JOURNAL OF AEROSPACE INFORMATION SYSTEMS
Vol. 21, No. 7, July 2024

Development of Real-Time Unmanned Aerial Vehicle Urban Object Detection System with Federated Learning

You-Ru Lu* and Dengfeng Sun†
Purdue University, West Lafayette, Indiana 47906

https://doi.org/10.2514/1.I011378

In this paper, an urban object detection system via unmanned aerial vehicles (UAVs) is developed to collect real-time traffic information, which can be further utilized in many applications such as traffic monitoring and urban traffic management. The system includes an object detection algorithm, deep learning model training, and deployment on a real UAV. For the object detection algorithm, the Mobilenet-SSD model is applied owing to its lightweight and efficiency, which make it suitable for real-time applications on an onboard microprocessor. For model training, federated learning (FL) is used to protect privacy and increase efficiency with parallel computing. Last, the FL-trained object detection model is deployed on a real UAV for real-time performance testing. The experimental results show that the object detection algorithm can reach a speed of 18 frames per second with good detection performance, which shows the real-time computation ability of a resource-limited edge device and also validates the effectiveness of the developed system.

I. Introduction

IN THE last decade, unmanned aerial vehicles (UAVs) have been widely applied in many real-time applications, such as package delivery [1–3], agriculture [4,5], and wind condition monitoring [6–8]. Among a variety of engineering applications, one of the most important application fields is traffic monitoring and analysis [9,10]. By applying UAVs, the traffic flow information can be collected, which is crucial in urban traffic management. The information can be further applied to traffic flow prediction, which is considered a key element for the development of Intelligent Transportation System [11]. Hence, it is crucial to build a monitoring and information-collecting system for urban object detection with UAVs.

In recent years, deep learning (DL) and convolutional neural networks (CNNs) have shown great success in computer vision applications such as image recognition and object detection. Compared with traditional object detection methods such as the Viola–Jones algorithm [12], which utilize primitive features such as corners and edges, CNNs can extract both low-level features (e.g., corners and edges) and high-level features (e.g., eyes, ears, wheels) on different scales, making them more suitable for complex environments such as urban areas [13]. However, to build the UAV traffic monitoring system using DL and CNNs, there are still several challenges that need to be addressed. First, deploying DL algorithms on a UAV requires a powerful microprocessor to reach real-time performance. Second, to train CNNs with traditional centralized learning on a server requires drones transmitting collected data, resulting in high communication costs and privacy concerns. Therefore, in this work, the object detection system is developed by utilizing federated learning to solve the issues of centralized learning, and the Mobilenet-SSD object detection model is used to provide efficient object detection. The model trained by the FL framework is deployed on the onboard microprocessor of a real UAV, and its performance is validated by real UAV image data. The experimental results suggest that the detection speed is able to reach 18 frames per second (FPS), which verifies the real-time operation capability of the developed urban object detection system.

Received 11 October 2023; revision received 7 February 2024; accepted for publication 13 March 2024; published online 23 April 2024.

*Graduate Research Assistant, School of Aeronautics and Astronautics; lu799@purdue.edu.

†Professor, School of Aeronautics and Astronautics; dsun@purdue.edu.

A. Related Works

1. CNN-Based Object Detection

Object detection aims to find instances of objects from known classes in an image. The latest state-of-the-art techniques rely on deep CNNs, which are widely used in image processing tasks such as detection, recognition, and segmentation. Numerous object detectors have been proposed by the DL community, including Faster Regional-Convolutional Neural Network (R-CNN) [13], You Only Look Once (YOLO) [14], and Single-Shot Detector (SSD) [15]. The CNN-based object detectors can be categorized into two classes: 1) two-stage detectors, which find potential object locations by region-proposal network (RPN) in the first stage and classify objects within the regions by another classifier module in the second stage, and 2) single-shot detectors, which utilize only one CNN architecture to perform end-to-end object detection.

