

LETTER

Learning Fast Deployment for UAV-Assisted Disaster System

Na XING^{†a)}, Lu LI[†], Ye ZHANG[†], and Shiyi YANG[†], *Nonmembers*

SUMMARY Unmanned aerial vehicle (UAV)-assisted systems have attracted a lot of attention due to its high probability of line-of-sight (LoS) connections and flexible deployment. In this paper, we aim to minimize the upload time required for the UAV to collect information from the sensor nodes in disaster scenario, while optimizing the deployment position of UAV. In order to get the deployment solution quickly, a data-driven approach is proposed in which an optimization strategy acts as the expert. Considering that images could capture the spatial configurations well, we use a convolutional neural network (CNN) to learn how to place the UAV. In the end, the simulation results demonstrate the effectiveness and generalization of the proposed method. After training, our CNN can generate UAV configuration faster than the general optimization-based algorithm.

key words: *unmanned aerial vehicle, deployment, convolutional neural network, data collection*

1. Introduction

With the rapid development of 6G and sensor technology, many delay-sensitive and computation-expansive sensor applications are emerging in our life, such as wearable devices [1] etc. But the sensor nodes usually have limited communication and energy resources, which makes it difficult to support long distance transmission [2]. Then, how to collect information from sensor nodes becomes a problem, especially in the dangerous and disaster situations where the infrastructure has been damaged.

Fortunately, the UAV can be used to provide services as an aerial base station or a fusion center. UAV has the advantages of low cost, high flexibility, convenient deployment and no casualties. Nowadays, the UAV is widely used in military and civil fields [3]. It can flexibly navigate to the location of sensors and collect or forward information. In addition, the high probability of LoS transmission can also improve the communication efficiency. Anyway, UAV-assisted system is considered as an important technology in the disaster scenario such as earthquake or forest fire.

Of course, an optimization problem is produced about the UAV deployment position. Many works have investigated this problem under different systems, including UAV-aided mobile edge computing system, UAV-enabled flying network, UAV-assisted relay communication and so on. In these systems, the UAV acts as a base station, access point or

relay. Some optimization and learning methods are proposed to optimize the deployment position or flight trajectory. A common approach is to transform the UAV position optimization into a convex problem using the successive convex approximation (SCA) [4], [5]. However, for each sensor node configuration, the SCA-based method needs to be recalculated, which is time-consuming and unsuitable for online applications. On the other hand, most of the learning methods are based on deep reinforcement learning [6], [7], which requires high trial and error costs. Therefore, it is necessary to develop a method with fast solving ability to get the UAV deployment position to adapt to the urgency of disasters.

Inspired by the existing works, the main contributions of this paper are summarized as follows:

- We consider a UAV-assisted disaster scenario where some ground sensor nodes need to upload data to the UAV. The Rician fading channel model is used to describe the fading property between the UAV and sensor nodes. To minimize the upload time, the UAV deployment position is optimized.
- The formulated problem is non-convex that is difficult to solve. Traditional methods is time-consuming and become prohibitively slow for large systems. Images could capture the spatial configuration well. To this end, a supervised learning method based on the CNN is proposed, which learns to deploy the UAV from an expert. We use the optimization algorithm to generate the dataset.

2. System Model and Problem Formulation

We consider a UAV-assisted disaster system, which consists of M sensor nodes and a UAV. These sensor nodes are used for monitoring the information of environments, buildings, etc. The set of those sensor nodes is denoted as $\mathcal{M} = \{1, 2, \dots, M\}$. We assume that the deployment locations of the sensor nodes are known. After the disaster, a UAV is deployed to collect information from sensor nodes. The flying altitude of the UAV is assumed to be fixed and denoted as z . The location of sensor node $m \in \mathcal{M}$ and the horizontal coordinate of the UAV are denoted as $\mathbf{w}_m = [x_m^{\text{node}}, y_m^{\text{node}}]^T$ and $\mathbf{q} = [x^{\text{uav}}, y^{\text{uav}}]^T$. Hence, the distance between the sensor node m and the UAV can be calculated by $d_m = \sqrt{\|\mathbf{q} - \mathbf{w}_m\|^2 + z^2}$.

