

Smart Drones, Smarter Learning: Federated Self-learning Minimal Learning Machine Classifier for Real-Time Hyperspectral Image Classification

Anna-Maria Raita-Hakola¹, Xinying Kilpi-Chen¹, Ilkka Pölönen¹

¹ Spectral Imaging Laboratory, Faculty of Information Technology, University of Jyväskylä, Finland. - anna.m.hakola@jyu.fi

Keywords: Self-learning, Federated learning, Distance-based machine-learning, hyperspectral image classification, UAV swarm, UAV trajectory optimization

Abstract

This paper presents a framework for real-time hyperspectral image classification using federated self-learning Minimal Learning Machines (SL-MLM) and trajectory-optimized UAV swarms. The proposed method enables on-board model training and prediction with low computational cost, supporting asynchronous collaboration between UAVs via adaptive Kalman filter-based model fusion. To optimize scanning efficiency, we integrate deep reinforcement learning-based trajectory planning using a Multi-Agent Deep Q-Network (MADQN), minimizing total flight duration and improving energy efficiency. Experimental results on the Salinas-A hyperspectral dataset demonstrate that our federated SL-MLM achieves high classification accuracy with minimal labeled data and communication overhead. The approach supports scalable and distributed remote sensing applications in bandwidth- and resource-constrained UAV environments.

1. Introduction

On-board computing enables real-time remote sensing across diverse applications such as agricultural management, forestry assessment, and disaster response. This capability allows for timely detection of hazards and more efficient interventions, replacing labor-intensive methods with automated, machine learning-based solutions. Furthermore, integrating hyperspectral imaging (HSI) with unmanned aerial vehicles (UAVs) extends monitoring capabilities beyond standard RGB imaging. Hyperspectral systems capture hundreds or thousands of spectral bands, providing detailed spectral signatures that significantly enhance tasks such as classification, regression, and object detection.

Despite the technological maturity and affordability of UAVs and HS imagers, operational limitations persist, particularly concerning energy constraints and computational resources on-board (Chen et al., 2023). Additionally, HS imagers generate large volumes of high-dimensional data, necessitating efficient computational and storage solutions. This paper introduces a novel on-board HS image classification framework involving multiple UAVs equipped with HS line scanners managed by a portable base station (BS) that orchestrates flight control and communication between UAVs and the BS. The proposed method employs a computationally efficient distance-based classifier, the Self-learning Minimal Learning Machine (SL-MLM), introduced in (Raita-Hakola and Pölönen, 2022), integrated within a federated learning (FL) framework to facilitate efficient data sharing and model collaboration among UAVs (Nguyen et al., 2021).

In this work we will concentrate on two related questions:

1. How to plan the optimal position of the portable BS, the scanning starting point of UAVs, and the optimized trajectory of UAVs to achieve energy efficiency and reduce scan duration?
2. How does a distance-based minimalistic self-learning model perform in a federated learning scenario with HSI?

In this study, we are demonstrating that the integration of trajectory optimization and hyperspectral image classification into a unified system for real-time on-board monitoring is both feasible and advantageous. Our findings indicate that, by employing a federated learning framework, updated model coefficients from UAVs can be efficiently transmitted to a BS, thereby reducing communication overhead and enabling near real-time model refinement. The merging of individual models is a key issue to be evaluated. Experimental comparisons suggest that dynamic, state-driven UAV trajectory adjustments provided by a Multi-Agent Deep Q-Network (MADQN) (Wang et al., 2019) can reduce flight distances and energy consumption. It should be noted that most results are in the testing phase and that actual real-life evaluations have not yet been completed.

2. Trajectory Planning Using Reinforcement Learning

The HSI task imposes constraints on UAV trajectory planning, including the field of view, ground sampling distance, flight altitude, imaging angles, and route design. An optimal trajectory ensures complete and efficient coverage of the target area while minimizing redundant scanning regions. In scenarios involving multiple UAVs, the area of interest must be partitioned among UAV agents, necessitating coordinated trajectories and distributed model aggregation.

To optimize the UAV trajectories for cooperative push-broom HSI scan, we employ a MADQN (Wang et al., 2019) method. This reinforcement learning (RL) approach simultaneously optimizes scan-area assignments and individual UAV trajectories. Unlike conventional optimization methods, MADQN enables UAVs to dynamically adjust their trajectories based on real-time state information, thus reducing total flight distances and energy consumption.