1) *Two-stage detectors:* Two-stage detectors separate the prediction process into two consecutive steps, which are object localization and classification. For instance, in Faster R-CNN [13], the RPN module is applied in the first stage to extract feature maps and identify regions of interest (box proposals) from the input image. The box proposals are locations that potentially contain target objects. In the second stage, the box proposals are used to crop features and pass them through a Detection Head, which consists of a classifier module in order to predict class probability and a localization module that refines a specific bounding box in the proposal. With two separate RPN modules and a Detection Head module, two-stage detectors are able to achieve high object localization and classification accuracy. However, the drawback of two-stage detectors like Faster R-CNN is that there are typically hundreds of proposals per image, which makes them computationally heavy and challenging to deploy in resource-limited embedded systems or edge devices.

2) *Single-shot detectors:* Single-shot detectors are designed to avoid the performance bottlenecks of RPN in two-stage detectors. The YOLO [14] framework transforms object detection into a regression problem. Different from a two-stage structure like RPN + Detection Head, YOLO deploys only one single CNN model to do localization and classification at once. YOLO divides the input image into a grid of cells, and for each cell, the predictor generates output predictions for bounding box coordinates, the confidence level of each box, and the class probability. YOLO is designed to mitigate the computational cost of a two-stage detector and aims to be applied for real-time execution. SSD [15] is another single-shot detector aiming to combine the efficiency of YOLO with the accuracy of two-stage detectors. SSD also applies the idea of grid cells and extends the CNN architecture by adding more feature extraction layers to detect objects on a variety of scales. With the multiscale feature pyramid architecture, SSD finds the balance between computation efficiency and object detection accuracy.

In general, a CNN-based object detection model can be separated into three parts: backbone, neck, and head. The function of the backbone is to extract features from source images; the neck is to further extract features for location and classification; and the head is to generate final detection results. The design of neck and head modules depends on different strategies, as mentioned previously. As for the design of the backbone, different feature extraction modules can be deployed inspired by image classification models. For instance, VGG net [16] is used in Faster-RCNN as the backbone feature extractor, and Darknet [14] is used in YOLO. In [17], the SSD is improved by replacing the backbone VGG net with ResNet [18]. In Mobilenet-SSD, Mobilenet [19] is used as the backbone feature extractor. Comparing to the backbone modules mentioned above, Mobilenet is much more efficient due to its special design, which will be introduced in detail later in this section. This advantage makes Mobilenet more suitable for mobile or edge device applications.

2. *Object Detection on UAV Images*

In recent years, due to the success of DL and CNNs in object detection tasks, many works have applied the CNN-based object detectors mentioned in the previous section on UAV object detection [20–23]. One branch of work utilizes two-stage detectors such as Faster-RCNN [24] or Casecade-RCNN [25] to pursue high detection accuracy, especially on small targets in aerial images. In [26], a Faster-RCNN-based multilayer feature fusion model is proposed to enhance the moving object tracking ability in aerial images. However, these methods require high-end Graphics Processing Units (GPUs) to perform high-accuracy object detection, which is not suitable in real UAV applications. The other branch of work utilizes single-shot detectors such as YOLO and SSD to increase efficiency. For instance, authors in [27–29] use the YOLO-based object detection algorithm to detect and track aircraft or other small objects in aerial images. Nevertheless, those works still used GPUs and desktops and did not apply the detectors to microprocessors with low computational power. In this work, we use single-shot detectors due to their efficiency and applicability to embedded systems. We also validated the performance and applicability of the microcomputer on a real UAV.