We assume the sensor nodes upload their data using time-division multiple access (TDMA) approach to prevent

Manuscript received November 17, 2023.

Manuscript revised February 5, 2024.

Manuscript publicized May 30, 2024.

[†]School of Information, North China University of Technology, Beijing 100144, China.

a) E-mail: xingna@ncut.edu.cn

DOI: 10.1587/transinf.2023EDL8082

co-channel interference. F (in bits) denote the amount of data that each sensor node needs to upload. The fixed transmission rate from the sensor node m to the UAV is represented by R_m . Therefore, the expected time to upload data can be expressed by $t_m = F/R_m$.

The conventional simplified LoS model is idealized and quite different from the practical channel. In comparison, the Rician angle-dependent fading channel model [8] provides a description that is more consistent with realistic scenarios. In this model, the received signal includes not only the direct transmission from the transmitter to the receiver, but also the signal transmission via reflection, refraction and scattering paths. The channel between node m and the UAV can be modelled as

$$h_m = \sqrt{\beta_m} g_m = \sqrt{\beta_0 d_m^{-\alpha}} g_m, \quad (1)$$

where β_m is the large-scale average channel power gain and g_m is the small-scale fading coefficient, β_0 is the average channel power gain at a reference distance $d_0 = 1$ m, α is the path loss exponent. The small-scale fading can be expressed by

$$g_m = \sqrt{\frac{k_m}{k_m + 1}} g + \sqrt{\frac{1}{k_m + 1}} \tilde{g}, \quad (2)$$

where g is the LoS component with $|g| = 1$, \tilde{g} is the random scattered component which is a zero-mean unit-variance circularly symmetric complex Gaussian random variable, k_m denotes the Rician factor which can be given by

$$k_m = K_1 \exp(K_2 \theta_m), \quad (3)$$

where K_1 and K_2 are constants determined by the specific environment, $\theta_m = \arcsin(z/d_m)$ is the elevation angle between the sensor node m and the UAV.

Therefore, the maximum achievable rate at the UAV from node m can be expressed as

$$C_m = B \log_2 \left(1 + \frac{|h_m|^2 p_m}{\sigma^2 \Gamma} \right), \quad (4)$$

where B denotes the bandwidth, p_m is the transmission power of node m , σ^2 is the noise power and $\Gamma > 1$ is the signal-to-noise ratio (SNR) gap.

The outage probability that the UAV cannot successfully receive data from node m can be given by

$$\begin{aligned} P_m &= \mathcal{P}(C_m < R_m) \\ &= \mathcal{P} \left(|g_m|^2 < \frac{\sigma^2 \Gamma (2^{R_m/B} - 1)}{\beta_0 d_m^{-\alpha} p_m} \right) \\ &= F_m \left(\frac{\sigma^2 \Gamma (2^{R_m/B} - 1)}{\beta_0 d_m^{-\alpha} p_m} \right), \end{aligned} \quad (5)$$

where F_m is the cumulative distribution function of $|g_m|^2$ which is a non-decreasing function w.r.t. R_m .

Let ε be the maximum tolerable outage probability with $0 < \varepsilon \leq 0.1$. To minimize the upload time and ensure the

data received by the UAV, R_m is chosen such that $P_m = \varepsilon$, $\forall m$. Therefore,

$$R_m = B \log_2 \left(1 + \frac{f_m p_m \beta_0}{\sigma^2 \Gamma d_m^\alpha} \right), \quad (6)$$

where f_m denotes the solution to $P_m = \varepsilon$ which can be approximated by the logistic model [8]

$$f_m \approx A_1 + \frac{A_2}{1 + \exp(-(A_3 + A_4 v_m))}, \quad (7)$$

where $v_m = z/d_m$, $A_1 > 0$ and $A_2 > 0$ with $A_1 + A_2 = 1$, $A_3 < 0$ reflects the positive logistic mid-point, $A_4 > 0$ is the logistic growth rate.