2.1 System Model and Problem Formulation

As shown in Fig. 1, we consider N UAVs jointly performing the HSI task to scan an agriculture field and then transmit their

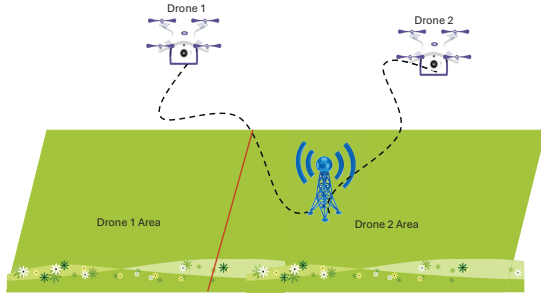


Figure 1. System model for the smart drones SL-MLM network.

trained model data to the base station (BS), where each UAV is assigned a specific area to carry out the scanning task and must depart from the common start point $O(x_s, y_s)$. The agricultural field is rectangular in dimensions $X \times Y$ m, and the BS is located at the same as the common start point O . To carry out this jointly scan task, the field is discretized into grid-based cells, each representing a scan unit with scanning width d . Assume that each UAV flies from O to its assigned area and then scans the area as push-broom method with fixed speed v and height H . We intend to optimize the common start point O of UAVs and their trajectory to achieve the minimum required scanning time for better energy efficiency.

Considering the rectangular field with dimension $X \times Y$ requires to be fully scanned using N UAVs, each equipped with a push-broom sensor of width d . The total number of vertical scanning strips can be given by:

$$T = \left\lfloor \frac{X}{d} \right\rfloor + 1. \quad (1)$$

Each UAV starts and ends its trajectory at the common start point O . The field is partitioned into N non-overlapping vertical strip sets, assigned to each UAV such that every portion of the field is covered exactly once.

Assume $\mathbf{p} = \{p_1, p_2, \dots, p_N\}$ to be a valid partition for each UAV of T strips, where p_i denotes the number of strips assigned to UAV- i . N UAVs require to push-broom scan the whole field area, which indicates the partition $\mathbf{p}$ satisfying:

$$\sum_{i=1}^N p_i = T \quad \forall i \in N. \quad (2)$$

The flight duration τ_i for UAV- i consists of: 1) the travel distance from the start point O to the first scan line; 2) the vertical scan motions in snake-like patterns and the lateral shifts between scan lines for the assigned area p_i ; 3) the return path from the final scan endpoint back to the common start point O . τ_i can be described as

$$\tau_i = \tau_i^{\text{go}} + \tau_i^{\text{scan}} + \tau_i^{\text{return}}. \quad (3)$$

Since UAV- i flies from O to its assigned area to perform a push-broom scan, we can reasonably consider that the trajectory of UAV- i is fixed once $\mathbf{p}$ and O are set. The optimization problem can be summarized to minimize the worst case of total flight duration by optimizing $\mathbf{p}$ and O as follows

$$\min_{(x_s, y_s), \mathbf{p}} \left\{ \max_{i \in \{1, \dots, N\}} \tau_i \right\} \quad (4a)$$

$$s.t. \quad \sum_{i=1}^N p_i = T \quad \forall i \in N. \quad (4b)$$

2.2 DQN for UAVs Trajectory Optimization

The objective of UAVs trajectory design is to minimize the maximum required task performing time among all drones. Each UAV's state, action, and reward function are defined as follows:

State: In our proposed model, the environment employs a simplified observation model, since the trajectory only depends on O and $\mathbf{p}$. The state space $\mathcal{S}$ is defined as static state space:

$$s \in \mathcal{S} = \{0.0\}. \quad (5)$$

This indicates that the environment is modeled as a single-step decision-making problem, where the learning process is driven entirely by the action-dependent reward.

Actions: The discrete action space $\mathcal{A}$ encodes both the common starting point O and the partitioning $\mathbf{p}$ of the field for multiple UAVs. Each action $a \in \mathcal{A}$ can be defined as:

- The discrete starting point $O(x_s, y_s)$ sampled from a uniform grid of resolution $G_x \times G_y$ over the $X \times Y$ m rectangular field.
- The partitioning vector $\mathbf{p} = [p_1, p_2, \dots, p_N]$, where $\sum_{i=1}^N p_i = T$.

The total action space can be derived as

$$\mathcal{A} = G_x \cdot G_y \cdot \mathcal{P}, \quad (6)$$

where $\mathcal{P}$ denotes the set of all valid integer partitions of T among N drones.

Reward: The reward $R(a)$ for an action a is computed based on the worst-case, i.e., the longest, trajectory among all UAVs, penalized by the spatial deviation of the common start point from the geometric center¹ of the field:

$$R(a) = - \max_{i \in \{1, \dots, N\}} \tau_i - \lambda \frac{\|(x_s, y_s) - (X/2, Y/2)\|}{\sqrt{(X/2)^2 + (Y/2)^2}}, \quad (7)$$

where λ is a weighting parameter controlling the penalty for non-centralized start positions. This design incentivizes a partition strategy that balances the scan workload among drones and selects a geometrically efficient start position to minimize the maximum completion time across all UAVs and provides better communication among UAVs and the BS.