3. *Federated Learning*

Federate learning (FL) was first introduced by McMahan et al. [30]. In their original work, they proposed a new machine learning scheme that is capable of tackling practical issues such as limited computation and communication power, data heterogeneity, and privacy concerns. In FL, the machine learning model is trained collaboratively by clients in parallel with their own local dataset, and the process is coordinated by a central server. The local data within each client is not transmitted to the central server, which can not only reduce the communication cost but also avoid privacy concerns. Due to the advantages mentioned above, FL has become one of the most popular research directions in the machine learning community, and many studies have been conducted to improve the vanilla FL paradigm in different aspects. Li et al. [31] proposed the algorithm FedProx to tackle the data heterogeneity issue by introducing a proximal term in the local objective function. Zhang et al. [32] utilized active learning and reinforcement learning to reveal the best client selection in each global communication round and achieve better convergence. As another branch of FL research, in [33], the author proposed a hybrid approach combining differential privacy with multiparty computation to address the privacy issue and prevent malicious attacks on the FL system.

There are several challenges when integrating FL with an object detection model. One of the main practical challenges is how to perform FL training and real-time object detection inference under the constraints of resource-limited edge devices. Another challenge is data heterogeneity [34]. Specifically, in urban areas, cars are much more frequently seen compared to bicycles. Such object-level class imbalance could affect the global model detection performance of minority classes after FL training.

B. Contribution

In this paper, an urban object detection system using UAVs is developed. In the system, the Mobilenet-SSD object detection model is deployed on the real UAV for urban object detection. The object detection model is trained in the FL framework because its efficiency and privacy protection properties are perfect for IoT and UAV applications. The main contributions of this paper are 1) implementing and applying FL to train an object detection model and 2) deploying the object detection model on a microprocessor with limited computational power to validate the real-time performance of detecting urban objects from a real aerial image dataset.

II. Methodology

A. Federated Learning

In this section, the FL framework is introduced. In FL, each client k has a local objective $F_k(\omega) = (1/n_k)\sum_{j=1}^{n_k} l(\omega, \boldsymbol{x}_j, \boldsymbol{y}_j)$, where ω is the machine learning model parameters, n_k is the number of samples in client k, $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ is the jth sample-target pair, and l is the loss function. The goal is to minimize the global average of loss, which is stated as follows:

$$\min_{\omega} F(\omega) \coloneqq \sum_{k=1}^{K} p_k F_k(\omega) \tag{1}$$

where K is the total number of clients, $p_k = n_k/n$, and $n = \sum n_k$. To solve this optimization problem, we use FedAvg [30], which consists of four steps in every global iteration, as demonstrated in the upper part of Fig. 1. For ith global iteration, in step 1, the server will randomly select clients and send the global model ω_i to them. In step 2, once the clients receive the global model, they use it as the initial state and start model training with their own local dataset for E epochs. The optimizer for local training is stochastic gradient descent (SGD) in this paper, but it can be any other optimizer, such as Adam [35]. In step 3, after the local training, the clients transmit the updated model ω_E^k back to the server. In step 4, the server aggregates the collated models and generates a new global model ω_{i+1} by weighted average:

$$\omega_{i+1} = \sum_{k=1}^{K} p_k \omega_E^k \tag{2}$$

B. Proposed System Framework

In Fig. 1, the overall proposed system framework is demonstrated. The system consists of two parts: FL training and object detection. In the first part, a global object detection model is trained under the FL

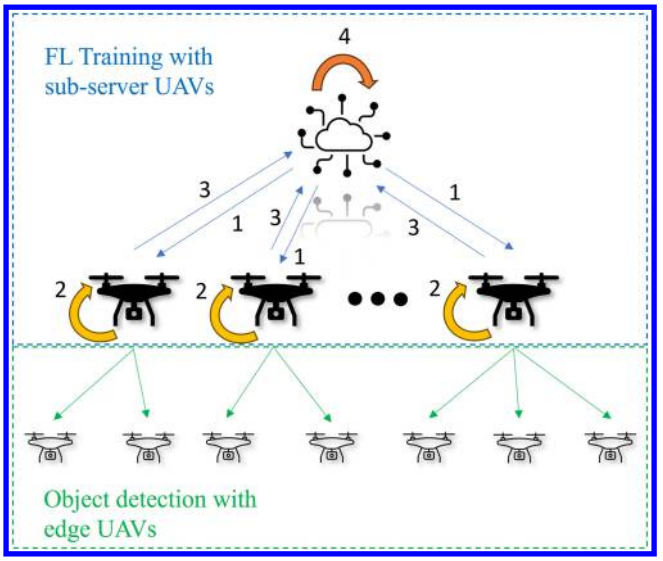

Fig. 1 Demonstration of the proposed FL training and object detection framework.