Based on the discussion above, the upload time is minimized by solving the following problem

$$\begin{aligned} (\mathbf{P1}) \quad & \min_{\mathbf{q}} \sum_{m=1}^M \frac{F}{R_m} \\ \text{s.t.} \quad & x^{\min} \leq x^{\text{uav}} \leq x^{\max}, \end{aligned} \quad (8a)$$

$$y^{\min} \leq y^{\text{uav}} \leq y^{\max}, \quad (8b)$$

$$v_m = \frac{z}{\sqrt{\|\mathbf{q} - \mathbf{w}_m\|^2 + z^2}}, \quad \forall m \in \mathcal{M}. \quad (8c)$$

Problem **P1** adjusts the deployment position of UAV \mathbf{q} in order to minimize the upload time. Here (8a) and (8b) are the allowed coordinates of the UAV. Note that the objective function is a non-convex function w.r.t. \mathbf{q} and v_m , which makes the Problem **P1** difficult to solve.

3. Problem Solution

Problem **P1** is usually converted into a convex problem and solved by an optimization approach. However, these optimization algorithms require long solution time and not suitable for online applications, especially in disaster scenarios. In addition, it's inexpensive to perform inference on a trained neural networks. Furthermore, images have the ability to encode the spatial information of nodes [9]. Therefore, we propose a data-driven approach that uses a CNN to learn UAV's position from an expert.

3.1 Dataset Generation

In our scheme, an optimization approach is used as the expert to minimize the upload time. First, the original problem is converted into a convex optimization problem, and then the CVX solver is used to solve it. Its detailed process is as follows.

By relaxing the constraint (8c) and defining a slack variable μ_m , Problem **P1** can be rewritten as

$$\begin{aligned} (\mathbf{P2}) \quad & \min_{\mathbf{q}, v_m, \mu_m} \sum_{m=1}^M \frac{F}{\mu_m} \\ \text{s.t.} \quad & (8a), (8b), \end{aligned} \quad (9a)$$

$$v_m \leq \frac{z}{\sqrt{\|\mathbf{q} - \mathbf{w}_m\|^2 + z^2}}, \forall m \in \mathcal{M}, \quad (9b)$$

$$\mu_m \leq R_m, \forall m \in \mathcal{M}. \quad (9c)$$

Lemma 1: The solution of Problem **P2** is equivalent to the solution of Problem **P1**.

Proof: Please refer to [2].

To our knowledge, the above Problem **P2** is often converted to a convex optimization problem by the SCA approach.

Lemma 2: According to the SCA approach, we get

$$\begin{aligned} R_m &\geq R_m^r - \lambda_{m,1}^r \left(e^{-(A_3+A_4v_m)} - e^{-(A_3+A_4v_m^r)} \right) \\ &\quad - \lambda_{m,2}^r \left(\|\mathbf{q} - \mathbf{w}_m\|^2 - \|\mathbf{q}^r - \mathbf{w}_m\|^2 \right) \\ &= R_m^{lb,r} \end{aligned} \quad (10)$$

where

$$\begin{aligned} \lambda_{m,1}^r &= \frac{A_2 B \gamma}{\eta_m^r (\eta_m^r (d_m^r)^\alpha + \gamma (A_1 \eta_m^r + A_2)) \ln 2}, \\ \lambda_{m,2}^r &= \frac{\alpha B \gamma (A_1 \eta_m^r + A_2)}{(d_m^r)^2 (\eta_m^r (d_m^r)^\alpha + \gamma A_1 \eta_m^r + \gamma A_2) \ln 4}, \end{aligned}$$

$\eta_m^r = 1 + e^{-(A_3+A_4v_m^r)}$ and $\gamma = \frac{\rho_m \beta_0}{\sigma^2 \Gamma}$. r represents the r th iteration.

Proof: Please refer to [2].