Therefore, the completed optimization problem can be formulated as

¹ This is because the geometric center can provide better and more fair wireless communication with each agent.

$$\min_{(x_s, y_s), \mathbf{p}} \left\{ \max_{i \in \{1, \dots, N\}} \tau_i + \lambda \frac{\|(x_s, y_s) - (X/2, Y/2)\|}{\sqrt{(X/2)^2 + (Y/2)^2}} \right\} \quad (8a)$$

$$s.t. \quad \sum_{i=1}^N p_i = T \quad \forall i \in N, \quad (8b)$$

$$(x_s, y_s) \in \mathcal{G}, \quad (8c)$$

where $\mathcal{G}$ is the discrete grid of possible start points.

This formulation ensures full coverage of the field, balanced workload across UAVs, and operational efficiency through centralized coordination. The proposed DQN-based UAVs trajectory optimization scheme can be summarized in Algorithm 1.

Algorithm 1 DQN-Based Multi-Drone Scan Optimization

- 1: **Initialization:** Field size (X, Y) , scan width d , number of drones N , grid size (G_x, G_y)
- 2: Compute total number strips T of UAVs according to (1)
- 3: Generate all valid partitions $\mathcal{P}$
- 4: Construct discrete action space $\mathcal{A}$
- 5: **for** each training step **do**
- 6: Randomly select action $a = (x_s, y_s, \mathbf{p})$
- 7: Decode action into: Common start point O and strip assignment $\mathbf{p}$
- 8: Calculate flight time τ_i of each UAV- i according to (3)
- 9: Calculate reward $R(a)$ according to (6)
- 10: Update DQN agent with $R(a)$
- 11: **end for**
- 12: **return** Optimal common start point O^* and partition $\mathbf{p}^*$

3. Federated Self-Learning Minimal Learning Machine

The MLM is a computationally efficient, distance-based classifier used for rapid inference and training in classification and anomaly detection tasks (Pölönen et al., 2020, Raita-Hakola and Pölönen, 2021, de Souza Junior et al., 2015). MLM relates distance matrices of the data and reference points, which are subset of data points, through linear regression solved using Ordinary Least Squares (OLS) (Mesquita et al., 2017). Given spectral observations X , reference subset R , and their corresponding labels (Y, T) , MLM constructs the linear model:

$$\Delta_y = D_x \mathbf{B} + \mathbf{E}, \quad (9)$$

where D_y represents distance matrix between reference point label and data labels. Matrix Δ_v represent distance matrices between reference points and data. Coefficients $\hat{\mathbf{B}}$ are estimated via OLS:

$$\hat{\mathbf{B}} = (D_x^T D_x)^{-1} D_x^T \Delta_y. \quad (10)$$

For the new spectrum $\mathbf{x}_n$ the distance between its label y_n and set of reference points labels is

$$\delta(y_n, T) = d(\mathbf{x}_n, R) \hat{\mathbf{B}}. \quad (11)$$

MLM estimates its label y_n by solving a quadratic multilateration optimization problem detailed in (de Souza Junior et al., 2015). MLM also generalizes nearest neighbour classification (NN-MLM) through ranked distance assignment (Hakola and Pölönen, 2020, Mesquita et al., 2017), which we utilized here.

To further facilitate continuous learning, Self-Learning MLM (SL-MLM) is build on Recursive Least Squares (RLS) shown

as algorithm 2. RLS efficiently updates model coefficients in real time without reprocessing historical data (Romberg, 2016, Haykin, 2008). Implementation details SL-MLM are described in (Raita-Hakola and Pölönen, 2022).

Algorithm 2 Recursive Least Squares (Romberg, 2016)

- 1: **Input:** X, R, Y, T
- 2: **Initialize:**
- 3: Calculate distance matrix $D_{x,0}$
- 4: Calculate distance matrix $\Delta_{y,0}$
- 5: Calculate $\mathbf{P}_0 = (D_{x,0}^T D_{x,0})^{-1}$
- 6: Calculate model $\hat{\mathbf{B}}_{x,0} = \mathbf{P}_0 D_{x,0}^T \Delta_{y,0}$
- 7: **for** $i = 1, 2, 3, \dots$ **do**
- 8: *New data X_i, Y_i appears*
- 9: Calculate distance matrix $D_{x,i}$
- 10: Calculate distance matrix $\Delta_{y,i}$
- 11: Calculate $\mathbf{P}_{x,i} = \mathbf{P}_{i-1} D_{x,i}^T$
- 12: Calculate $\mathbf{P}_i = \mathbf{P}_{i-1} - \mathbf{P}_{x,i} (\mathbf{I} + D_{x,i} \mathbf{P}_{x,i})^{-1} D_{x,i} \mathbf{P}_{i-1}$
- 13: Calculate $\mathbf{K}_i = \mathbf{P}_i D_{x,i}^T$
- 14: Update model $\hat{\mathbf{B}}_{x,i} = \hat{\mathbf{B}}_{x,i-1} + \mathbf{K}_i (\Delta_{y,i} - D_{x,i} \hat{\mathbf{B}}_{x,i-1})$
- 15: **end for**
- 16: **return** $\hat{\mathbf{B}}_{x,i}$ and $\mathbf{P}_i$