framework collaboratively by subserver UAVs. These UAVs with better computational power are capable of collecting aerial image data and training locally with their own data. Once the global model is well trained, the subserver UAVs will distribute the model to edge UAVs with limited resources, as illustrated in the lower part of Fig. 1. The edge UAVs perform real-time object detection tasks with the given global model exclusively. The motivation behind this two-tier structure is that using low-power edge UAVs for both training and inference is inefficient. Therefore, the FL training task is done by the subserver UAVs with fewer numbers but better hardware. In this way, the advantages of FL, such as parallel computation and data privacy, can be fully utilized, while the overall system efficiency can also be improved.

C. Mobilenet-SSD Object Detector

Mobilenet-SSD is an object detection method that uses Mobilenet [19] as the backbone feature extractor and SSD [15] as the detection network. It is a single-shot object detection framework designed to be deployed on mobile or embedded systems. The overall CNN architecture is shown in Fig. 2. The following subsections describe the steps performed in the Mobilenet-SSD algorithm.

1. *Feature Extraction*

The purpose of feature extraction is to extract high-level features from the original images. The features are further used in object localization and classification. Feature extraction heavily relies on convolutional operations, which are time-consuming. To accelerate the computational speed, the Mobilenet model is built on depthwise separable convolutions (DSCs), which are a form of factorized convolution operations. The DSC operation is illustrated in Fig. 3. DSCs factorize a standard convolution into a depthwise convolution and a 1×1 pointwise convolution. This factorization has the effect of drastically reducing computation and model size. For example, in Fig. 3, assume the input size of a convolution layer is $F \times F$, kernel size is 3×3, and the output channels are 4. Conventional convolution requires $3 \times 3 \times 3 \times 4 \times F^2 = 108F^2$ operations. On the other hand, the DSC only requires $3 \times 3 \times 3 \times F^2 + 1 \times 1 \times 3 \times 4 \times F^2 = 39F^2$. In general, the reduction in computation is $(1/M) + (1/D_k^2)$, where M is the number of output channels and $1/D_k$ is the kernel size. Besides the backbone Mobilenet, the extra feature pyramid network and Detection Head networks in SSD also use DSCs to replace conventional convolution operations for efficiency. The detailed layer structures are listed in Table 1.

2. *Prediction of Bounding Box and Class Probability*

The extracted features are sent to the Detection Head, which is a module consisting of convolution layers. The Detection Head takes extracted features from different scales and generates bounding boxes and class probabilities for localization and classification, respectively. The bounding box is defined by four values (b_x, b_y, b_w, b_h), where b_x, b_y are the coordinates of the bounding box center and b_w, b_h are the width and height of the box. The prediction of class probability is represented by a c-dimensional vector, where c is the number of classes.

3. *Nonmaximum Suppression*

In the end, the nonmaximum suppression (NMS) is applied to suppress the nonmaximum bounding boxes and find the best bounding box prediction as the final prediction result.

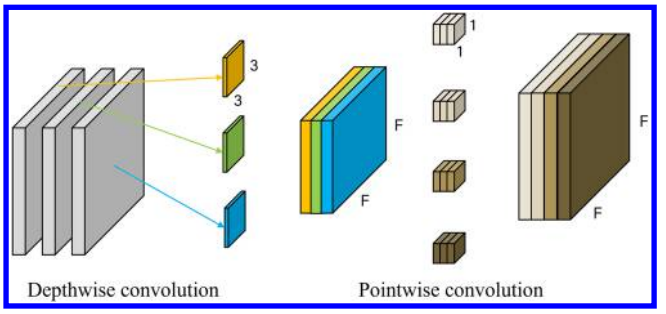

Fig. 3 Example of a depthwise separable convolution.

