAV's positions being initialized by following a Gaussian distribution. However, it was confirmed that as the number of UAVs in the swarm increased, the converged reward improved significantly. This performance improvement is attributed to flexibility of data transmission paths and resource utilization efficiency within the UAV swarm. The optimal solution was achieved by the AlexNet architecture with a slight difference from the maximum reward, a marginal difference was conformed with the YOLO architecture, and a moderate difference was exhibited with the VGG16 architecture. Additional UAVs may be necessary to reach optimal performance in VGG16, as it requires more average computational requirements required by a layer than other deployed CNN models. The limitations observed in the simulations are that as the swarm size increases, the instability in the early stages of learning increases and the model requires more episodes to reach saturation. This phenomenon can be attributed to the increased environmental complexity, which is a consequence of the expanded state and action spaces concomitant with an increased number of UAVs in the swarm.



(**a**) AlexNet (**b**) YOLO (**c**) VGG16

Figure 3. Average cumulative rewards by CNN models: (**a**) AlexNet; (**b**) YOLO; (**c**) VGG16.

The performance of the proposed model compared to three baselines is shown in Figure 4. The performance of the proposed model is evaluated under the same conditions, including fifteen UAVs, 1 GB memory, and a computational capacity of 11.2 GFLOPS. The Layer-Avg method [24] refers to the method that distributes layers to each UAV as uniformly as possible according to the layers of the CNN model. The Comp-Avg method [24] distributes the computational load assigned to each UAV as uniformly as possible according to the computational load of the CNN layers. The Heuristic method [25] searches the nearest neighboring node with sufficient resources to process the remaining layers during intermediate data transmission. The proposed model outperformed the

Layer-Avg and Comp-Avg methods in all three CNN models. The Layer-Avg and Comp-Avg methods focus on evenly distributing the requirements of the CNN layers. However, the Layer-Avg and Comp-Avg methods do not adequately consider the physical distances between UAVs, which significantly impact transmission latency. In contrast, the proposed model achieves superior performance by accounting for transmission latency between UAVs. The heuristic method prioritizes short-term benefits when distributing layers. On the contrary, the proposed system achieves long-term benefits by distributing layers with a comprehensive consideration of available resources and UAV arrangement. Consequently, the proposed model was confirmed to have lower the end-to-end latency than the three baseline methods by thoroughly considering factors such as UAV locations, communication links, and resource constraints during layer distribution.

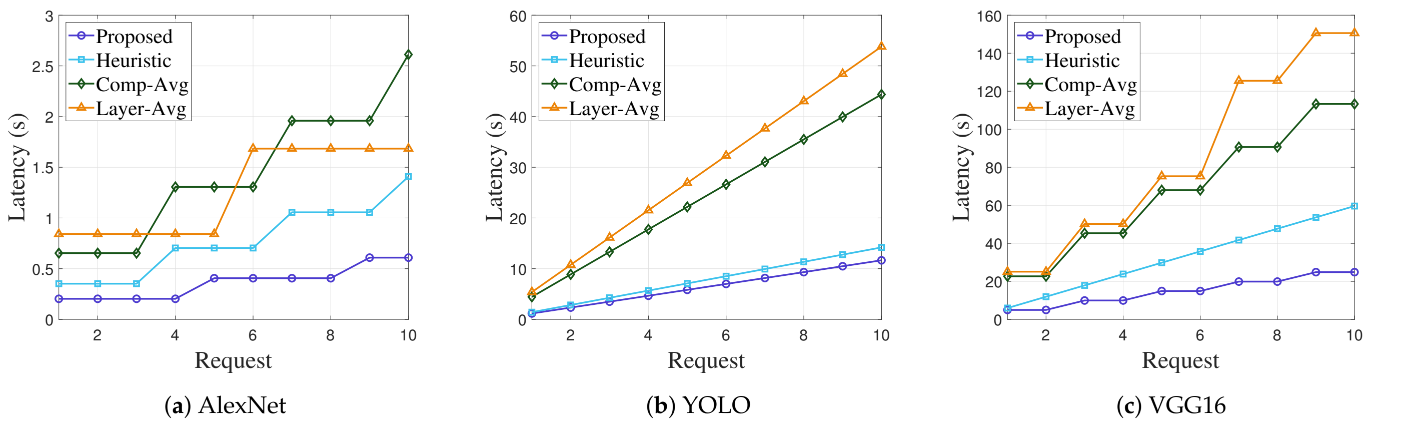
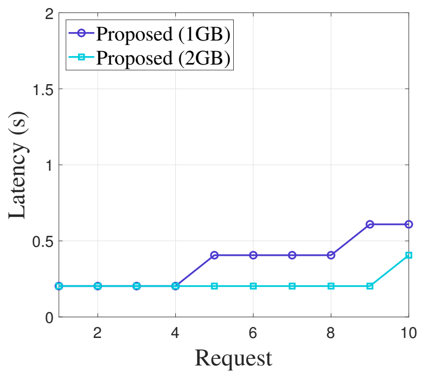