To improve the quality of the training data, we integrated an anomaly detection mechanism directly into the learning algorithm (Raita-Hakola and Pölönen, 2021). Specifically, we used lower and upper thresholds on the variance of the distances between predicted labels and reference classes, denoted as

$$\text{Var}(\delta(y_n, T)). \quad (12)$$

By applying this variance-based filter, only high-quality and consistent samples were admitted into the recursive least squares update, thereby enhancing the robustness of the learning process.

In our federated learning approach, each UAV independently updates its local MLM model using onboard, communicating only updated coefficient matrices $\hat{\mathbf{B}}$ between UAVs and the BS. This significantly reduces communication bandwidth, enabling near real-time model updates.

To fuse independently learned models from multiple UAVs into a coherent and continuously updating base model, we employ an adaptive Kalman filter framework (Mehra, 1970). Each UAV produces a regression model estimate $\hat{\mathbf{B}}_i$ and a corresponding uncertainty covariance matrix $\mathbf{P}_i$, which are treated as noisy observations of the true underlying model. The fusion process is implemented through recursive Kalman updates, ensuring principled integration of new model information while maintaining robustness against overconfidence and filter divergence.

At each timestep i , the filtering process comprises a prediction and update step. In the prediction step, the current fused estimate $\mathbf{B}_{\text{fuse}}^{(i-1)}$ and its covariance $\mathbf{P}_{\text{fuse}}^{(i-1)}$ are propagated forward by adding a process noise term $\mathbf{Q}^{(i)}$, initially set proportional to the identity matrix and later adapted. To maintain numerical stability and avoid singularities in the covariance matrices, a small regularization term $\epsilon \mathbf{I}$ is added.

In the update step, the new UAV model estimate $\hat{\mathbf{B}}_i$ is incorporated using the classical Kalman gain

$$\mathbf{K}_i = \mathbf{P}_{\text{pred}}^{(i)} \left(\mathbf{P}_{\text{pred}}^{(i)} + \mathbf{P}_i \right)^{-1}. \quad (13)$$

The fused model is then updated as

$$\mathbf{B}_{\text{fuse}}^{(i)} = \mathbf{B}_{\text{pred}}^{(i)} + \mathbf{K}_i \left(\hat{\mathbf{B}}_i - \mathbf{B}_{\text{pred}}^{(i)} \right), \quad (14)$$

and the covariance is updated accordingly:

$$\mathbf{P}_{\text{fuse}}^{(i)} = (\mathbf{I} - \mathbf{K}_i) \mathbf{P}_{\text{pred}}^{(i)}. \quad (15)$$

To ensure the filter remains responsive over time and does not become overly confident, we use an adaptive process noise scaling mechanism. Specifically, we monitor the Frobenius norm $\|\mathbf{K}_i\|_F$ of the Kalman gain at each step and adjust the process noise as:

$$\mathbf{Q}^{(i+1)} = \|\mathbf{K}_i\|_F \cdot \mathbf{I}. \quad (16)$$

This formulation ties the system's responsiveness to the informational content of each incoming UAV model. Additionally, we track the log-determinant $\log \det(\mathbf{P}_{\text{fuse}})$ over time to monitor filter confidence and detect potential degeneracy.

This approach enables asynchronous, uncertainty-aware fusion of model parameters from distributed sources and supports continual learning in real-time remote sensing scenarios.