III. Experimental Results

A. Dataset

To train and test the Mobilenet-SSD object detection model for UAV images, the public dataset Visdrone [36] is used. Visdrone is an open-source image dataset containing aerial images captured by UAVs. The training set used in the training process consists of 6471 images, and the testing set used in validation consists of 548 images. The 12 class names and their distribution in the training set are listed in Table 2.

B. Experimental Setup

The Mobilenet-SSD object detection model and FL training script are implemented in Pytorch. The proposed training scheme is simulated on an offline computer. For the FL setting, the Visdrone dataset is divided and distributed to four clients, simulating the subserver UAVs. The global training iteration is 90. In each global iteration, the clients do 10 local training epochs. The following are the hyperparameter settings in clients' local training for the experiment. The input image is resized to 512×512, the optimizer is SGD with momentum set to 0.9, the learning rate is 0.05, and the batch size is 24. Image data augmentation includes random flip, random crop, and PhotometricDistort. The trained model is deployed to Jetson Nano 4 GB, the onboard computer for the edge UAV, for performance testing. The edge UAV and its onboard computer used in the experiment are shown in Fig. 4.

C. Evaluation Index

To quantitatively evaluate the detection performance of the object detection network, the same protocol as Common Objects in Context (COCO) is used. Four indices, recall R, precision P, average precision AP, and mean average precision mAP, are used in the protocol. The definition of recall is

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

where TP is true positive and FN is false negative. TP means that the predictor detects a positive class object and that there is one in reality. FN means that the predictor says there is no class object but there is actually one object. R is an index showing the percentage of objects that the detector can find. The definition of precision is

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

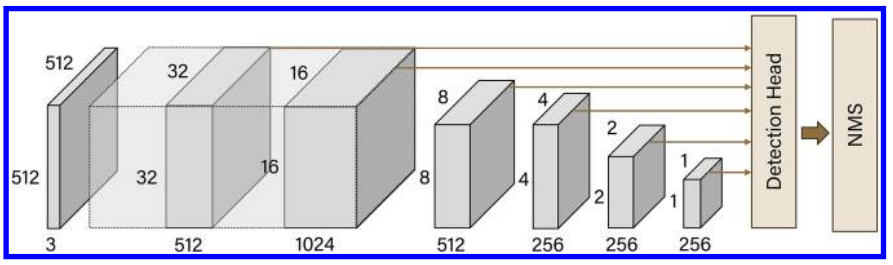

Fig. 2 Mobilenet-SSD model architecture.