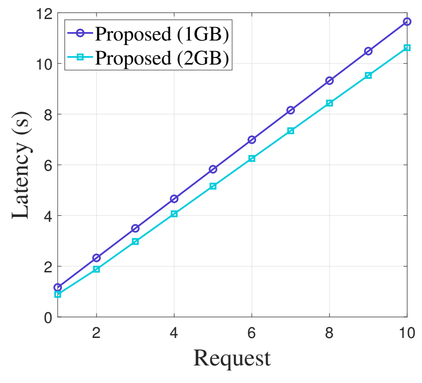
Figure 4. The performance of the proposed model compared to three baselines: (**a**) AlexNet; (**b**) YOLO; (**c**) VGG16.

A comparative performance of the proposed system under different memory constraints (1 GB and 2 GB) is presented in Figure 5. Simulations were performed under the same conditions, except for memory constraints, with 15 UAVs and a computational capacity of 11.2 GFLOPS. For AlexNet, there was no observable performance variation between using 1 GB and 2 GB of memory until four requests were processed. This implies that 1 GB of memory is sufficient for up to four parallel requests. After five requests, it was confirmed that performance improved with the use of 2 GB of memory. This phenomenon can be attributed to the positive correlation between the expansion of available memory resources in the UAV swarm and the consequent increase in the number of requests processed in parallel. For YOLO and VGG16, it was confirmed that using 2 GB of memory performs better than 1 GB. The increased memory capacity allows a single UAV to process more CNN layers. This can reduce the number of intermediate data transmissions between UAVs and consequently lowers overall transmission latency.

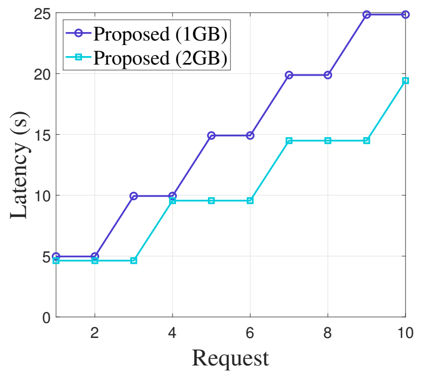
The end-to-end latency in a scenario with inoperable UAVs is shown in Figure 6. The simulation was conducted using AlexNet with 15 UAVs. Simulations were conducted with 15 UAVs, 1 GB of memory, and a computational capacity of 11.2 GFLOPS using the AlexNet model. The end-to-end latency for each model is illustrated in Figure 6 under three conditions: all UAVs fully operational, one inoperable UAV, and two inoperable UAVs. It was confirmed that the proposed model exhibits few increase in the end-to-end latency than the comparison of the three baseline methods. Therefore, the proposed model can maintain mission continuity by reallocating layers to the remaining UAVs using the available resources. In contrast, the comparison models lack these resilient mechanisms and are more adversely affected by inoperable UAVs. The proposed DRL-RLD model demonstrates superior performance compared to the baseline methods when UAVs becomes unexpectedly inoperable.



(**a**) AlexNet



(**b**) YOLO



(**c**) VGG16

Figure 5. Performance comparison of the proposed system under different memory constraints: (**a**) AlexNet; (**b**) YOLO; (**c**) VGG16.

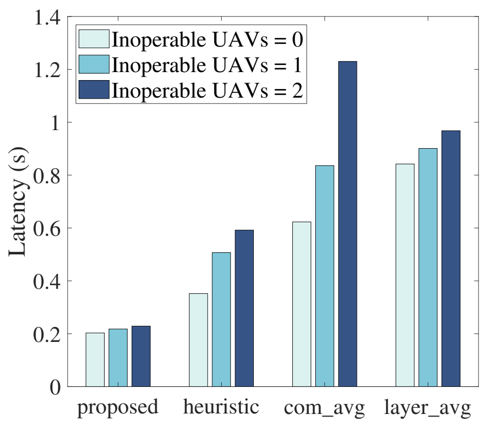


Figure 6. The performance of the proposed model in a scenario with inoperable UAVs.