Algorithm 3 Kalman Filter Fusion for UAV Model Aggregation

```

1: Input: Initial model  $\hat{\mathbf{B}}_0$ , covariance  $\mathbf{P}_0$ , reference samples  $\mathbf{R}$ , reference labels  $\mathbf{y}_R$ 
2: Hyperparameters: Regularization  $\epsilon$ , dynamic process noise  $\mathbf{Q}$ 
3: Initialize:  $\mathbf{B}_{\text{fuse}} \leftarrow \hat{\mathbf{B}}_0$ ,  $\mathbf{P}_{\text{fuse}} \leftarrow \mathbf{P}_0$ 
4: for each timestep  $i = 1, 2, \dots$  do
5:   New UAV model  $\hat{\mathbf{B}}_i$ , covariance  $\mathbf{P}_i$  arrives
6:   Prediction step:
7:    $\mathbf{B}_{\text{pred}} \leftarrow \mathbf{B}_{\text{fuse}}$ 
8:    $\mathbf{P}_{\text{pred}} \leftarrow \mathbf{P}_{\text{fuse}} + \mathbf{Q}$ 
9:    $\mathbf{P}_{\text{pred}} \leftarrow \mathbf{P}_{\text{pred}} + \epsilon \cdot \mathbf{I}$  *Regularization
10:  Update step:
11:   $\mathbf{K}_i \leftarrow \mathbf{P}_{\text{pred}} (\mathbf{P}_{\text{pred}} + \mathbf{P}_i)^{-1}$  *Kalman gain
12:   $\mathbf{B}_{\text{fuse}} \leftarrow \mathbf{B}_{\text{pred}} + \mathbf{K}_i (\hat{\mathbf{B}}_i - \mathbf{B}_{\text{pred}})$ 
13:   $\mathbf{P}_{\text{fuse}} \leftarrow (\mathbf{I} - \mathbf{K}_i) \mathbf{P}_{\text{pred}}$ 
14:  Adaptive noise tuning:
15:   $g_i \leftarrow \|\mathbf{K}_i\|_F$  *Gain norm
16:   $\mathbf{Q} \leftarrow g_i \cdot \mathbf{I}$ 
17:  Track  $\log \det(\mathbf{P}_{\text{fuse}})$ 
18: end for
19: return  $\mathbf{P}_{\text{fuse}}$  and  $\mathbf{B}_{\text{fuse}}$ 

```

4. Simulation and Results

4.1 Trajectory planning

We provide simulation results to demonstrate the effectiveness of our DRL-based UAVs trajectory optimization scheme. In the simulation, assume that the field to be scanned has the dimension of 318 m and the scan width of the HSI camera on each UAV is 3.7 m. The uniform grid resolution is set to

$G_x = G_y = 5$ m. Due to the fact that UAVs need to transmit their trained model to the BS, a centralized BS can achieve better and more fair transmission performance. Therefore, we set the penalty weighting parameter $\lambda = 10$ to increase the weight of centralized common start positions.

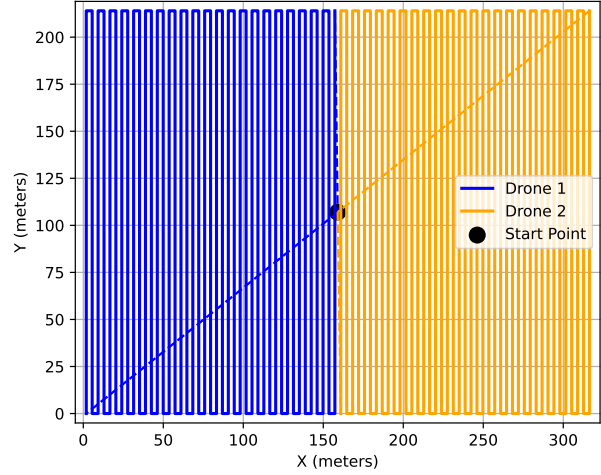


Figure 2. Optimized trajectories of two UAVs and the corresponding common start point.

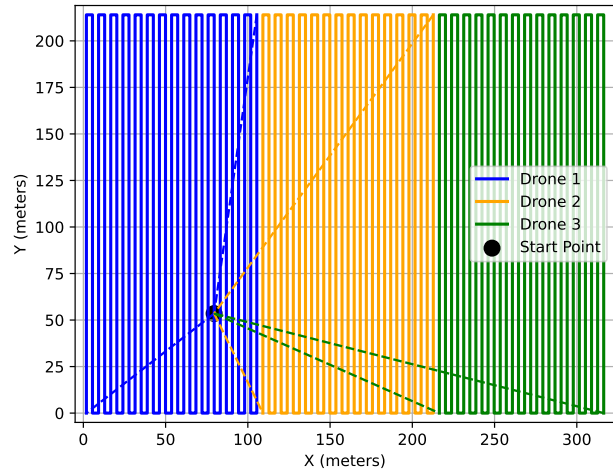


Figure 3. Optimized trajectories of three UAVs and the corresponding common start point.

In Fig. 2, we optimize the common starting point location O , along with the scan area assignment, for the case of $N = 2$ UAVs. The results suggest that the optimal location is at $O^*(159.0, 107.0)$, which lies approximately at the center of the field, seemingly an intuitive outcome. Each UAV is assigned 43 scanning strips, resulting in total flight lengths of $[9654.53, 9654.70]$ m. Thus, the overall network scanning time is $9654.70/v$, where v denotes the UAV scanning speed.