Table 1 Mobilenet-SSD architecture

Input size	Type/stride	Filter shape	Note
$512 \times 512 \times 3$	Conv/s2	$3 \times 3 \times 3 \times 32$	
$256 \times 256 \times 32$	DSConv/s1	$3 \times 3 \times 32 + 1 \times 1 \times 32 \times 64$	
$256 \times 256 \times 64$	DSConv/s2	$3 \times 3 \times 64 + 1 \times 1 \times 64 \times 128$	
$128 \times 128 \times 128$	DSConv/s1	$3 \times 3 \times 128 + 1 \times 1 \times 128 \times 128$	
$128 \times 128 \times 128$	DSConv/s2	$3 \times 3 \times 128 + 1 \times 1 \times 128 \times 256$	
$64 \times 64 \times 256$	DSConv/s1	$3 \times 3 \times 256 + 1 \times 1 \times 256 \times 256$	
$64 \times 64 \times 256$	DSConv/s2	$3 \times 3 \times 256 + 1 \times 1 \times 256 \times 512$	
$32 \times 32 \times 512$	DSConv/s1	$5 \times (3 \times 3 \times 512 + 1 \times 1 \times 512 \times 512)$	Output used in Detection Head
$32 \times 32 \times 512$	DSConv/s2	$3 \times 3 \times 512 + 1 \times 1 \times 512 \times 1024$	
$16 \times 16 \times 1024$	DSConv/s1	$3 \times 3 \times 1024 + 1 \times 1 \times 1024 \times 1024$	
$16 \times 16 \times 1024$	Conv/s1	$1 \times 1 \times 1024 \times 256$	
$16 \times 16 \times 256$	DSConv/s2	$3 \times 3 \times 256 + 1 \times 1 \times 512 \times 128$	Output used in Detection Head
$8 \times 8 \times 512$	Conv/s1	$1 \times 1 \times 512 \times 128$	
$8 \times 8 \times 128$	DSConv/s2	$3 \times 3 \times 128 + 1 \times 1 \times 64 \times 256$	Output used in Detection Head
$4 \times 4 \times 256$	Conv/s1	$1 \times 1 \times 256 \times 128$	
$4 \times 4 \times 128$	DSConv/s2	$3 \times 3 \times 128 + 1 \times 1 \times 128 \times 256$	Output used in Detection Head
$2 \times 2 \times 256$	Conv/s1	$1 \times 1 \times 256 \times 128$	
$2 \times 2 \times 128$	DSConv/s2	$3 \times 3 \times 128 + 1 \times 1 \times 128 \times 256$	Output used in Detection Head
$1 \times 1 \times 256$	Conv/s1	$1 \times 1 \times 256 \times 128$	
$1 \times 1 \times 128$	DSConv/s2	$3 \times 3 \times 128 + 1 \times 1 \times 128 \times 256$	Output used in Detection Head

Table 2 Visdrone training dataset and object class distributions

Class name	Object number
Ignored regions	8,813
Pedestrian	79,337
People	27,059
Bicycle	10,480
Car	144,867
Van	24,959
Truck	12,875
Tricycle	4,812
Awning tricycle	3,246
Bus	5,926
Motor	29,647
Others	1,532

where FP is false positive. FP means that the predictor detects a positive class object but actually there is no object. P is an index showing the correctness of the detector's prediction. The definition of average precision is

$$\mathrm{AP} = \int_0^1 P(R)\mathrm{d}R \tag{5}$$

AP is an index showing the overall performance considering R and P at once. Last, mAP is the average of AP's of all classes.

$$\mathrm{mAP} = \sum_{i=1}^{C} \mathrm{AP}_i \tag{6}$$

In this paper, the indices AP and mAP are used since only R or P itself cannot precisely represent the performance. This is because when R is large it is less false negative but it can potentially have more incorrect detections, which leads to low precision P. Similarly, when P is large it can still have many miss detections, which leads to a low recall R. Therefore, it is more reasonable to consider AP, which takes both R and P into account at the same time. We are also interested in mAP since it indicates the overall performance among all the different classes.

D. Results

The object detection results are illustrated in Fig. 5. The proposed detection model can successfully detect most of the large and medium-sized traffic objects in different scenes, such as highway, intersection, and construction zone and at night. All the detection results are generated by the onboard microcomputer Jetson Nano, and the processing speed is around 18 FPS. This result validates the real-time operation capability of the resource-limited hardware. The evaluation indices are also shown in Table 3. The performance is compared with the model trained in a centralized scheme, and the detection result comparison is demonstrated in Fig. 6. One can observe that the performance of the model trained in the FL scheme is slightly degraded, but the difference is insignificant, which verifies the efficiency and effectiveness of FL in training the object detection model. The AP results for different classes are also shown in the same

Fig. 4 The UAV (left) and the onboard computer (right) used in the experiment in this work.

Fig. 5 Object detection results in different scenarios.