The proposed DRL-RLD system outperformed existing methods by optimizing CNN layer allocation to reduce the end-to-end latency. As the number of UAVs in swarm increases, the converged reward approaches its maximum value. Additionally, the increase in the end-to-end latency can be diminished by dynamically reallocating layers when arbitrary UAVs become inoperative. Overall, the system operated efficiently across various UAV swarm configurations and resource constraints.

6. Conclusions

In this paper, a deep reinforcement learning-based resilient layer distribution (DRL-RLD) system was proposed to optimize resilient distributed CNN inference in resource-constrained UAV swarm. The proposed system minimizes the end-to-end latency by dynamically allocating CNN layers based on the position of the UAVs and their available resources, considering the U2U and U2G communication links. The system can also maintain the continuity of the network by reallocating layers when UAVs are inoperable due to internal or external factors. The performance of the proposed system was evaluated through simulations under different CNN models, as well as the number of UAVs and available resources. The simulation results proved that the proposed system can outperform the baselines in terms of end-to-end latency. It was further verified that mission continuity can be maintained through layer reassignment, even when inoperable UAVs occur. The proposed system can be a viable approach for disaster management, search and rescue, and surveillance due to its enhanced resilience and capabilities for real-time inference. In future work, the impact of various environmental factors, such as weather conditions, terrain, and obstacles (e.g., buildings or dense foliage), should be further examined to verify the robustness of the proposed system model. Additionally, a dynamic environment model that accounts for the Doppler effect is necessary to better understand the impact of UAV mobility on communication link quality. Furthermore, the battery life constraints of UAVs should be integrated into future works, as real-world deployments are often limited by the power supply available to each UAV. Addressing these factors will improve the robustness, and generalizability of the proposed system in practical, real-world applications.

Author Contributions: Conceptualization, J.K. (Jeongho Kim); methodology, J.K. (Jeongho Kim) and J.S.; software, J.K. (Jeongho Kim) and S.K.; validation, J.K. (Jeongho Kim), S.K. and J.S.; formal analysis, J.K. (Jeongho Kim) and S.L.; investigation, J.K. (Jeongho Kim), B.H. and J.K. (Jinwook Kim); resources, J.K. (Jeongho Kim), J.K. (Jinwook Kim) and Y.S.; data curation, J.K. (Jeongho Kim), S.K. and Y.S.; writing—original draft preparation, J.K. (Jeongho Kim); writing—review and editing, J.S., S.K., S.L., B.H., J.K. (Jinwook Kim) and Y.S.; visualization, J.K. (Jeongho Kim); supervision, J.K. (Jinyoung Kim); project administration, J.K. (Jinyoung Kim). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2021-0-00892-004, Research on advanced physical layer technologies of LEO(low-earth orbit) satellite communication systems for ubiquitous intelligence in space), and in part by Research Grant of Kwangwoon University in 2024.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://www.image-net.org/challenges/LSVRC/2014/ and http://host.robots.ox.ac.uk/pascal/VOC/voc2007/, accessed on 20 November 2024.

Conflicts of Interest: Author Youngghyu Sun was employed by the company SMART EVER, Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
CNN	Convolutional Neural Network
CoEI	Collaborative Edge Intelligence
DNN	Deep Neural Network
DRL	Deep Reinforcement Learning
DRL-RLD	Deep Reinforcement Learning-Based Resilient Layer Distribution
U2U	UAV-to-UAV Communication
U2G	UAV-to-Ground Communication
LoS	Line of Sight
NLoS	Non-Line of Sight
ILP	Integer Linear Programming
MDP	Markov Decision Process
ReLU	Rectified Linear Unit
PPO	Proximal Policy Optimization
FCN	Fully Connected Network

References

1. Wu, J.; Sun, Y.; Li, D.; Shi, J.; Li, X.; Gao, L.; Yu, L.; Han, G.; Wu, J. An adaptive conversion speed Q-learning algorithm for search and rescue uav path planning in unknown environments. *IEEE Trans. Veh. Technol.* **2023**, *72*, 15391–15404. https://doi.org/10.1109/TVT.2023.3297837.
2. Liu, D.; Zhu, X.; Bao, W.; Fei, B.; Wu, J. Smart: Vision-based method of cooperative surveillance and tracking by multiple uavs in the urban environment. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 24941–24956. https://doi.org/10.1109/TITS.2022.3203411.