In Fig. 3, for the scenario with three UAVs, the optimized starting point is at $O^* = (79.5, 53.5)$, which is no longer at the center of the field. The scan strip allocation is $[29, 29, 28]$, and the corresponding flight lengths are $[6566.48, 6579.37, 6481.75]$ m. Consequently, the total network scanning time is $6579.3/v$.

4.2 Federated Self-learning MLM

To evaluate the proposed model fusion method based on adaptive Kalman filtering, we conducted experiments using the widely adopted Salinas-A hyperspectral dataset. This dataset contains hyperspectral imagery acquired over agricultural fields in the Salinas Valley, California, comprising 204 spectral bands and eight ground truth classes representing different crop types.

The original image was divided into training and test sets, both preserving the spatial and spectral structure of the scene. As a preprocessing step, ground truth labels were re-indexed to ensure a continuous class index starting from zero. Background pixels labeled as zero were excluded from training and evaluation.

To simulate a distributed sensing scenario, the training image was spatially split column-wise into two halves, mimicking two UAVs observing different regions of the field. A small representative reference set R and corresponding labels T were selected, with five samples per class used as reference points for initializing the model.

The model was first initialized using this reference data, then updated row by row in a simulated real-time fashion, where each UAV sequentially provided local observations. Model updates were performed using a recursive least squares scheme [2], and model fusion was achieved using an adaptive Kalman filter [3]. The Kalman gain and the determinant of the covariance matrix were monitored to assess confidence in the fused model. Classification was based on five nearest neighbors.

Anomaly detection thresholds were optimized by performing a grid search over lower and upper bounds for the variance of label distances. This tuning was conducted using only the training data. For each threshold pair, a simulation was run and classification accuracy recorded. The best-performing thresholds, yielding the highest accuracy, were selected to filter out unreliable samples during model updates. The selected boundaries were $[0.95, 1.95]$.

This setup provides a realistic simulation of how local models from two UAVs, observing different regions of a hyperspectral scene, can be effectively fused into a coherent global model under strict supervision constraints. Whole simulation can be seen in the algorithm 4.

Prediction performance was evaluated on the unseen test set after each update step. In Figure 4, the classification result on the test set after the final update is shown on the left, while the ground truth of the Salinas-A dataset is shown on the right. From Figures 5 and 6, we observe that although both UAVs were initialized with the same reference sets R and T , their learning performances differ. After fusing the models at the base station, the performance trends of the UAVs and the base model converge, achieving consistently high accuracy and F1-scores.

The effect of model updates and fusion can also be observed in the model weights. As shown in Figure 7, each line represents the evolution of a single weight in the model matrix B over time. Small but consistent changes after each update indicate that the model continues to adapt throughout the simulation.

Algorithm 4 Simulation of Federated UAV Learning with Adaptive Kalman Filter

```

1: Input: Uav's data,  $R$  and  $T$ 
2: Initialize model parameters  $\hat{B}_1, P_1, \hat{B}_2, P_2 \leftarrow (R, y_R)$ 
3: Initialize fused model  $\hat{B}_f, P_f, Q \leftarrow (\hat{B}_1, P_1, Q)$ 
4: for each row  $r$  in training data do
5:   Extract row  $x_{r1}, x_{r2}$  from UAV1 and UAV2
6:    $\hat{y}_{r1}, s_{r1} \leftarrow \text{predict\_model}(\hat{B}_1, x_{r1}, R, y_R)$ 
7:    $\hat{y}_{r2}, s_{r2} \leftarrow \text{predict\_model}(\hat{B}_2, x_{r2}, R, y_R)$ 
8:   Select inliers using anomaly score:
        $mask_1 \leftarrow s_{r1} \in (\tau_b, \tau_u)$ 
9:   if  $mask_1$  has any true values then
10:     $x'_{r1}, y'_{r1} \leftarrow$  masked inputs and predictions
11:    Compute distances:
12:     $D_{r1} \leftarrow \text{euclidean\_distances}(x'_{r1}, R)$ 
13:     $Y_{\delta 1} \leftarrow \text{euclidean\_distances}(y'_{r1}, y_R)$ 
14:     $\hat{B}_1, P_1 \leftarrow \text{recursive\_ls}(P_1, D_{r1}^T, Y_{\delta 1}^T, \hat{B}_1)$ 
15:   end if
16:   if  $mask_2$  has any true values then
17:     repeat lines 10–14 for UAV2
18:   end if
19:   if  $mask_1$  has any true values then
20:     $\hat{B}^-, P^- \leftarrow \text{kalman\_predict}(\hat{B}_f, P_f, Q)$ 
21:     $\hat{B}^+, P^+, K \leftarrow \text{kalman\_fuse}(\hat{B}^-, P^-, \hat{B}_1, P_1, \dots)$ 
22:    Update Q:  $Q \leftarrow \|K\| \cdot I$ 
23:     $\hat{B}_f, P_f \leftarrow \hat{B}^+, P^+$ 
24:    Synchronize UAVs to fused model
25:   end if
26:   if  $mask_2$  has any true values then
27:     repeat lines 20–24 for UAV2
28:   end if
29: end for