Table 3 AP and mAP of the models for different classes on Visdrone test set

Class name	Mobilenet-SSD (centralized), %	Mobilenet-SSD (FL), %
Ignored regions	1.3	3.0
Pedestrian	11.5	11.1
People	11.0	10.6
Bicycle	9.7	9.5
Car	42.9	40.1
Van	20.1	17.8
Truck	24.1	18.9
Tricycle	11.8	10.9
Awning tricycle	11.1	11.3
Bus	39.8	36.4
Motor	13.3	9.8
Others	1.4	3.1
mAP	17.9	16.3

table. Most of the missed detections are for small objects, such as cars far away or pedestrians. Another typical classification error is between pedestrians and people, which are extremely difficult to classify by attributes.

IV. Conclusions

In this paper, a system for urban object detection utilizing UAVs with FL is developed. The system is in a two-tier structure, including training a global object detection model via FL by subserver UAVs collaboratively and deploying on-edge UAVs for inference exclusively. The trained object detection model is deployed and tested on the real UAV's onboard microprocessor. For the object detection model, the Mobilenet-SSD object detector is used due to its lightweight and efficiency, which are suitable for a resource-limited microprocessor on an edge UAV. To train the model more efficiently

Fig. 6 Detection result comparison between the FL-trained model (up) and the centralized learning trained model (down).

and to consider the privacy issue, the FL framework is applied to train the object detection model. The model trained by FL is deployed on a real UAV and tested with a real-world traffic dataset in the experiments. The object detection model can not only successfully detect

vehicles in different scenarios but also small objects such as pedestrians. The experiments also indicate that the detection speed can reach 18 FPS for the microprocessor, which validates the real-time operation capability on edge devices. We compare the performance of the model trained by the FL scheme with the model trained by the centralized scheme as well. The results suggest that training with FL can affect performance, but the degradation is insignificant, which verifies the effectiveness and efficiency of FL in this application.

For future work, some tracking algorithms, can be aggregated into the system for broader engineering applications. Applying more advanced neural network structures, such as attention mechanisms, can be another potential research direction. Last but not the least, finding the balance between performance and resource constrains is also an important research direction for real-world applications in the future.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1955890.

References

[1] Kaneko, S., and Martins, J. R., "Fleet Design Optimization of Package Delivery Unmanned Aerial Vehicles Considering Operations," *Journal of Aircraft*, Vol. 60, No. 4, 2023, pp. 1061–1077. https://doi.org/10.2514/1.C036921

[2] Liu, R., Shin, H.-S., and Tsourdos, A., "Edge-Enhanced Attentions for Drone Delivery in Presence of Winds and Recharging Stations," *Journal of Aerospace Information Systems*, Vol. 20, No. 4, 2023, pp. 216–228. https://doi.org/10.2514/1.I011171

[3] Xie, G., and Chen, X., "Efficient and Robust Online Trajectory Prediction for Non-Cooperative Unmanned Aerial Vehicles," *Journal of Aerospace Information Systems*, Vol. 19, No. 2, 2022, pp. 143–153. https://doi.org/10.2514/1.I010997

[4] Ching, P. L., Tan, S. C., and Ho, H. W., "Ultra-Wideband Localization and Deep-Learning-Based Plant Monitoring Using Micro Air Vehicles," *Journal of Aerospace Information Systems*, Vol. 19, No. 11, 2022, pp. 717–728. https://doi.org/10.2514/1.I011075

[5] Ko, C.-H., Ren, H., Tsai, J.-R., Wang, B.-J., Lin, S.-F., Huang, C.-H., Hong, C.-T., and Chiu, W.-H., "Agriculture Application with Airborne Hyperspectral Images from Two-Dimensional Concave Grating System," *AIAA Scitech 2019 Forum*, AIAA Paper 2019-1542, 2019. https://doi.org/10.2514/6.2019-1542

[6] Glasheen, K., Pinto, J., Steiner, M., and Frew, E., "Assessment of Finescale Local Wind Forecasts Using Small Unmanned Aircraft Systems," *Journal of Aerospace Information Systems*, Vol. 17, No. 4, 2020, pp. 182–192. https://doi.org/10.2514/1.I010747