Similarly, for a given \mathbf{q}^r , there exists a lower bound $v_m^{lb,r}$ for v_m

$$\begin{aligned} \frac{z}{\sqrt{\|\mathbf{q} - \mathbf{w}_m\|^2 + z^2}} &\geq v_m^r - \frac{z}{2 (\|\mathbf{q}^r - \mathbf{w}_m\|^2 + z^2)^{3/2}} \\ &\quad \times \left(\|\mathbf{q} - \mathbf{w}_m\|^2 - \|\mathbf{q}^r - \mathbf{w}_m\|^2 \right) \\ &= v_m^{lb,r} \end{aligned} \quad (11)$$

Using Lemma 1 and Lemma 2, Problem **P2** can be translated into the following convex problem.

$$(\mathbf{P3}) \quad \min_{\mathbf{q}, v_m, \mu_m} \sum_{m=1}^M \frac{F}{\mu_m} \quad (12a)$$

$$\text{s.t.} \quad (8a), (8b),$$

$$v_m \leq v_m^{lb,r}, \forall m \in \mathcal{M}, \quad (12b)$$

$$\mu_m \leq R_m^{lb,r}, \forall m \in \mathcal{M}. \quad (12c)$$

This problem can be solved using the CVX solver.

We use the above optimization method as the expert to generate training samples consisting of pairs of images that capture the given sensor nodes and UAV configurations. Each image pixel represents a fixed metric distance \bar{d} . Figure 1 illustrates an example to show the entire sample generation process. Firstly, we sample the configurations of sensor nodes in a random distribution (Fig. 1 (a)). Next, Problem **P3** can be iteratively solved to obtain the optimal UAV configuration (Fig. 1 (c)). Then, in order to avoid sparsity issues, we use a Gaussian kernel to mark the configurations of sensor

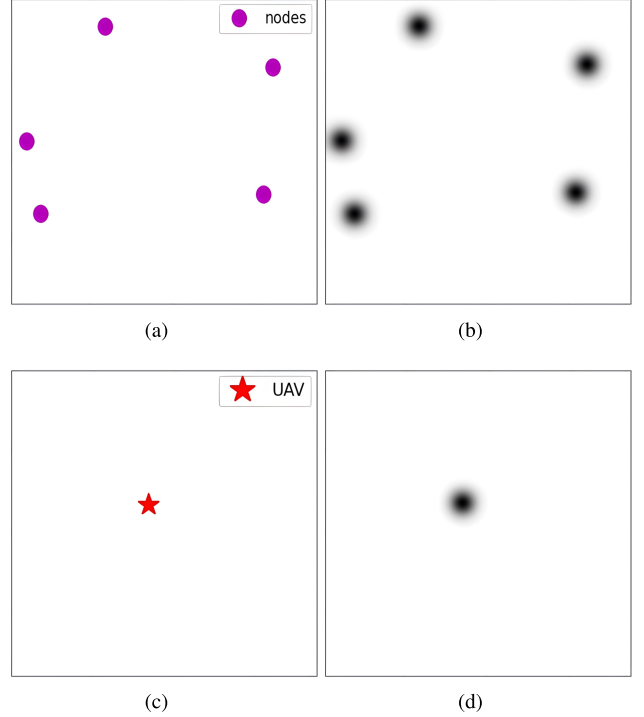


Fig. 1 Training sample generation process.

nodes and UAV separately (Fig. 1 (b) and Fig. 1 (d)), which constitute the input and output images of the CNN. By the above steps, we can obtain the dataset. Note that the samples are randomly generated to ensure the richness and diversity of our dataset.

3.2 Learning Architecture

Choosing an appropriate model is important to minimize the upload time. It may be surprising that we choose the CNN. The main reasons are as follows: Firstly, our problem depends on the relative positions between the sensor nodes and the UAV, rather than the absolute positions. A CNN captures this feature very well, which makes it more applicable to our problem. Secondly, images can represent spatial information and its performance will not degrade as the number of sensor nodes grows. For CNNs, its processing is the same, which makes no difference between 5 nodes or 10 nodes. However, traditional optimization methods need longer time for large-scale sensor nodes. Therefore, we don't use a learning model that inputs the position of sensor nodes and outputs the UAV position.