```

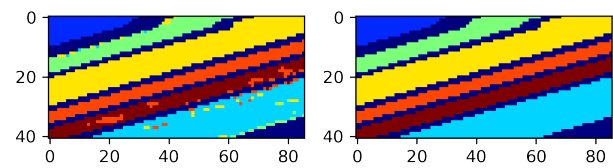


Figure 4. Classification results on the test set after the final update step (left) compared to the ground truth map (right) from the Salinas-A dataset.

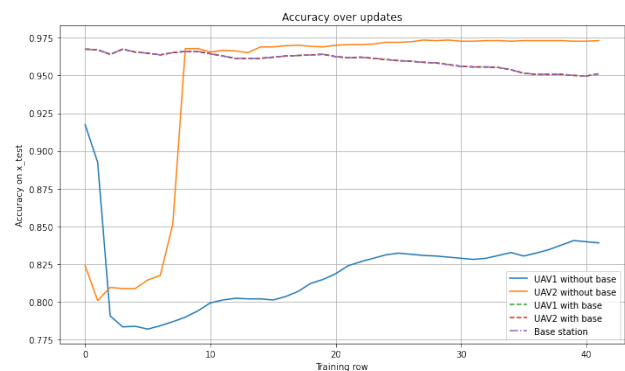


Figure 5. Accuracy of different models over update steps.

5. Discussion

This study investigated two key research questions: (1) How can the positions of a portable base station (BS), the initial scanning locations of UAVs, and their trajectories be optimized to achieve energy-efficient and time-effective hyperspectral imaging? (2) How does a distance-based, minimalistic self-learning

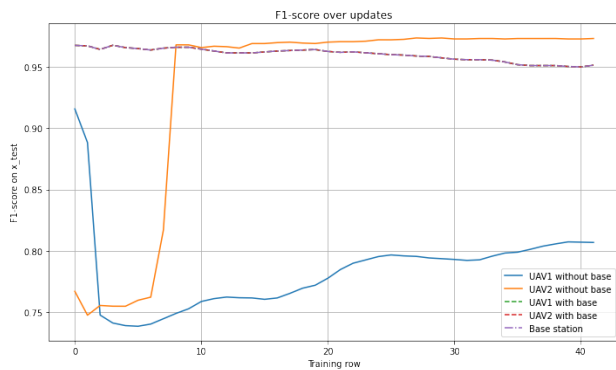


Figure 6. F1-score of different models over update steps.

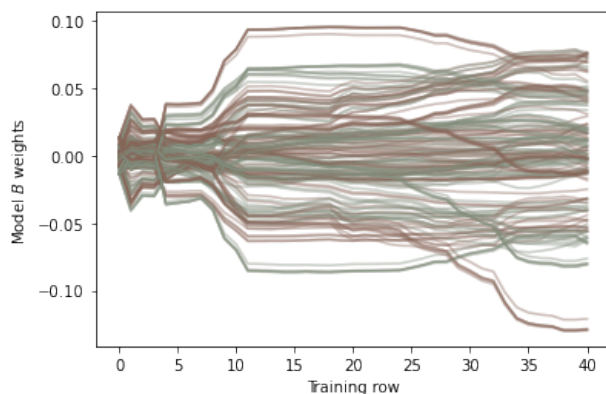


Figure 7. Evolution of individual model weights in the matrix B over successive update and fusion rounds. Each line corresponds to one model coefficient.

model perform in a federated learning framework applied to hyperspectral image (HSI) classification?

To address the first question, a reinforcement learning-based trajectory optimization approach was implemented using a Multi-Agent Deep Q-Network (MADQN). The UAVs' flight paths and initial positions were optimized with respect to a reward function that penalized excessive scan time and distance from the central BS. The results demonstrated that the MADQN-based planning strategy effectively minimized mission duration and energy expenditure, especially in scenarios with multiple UAVs. Notably, while a two-UAV setup resulted in symmetric path allocation around the field center, three-UAV configurations exhibited more asymmetric, yet optimal, trajectory patterns. These outcomes underscore the potential of deep reinforcement learning for real-time, adaptive planning in collaborative UAV missions.