3. Li, X.; Lu, X.; Chen, W.; Ge, D.; Zhu, J. Research on UAVs reconnaissance task allocation method based on communication preservation. *IEEE Trans. Consum. Electron.* **2024**, *70*, 684–695. https://doi.org/10.1109/TCE.2024.3368062.
4. Sawadsitang, S.; Niyato, D.; Tan, P.S.; Wang, P. Joint ground and aerial package delivery services: A stochastic optimization approach. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2241–2254. https://doi.org/10.1109/TITS.2018.2865893.
5. Xu, H.; Wang, L.; Han, W.; Yang, Y.; Li, J.; Lu, Y.; Li, J. A survey on UAV applications in smart city management: Challenges, advances, and opportunities. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 8982–9010. https://doi.org/10.1109/JSTARS.2023.3317500.
6. Ho, D.T.; Grøtli, E.I.; P.B, S.; Johansen, T.A.; Sousa, J.B. Optimization of wireless sensor network and UAV data acquisition. *J. Intell. Robot Syst.* **2015**, *78*, 159–179. https://doi.org/10.1007/s10846-015-0175-5.
7. Baniasadi, P.; Foumani, M.; Smith-Miles, K.; Ejov, V. A transformation technique for the clustered generalized traveling salesman problem with applications to logistics. *Eur. J. Oper. Res.* **2020**, *285*, 444–457. https://doi.org/10.1016/j.ejor.2020.01.053.
8. Banafaa, M.K.; Pepeoğlu, Ö.; Shayea, I.; Alhammadi, A.; Shamsan, Z.A.; Razaz, M.A.; Alsagabi, M.; Al-Sowayan, S. A comprehensive survey on 5G-and-beyond networks with UAVs: applications, emerging technologies, regulatory aspects, research trends and challenges. *IEEE Access* **2024**, *12*, 7786–7826. https://doi.org/10.1109/ACCESS.2023.3349208.
9. Dai, M.; Huang, N.; Wu, Y.; Gao, J.; Su, Z. Unmanned-aerial-vehicle-assisted wireless networks: Advancements, challenges, and solutions. *IEEE Internet Things J.* **2023**, *10*, 4117–4147. https://doi.org/10.1109/JIOT.2022.3230786.
10. Zhou, Y.; Rao, B.; Wang, W. UAV swarm intelligence: Recent advances and future trends. *IEEE Access* **2020**, *8*, 183856–183878. https://doi.org/10.1109/ACCESS.2020.3028865.
11. Liu, B.; Wang, S.; Li, Q.; Zhao, X.; Pan, Y.; Wang, C. Task Assignment of UAV Swarms Based on Deep Reinforcement Learning. *Drones* **2023**, *7*, 297–317. https://doi.org/10.3390/drones7050297.
12. Wang, X.; Tan, G.z.; Lu, F.L.; Zhao, J.; Dai, Y.s. A Molecular Force Field-Based Optimal Deployment Algorithm for UAV Swarm Coverage Maximization in Mobile Wireless Sensor Network. *Processes* **2020**, *8*, 369–389. https://doi.org/10.3390/pr8030369.
13. Zhang, X.; Duan, L. Energy-Saving Deployment Algorithms of UAV Swarm for Sustainable Wireless Coverage. *IEEE Trans. Veh. Technol.* **2020**, *69*, 10320–10335. https://doi.org/10.1109/TVT.2020.3004855.
14. Citroni, R.; Di Paolo, F.; Livreri, P. A novel energy harvester for powering small UAVs: Performance analysis, model validation and flight results. *Sensors* **2019**, *19*, 1771–1791. https://doi.org/10.3390/s19081771.
15. Hatcher, W.G.; Yu, W. A survey of deep learning: Platforms, applications and emerging research trends. *IEEE Access* **2018**, *6*, 24411–24432. https://doi.org/10.1109/ACCESS.2018.2830661.
16. Sai, S.; Garg, A.; Jhawar, K.; Chamola, V.; Sikdar, B. A comprehensive survey on artificial intelligence for unmanned aerial vehicles. *IEEE Open J. Veh. Technol.* **2023**, *4*, 713–738. https://doi.org/10.1109/OJVT.2023.3316181.
17. Hayat, S.; Jung, R.; Hellwagner, H.; Bettstetter, C.; Emini, D.; Schnieders, D. Edge computing in 5G for drone navigation: What to offload? *IEEE Robot. Autom. Lett.* **2021**, *6*, 2571–2578. https://doi.org/10.1109/LRA.2021.3062319.
18. Tang, J.; Nie, J.; Zhang, Y.; Duan, Y.; Xiong, Z.; Niyato, D. Air-ground collaborative edge intelligence for future generation networks. *IEEE Netw.* **2023**, *37*, 118–125. https://doi.org/10.1109/MNET.008.2200287.