Considering that the input and output of our network are both an image, we adopt a learning model based on an autoencoder [9]. The learning model only consists of convolutional layers and convolutional transpose layers, which constitute the encoder and decoder, respectively, as shown in Fig. 2. The encoder processes the 768×768 image with the 8×8 kernel, the 4×4 kernel and the 2×2 kernel. Then, the image is transformed into a volume with dimension $3 \times 3 \times \tilde{F}$, where \tilde{F} is the number of filters. The decoder performs con-

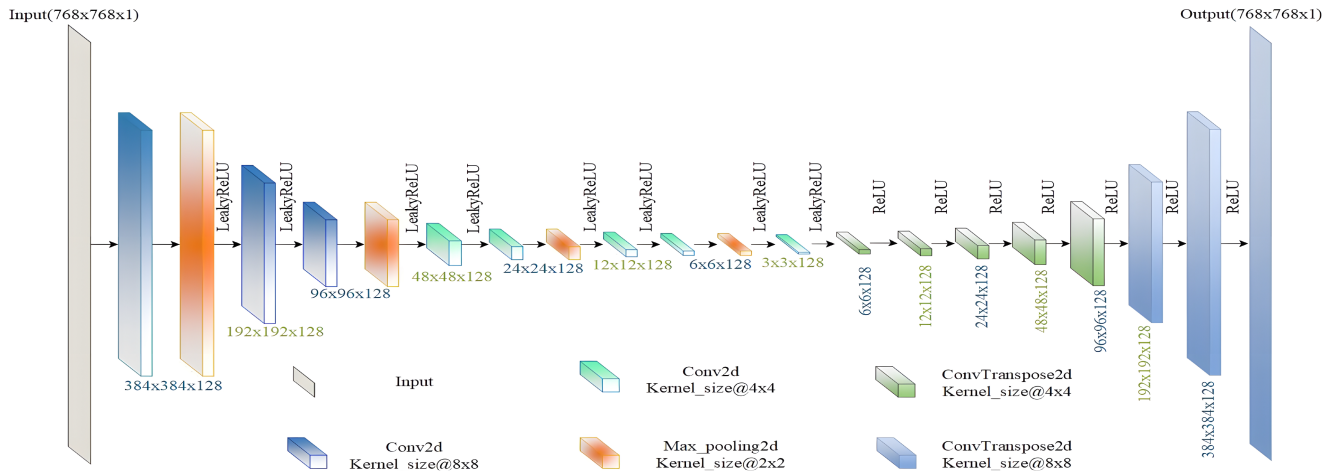


Fig. 2 Our CNN architecture.

volitional transpose. We do not use the fully connected layers because the convolutional layers can achieve better generalization performance. On the other hand, the auto-encoder with convolutional layers can handle images of arbitrary size. Our model is compatible with image resolution $64N + 768$, where N is an integer with $N \geq 0$. The deepest convolutional layer can extract the most salient spatial features from the input image, which inform the optimal position of the UAV in the output image.

4. Simulation Results

In this section, we verify the effectiveness and generalization of the proposed method. We consider the height of the UAV is set to be $z = 100$ m. The bandwidth is $B = 10$ MHz. The channel power gain at the reference distance $d_0 = 1$ m is $\beta_0 = -30$ dB. The transmit power of each sensor node is set to be 0.1 W and the noise power is $\sigma^2 = 10^{-9}$ W. The SNR gap is $\Gamma = 8.2$ dB. The parameters of the logistic model are set as $A_1 = 0.01$, $A_2 = 0.99$, $A_3 = -4.7$ and $A_4 = 8.9$ [2]. We train our CNN network model with 2–10 sensor nodes. Each image pixel represents 1.3 m. The Adam optimizer is used with a learning rate of 10^{-4} , a batch size of 8. The training processes are executed on GPU, NVIDIA GeForce RTX 1080. We compare our learning approach with the method of solving Problem P3 by CVX solver.