Regarding the second question, we evaluated the proposed federated self-learning Minimal Learning Machine (SL-MLM) using the widely adopted Salinas-A dataset. This dataset, while relatively homogeneous and structured, serves as a suitable benchmark for validating initial feasibility. Each UAV was assigned a disjoint spatial partition of the training image and operated independently, updating its model using recursive least squares (RLS) with anomaly-aware sample selection. An adaptive Kalman filter was employed to fuse the individual UAV models at a centralized base station, accounting for estimation uncertainty via dynamic adjustment of process noise and tracking of covariance determinants.

Despite the limited complexity of the dataset, the experimental results revealed that the SL-MLM framework required only five labeled samples per class for initialization. This demonstrates the system's potential in field-deployable settings where annotated data is scarce. The communication footprint was minimal, consisting solely of model parameters (B) and associated covariance matrices (P), which is advantageous in bandwidth-constrained UAV networks.

Although the proposed fusion approach based on adaptive Kalman filtering was effective, alternative strategies, such as ensemble learning or Bayesian aggregation, may further enhance robustness and should be examined in future work. Additionally, early-stage discrepancies in model performance between UAVs point to the need for coordination mechanisms, such as cross-validation or confidence-based weighting during fusion.

The selection of hyperparameters, including the size of the reference set, number of nearest neighbors, and anomaly detection thresholds, was performed manually. Automating these through meta-learning or optimization-based tuning would increase adaptability to diverse mission contexts.

Lastly, while trajectory planning and model learning were evaluated separately in this work, their integration offers a compelling direction for future research. Coupling path planning with real-time model uncertainty could enable UAVs to actively seek informative or under-sampled regions, leading to more efficient and accurate scene understanding.

Future work will further enhance the framework by integrating real-time trajectory planning into the federated learning scheme using advanced reinforcement learning methods. In addition, the transition from conventional computing platforms to compact, low-power devices, such as Raspberry Pi computers, will be investigated to promote scalability and operational efficiency in diverse remote sensing applications.

6. Conclusion

We introduced a real-time hyperspectral image classification system combining federated self-learning with optimized UAV trajectory planning. The SL-MLM classifier enables lightweight on-board learning, while adaptive Kalman fusion effectively integrates distributed models under communication constraints. MADQN-based trajectory optimization further reduces mission time and energy usage. Results on benchmark data highlight the system's effectiveness and practicality, even with minimal supervision. Future work will focus on dynamic hyperparameter tuning, joint optimization of learning and path planning, and validation in more complex natural environments.

References

- Chen, X., Zhao, N., Chang, Z., Hämäläinen, T., Wang, X., 2023. UAV-Aided Secure Short-Packet Data Collection and Transmission. *IEEE Transactions on Communications*, 71(4), 2475-2486.
- de Souza Junior, A. H., Corona, F., Barreto, G. A., Miche, Y., Lendasse, A., 2015. Minimal Learning Machine: A novel supervised distance-based approach for regression and classification. *Neurocomputing*, 164, 34-44. <https://www.sciencedirect.com/science/article/pii/S0925231215003021>.

Hakola, A.-M., Pölönen, I., 2020. Minimal learning machine in hyperspectral imaging classification. L. B. Santi, F. Bovolo, Emanuele (eds), *Image and Signal Processing for Remote Sensing XXVI*, 11533, 26.

Haykin, S. S., 2008. *Adaptive filter theory*. Pearson Education India.

Mehra, R., 1970. On the identification of variances and adaptive Kalman filtering. *IEEE Transactions on Automatic Control*, 15(2), 175–184.

Mesquita, D. P., Gomes, J. P., Souza Junior, A. H., 2017. Ensemble of Efficient Minimal Learning Machines for Classification and Regression. *Neural Processing Letters*, 46(3), 751–766.

Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., Vincent Poor, H., 2021. Federated Learning for Internet of Things: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 23(3), 1622–1658.

Pölönen, I., Riihiho, K., Hakola, A.-M., Annala, L., 2020. Minimal learning machine in anomaly detection from hyperspectral images. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 43(B3), 467–472.

Raita-Hakola, A.-M., Pölönen, I., 2021. Piecewise anomaly detection using minimal learning machine for hyperspectral images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-3-2021, 89–96. <https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-3-2021/89/2021/>.

Raita-Hakola, A.-M., Pölönen, I., 2022. Updating strategies for distance based classification model with recursive least squares. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3, 163–170.

Romberg, J., 2016. Ece 6250 fall 2016 lecture notes. Lecture notes available from Georgia Tech University.

Wang, Y., Liu, H., Zheng, W., Xia, Y., Li, Y., Chen, P., Guo, K., Xie, H., 2019. Multi-objective workflow scheduling with deep-Q-network-based multi-agent reinforcement learning. *IEEE access*, 7, 39974–39982.