19. Ampatzidis, Y.; Partel, V.C.L.A. Agroview: Cloud-based application to process, analyze and visualize uav-collected data for precision agriculture applications utilizing artificial intelligence. *Comput. Electron. Agric.* **2020**, *174*, 105457. https://doi.org/10.1016/j.compag.2020.105457.
20. Luo, F.; Jiang, C.; Yu, S.; Wang, J.; Li, Y.; Ren, Y. Stability of cloud-based UAV systems supporting big data acquisition and processing. *IEEE Trans. Cloud Comput.* **2019**, *7*, 866–877. https://doi.org/10.1109/TCC.2017.2696529.
21. Qu, Y.; Sun, H.; Dong, C.; Kang, J.; Dai, H.; Wu, Q.; Guo, S. Elastic collaborative edge intelligence for UAV swarm: Architecture, challenges, and opportunities. *IEEE Commun. Mag.* **2024**, *62*, 62–68. https://doi.org/10.1109/MCOM.002.2300129.
22. Chen, X.; Bi, Y.; Han, G.; Zhang, D.; Liu, M.; Shi, H.; Zhao, H.; Li, F. Distributed computation offloading and trajectory optimization in multi-uav-enabled edge computing. *IEEE Internet Things J.* **2022**, *9*, 20096–20110. https://doi.org/10.1109/JIOT.2022.3175050.
23. Zhao, N.; Ye, Z.; Pei, Y.; Liang, Y.C.; Niyato, D. Multi-agent deep reinforcement learning for task offloading in uav-assisted mobile edge computing. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 6949–6960. https://doi.org/10.1109/TWC.2022.3153316.
24. Ren, W.; Qu, Y.; Qin, Z.; Dong, C.; Zhou, F.; Zhang, L.; Wu, Q. Efficient pipeline collaborative DNN inference in resource-constrained UAV swarm. In Proceedings of the 2024 IEEE Wireless Communications and Networking Conference (WCNC), Dubai, United Arab Emirates, 21–24 April 2024; pp. 1–6. https://doi.org/10.1109/WCNC57260.2024.10570535.
25. Jouhari, M.; Al-Ali, A.K.; Baccour, E.; Mohamed, A.; Erbad, A.; Guizani, M.; Hamdi, M. Distributed CNN inference on resource-constrained UAVs for surveillance systems: Design and optimization. *IEEE Internet Things J.* **2022**, *9*, 1227–1242. https://doi.org/10.1109/JIOT.2021.3079164.
26. Dhuheir, M.A.; Baccour, E.; Erbad, A.; Al-Obaidi, S.S.; Hamdi, M. Deep reinforcement learning for trajectory path planning and distributed inference in resource-constrained UAV swarms. *IEEE Internet Things J.* **2023**, *10*, 8185–8201. https://doi.org/10.1109/JIOT.2022.3231341.
27. Baccour, E.; Erbad, A.; Mohamed, A.; Hamdi, M.; Guizani, M. RL-PDNN: Reinforcement learning for privacy-aware distributed neural networks in IoT systems. *IEEE Access* **2021**, *9*, 54872–54887. https://doi.org/10.1109/ACCESS.2021.3070627.
28. Qin, Y.; Kishk, M.A.; Alouini, M.S. Drone charging stations deployment in rural areas for better wireless coverage: Challenges and solutions. *IEEE Internet Things Mag.* **2022**, *5*, 148–153. https://doi.org/10.1109/IOTM.001.2100083.

29. Bai, Y.; Zhao, H.; Zhang, X.; Chang, Z.; Jäntti, R.; Yang, K. Toward autonomous multi-UAV wireless network: A survey of reinforcement learning-based approaches. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 3038–3067. https://doi.org/10.1109/COMST.2023.3323344.
30. Feriani, A.; Hossain, E. Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 1226–1252. https://doi.org/10.1109/COMST.2021.3063822.
31. Luong, N.C.; Hoang, D.T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.C.; Kim, D.I. Applications of deep reinforcement learning in communications and networking: A survey. *Commun. Surv. Tutor.* **2019**, *21*, 3133–3174. https://doi.org/10.1109/COMST.2019.2916583.
32. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
33. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*; Pereira, F., Burges, C., Bottou, L., Weinberger, K., Eds.; Curran Associates, Inc.: New York, NY, USA, 2012; Volume 25.
34. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2015**, arXiv:1409.1556.
35. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. https://doi.org/10.1109/CVPR.2016.91.
36. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848.
37. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. https://doi.org/10.1007/s11263-009-0275-4.