To verify the effectiveness of the optimization method, we randomly generate 20 sensor nodes and the initial positions of the UAV while keeping the parameters unchanged, and use our optimization method to iteratively find the final optimal position of the UAV. From Fig. 3(a), we can see that the UAV is gradually optimized from the initial position to the final optimal position through step-by-step iterations, demonstrating the effectiveness of the entire optimization process. Figure 3(b) shows that, the upload time decreases with each iteration. This demonstrate the effectiveness of the our method.

To evaluate the performance of the proposed method, we applied it to a test dataset with 2–10 sensor agents with

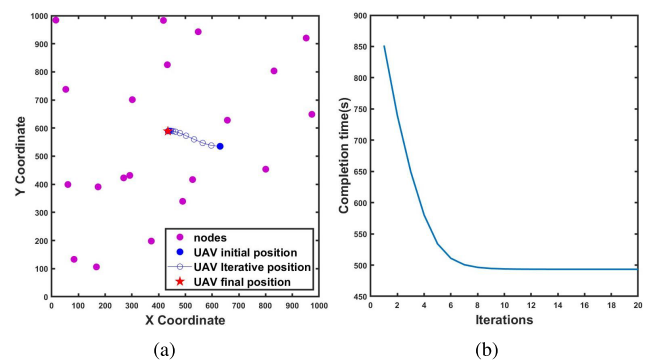


Fig. 3 (a) The UAV position obtained by our method. (b) Upload time convergence curve.

10000 samples. Our method performs well. For an intuitive representation, we take examples in which 5 and 8 sensor nodes are randomly configured, respectively. In addition to comparing with optimization algorithms, we consider the method with the sensor's center of gravity as UAV's position in Fig. 4. We can see that the UAV configuration produced by our proposed method is very similar to the optimization method. This is not surprising as we provide ample opportunity for the model to learn the UAV configuration. What's more, the UAV's position by our proposed method is different from the position of the sensor's center of gravity.

Furthermore, it is very time consuming to generate data and retrain the network for every size. We expect our network has learned something that would be applied to a larger scale. Therefore, we apply our trained network with 2–10 sensor nodes to test examples with 20 and 40 sensor nodes. The result is shown in Fig. 4. It can be seen that our model generalizes well.

The traditional optimization method becomes slow as the number of sensor nodes increases and are not suitable for real-time scenarios. However, the proposed method is exceptionally scalable. Figures 5(a) and (b) show the comparison of optimization time and upload time, which are based on Monte Carlo simulation. For the optimization time, we can

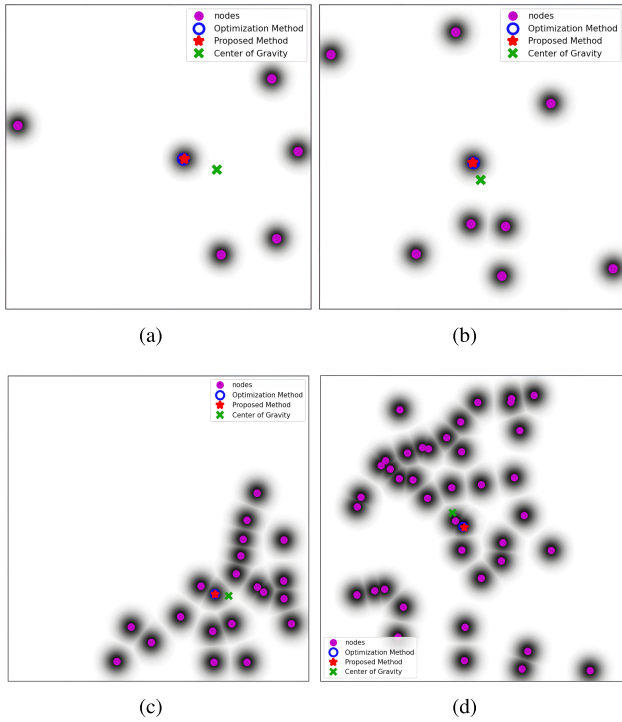


Fig. 4 Results for test samples with (a) 5 sensor nodes and (b) 8 sensor nodes and (c) 20 sensor nodes and (d) 40 sensor nodes.

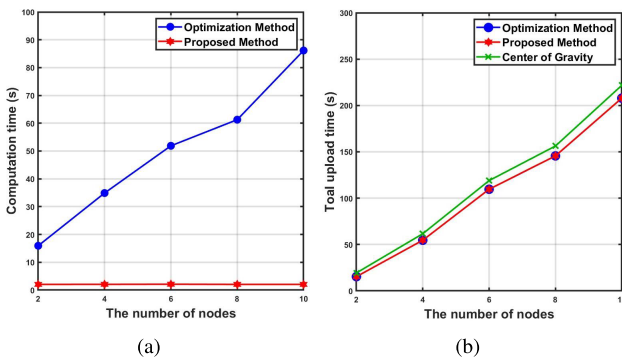


Fig. 5 (a) The optimization time required to obtain the UAV configuration result. (b) The upload time required for the UAV configuration.

see that the advantage of our method is clear. For 10 sensor nodes, our proposed method is nearly 40 times faster than the traditional method. This is not a trivial result. For the upload time, our proposed method is better than the position of sensor's center of gravity.

5. Conclusions

This paper aimed to minimize the upload time of the UAV-assisted disaster system. We proposed a supervised learning approach based on a convolutional neural network that

learned how to deploy UAV. It took advantage of the fact that images can capture spatial configurations well. While general optimization-based algorithms become slow as the scale of system increases, our approach runs faster and can quickly give the same results. The simulation results verified the effectiveness of the proposed method. For the vertical position optimization of UAV, or the association optimization brought by multiple UAVs, we could use a 3D CNN, which will be our future research work.

Acknowledgments

This work was supported in part by the Program of Beijing Municipal Education Commission (KM202310009001), in part by the Scientific Research Foundation of North China University of Technology, and in part by the National Key Research and Development Program of China under Grant 2020YFC0811004.

References

- [1] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, and I.D. Munari, "IoT wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment," *IEEE Internet Things J.*, vol.6, no.5, pp.8553–8562, Oct. 2019.
- [2] S. Sun, G. Zhang, H. Mei, K. Wang, and K. Yang, "Optimizing multi-UAV deployment in 3-D space to minimize task completion time in UAV-enabled mobile edge computing systems," *IEEE Commun. Lett.*, vol.25, no.2, pp.579–583, Feb. 2021.
- [3] N. Xing, Q. Zong, L. Dou, B. Tian, and Q. Wang, "A game theoretic approach for mobility prediction clustering in unmanned aerial vehicle networks," *IEEE Trans. Veh. Technol.*, vol.68, no.10, pp.9963–9973, Oct. 2019.
- [4] Y. Wu, W. Yang, X. Guan, and Q. Wu, "UAV-enabled relay communication under malicious jamming: Joint trajectory and transmit power optimization," *IEEE Trans. Veh. Technol.*, vol.70, no.8, pp.8275–8279, Aug. 2021.
- [5] N. Xing, Y. Wang, L. Teng, and L. Li, "Communication and trajectory design in UAV-enabled flying network," *Digital Signal Process.*, vol.126, Art. no.103489, June 2022.
- [6] R. Ding, Y. Xu, F. Gao, and X. Shen, "Trajectory design and access control for air-ground coordinated communications system with multi-agent deep reinforcement learning," *IEEE Internet Things J.*, vol.9, no.8, pp.5785–5798, April 2022.
- [7] W. Zhang, Q. Wang, X. Liu, Y. Liu, and Y. Chen, "Three-dimension trajectory design for multi-UAV wireless network with deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol.70, no.1, pp.600–612, Jan. 2021.
- [8] C. You and R. Zhang, "3D trajectory optimization in Rician fading for UAV-enabled data harvesting," *IEEE Trans. Wireless Commun.*, vol.18, no.6, pp.3192–3207, June 2019.
- [9] D. Mox, V. Kumar, and A. Ribeiro, "Learning connectivity-maximizing network configurations," *IEEE Robot. Automat. Lett.*, vol.7, no.2, pp.5552–5559, April 2022.