[7] Tian, P., Chao, H., Rhudy, M., Gross, J., and Wu, H., "Wind Sensing and Estimation Using Small Fixed-Wing Unmanned Aerial Vehicles: A Survey," *Journal of Aerospace Information Systems*, Vol. 18, No. 3, 2021, pp. 132–143. https://doi.org/10.2514/1.I010885

[8] Rhudy, M. B., "Predicting the Parameters of Stochastic Wind Models for Time-Varying Wind Estimation Techniques," *Journal of Aerospace Information Systems*, Vol. 16, No. 2, 2019, pp. 71–76. https://doi.org/10.2514/1.I010652

[9] Elloumi, M., Dhaou, R., Escrig, B., Idoudi, H., and Saidane, L. A., "Monitoring Road Traffic with a UAV-Based System," *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, Inst. of Electrical and Electronics Engineers, New York, 2018, pp. 1–6. https://doi.org/10.1109/WCNC.2018.8377077

[10] Khan, M. A., Ectors, W., Bellemans, T., Janssens, D., and Wets, G., "UAV-Based Traffic Analysis: A Universal Guiding Framework Based on Literature Survey," *Transportation Research Procedia*, Vol. 22, Jan. 2017, pp. 541–550. https://doi.org/10.1016/j.trpro.2017.03.043

[11] Lv, Y., Duan, Y., Kang, W., Li, Z., and Wang, F.-Y., "Traffic Flow Prediction with Big Data: A Deep Learning Approach," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 2, 2014, pp. 865–873. https://doi.org/10.1109/TITS.2014.2345663

[12] Viola, P., and Jones, M., "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Vol. 1, Inst. of Electrical and Electronics Engineers, New York, 2001, pp. I-511–I-518. https://doi.org/10.1109/CVPR.2001.990517

[13] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 6, 2017, pp. 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

[14] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You Only Look Once: Unified, Real-Time Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Inst. of Electrical and Electronics Engineers, New York, 2016, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91

[15] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., "SSD: Single Shot Multibox Detector," *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, Springer–Verlag, Berlin, 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[16] Simonyan, K., and Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014.

[17] Lu, X., Kang, X., Nishide, S., and Ren, F., "Object Detection Based on SSD-ResNet," *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, Inst. of Electrical and Electronics Engineers, New York, 2019, pp. 89–92. https://doi.org/10.1109/CCIS48116.2019.9073753

[18] He, K., Zhang, X., Ren, S., and Sun, J., "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Inst. of Electrical and Electronics Engineers, New York, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

[19] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H., "Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.

[20] Valasek, J., Kirkpatrick, K., May, J., and Harris, J., "Intelligent Motion Video Guidance for Unmanned Air System Ground Target Surveillance," *Journal of Aerospace Information Systems*, Vol. 13, No. 1, 2016, pp. 10–26. https://doi.org/10.2514/1.I010198

[21] Hu, Y., Cao, Y., Ding, M., and Zhuang, L., "Airport Detection for Fixed-Wing Unmanned Aerial Vehicle Landing Using a Hierarchical Architecture," *Journal of Aerospace Information Systems*, Vol. 16, No. 6, 2019, pp. 214–223. https://doi.org/10.2514/1.I010615

[22] Desai, A., and Lee, D.-J., "Efficient Feature Descriptor for Unmanned Aerial Vehicle Ground Moving Object Tracking," *Journal of Aerospace Information Systems*, Vol. 14, No. 6, 2017, pp. 345–350. https://doi.org/10.2514/1.I010503

[23] Kang, C., Chaudhry, H., Woolsey, C. A., and Kochersberger, K., "Development of a Peripheral–Central Vision System for Small Unmanned Aircraft Tracking," *Journal of Aerospace Information Systems*, Vol. 18, No. 9, 2021, pp. 645–658. https://doi.org/10.2514/1.I010909

[24] Avola, D., Cinque, L., Diko, A., Fagioli, A., Foresti, G. L., Mecca, A., Pannone, D., and Piciarelli, C., "MS-Faster R-CNN: Multi-Stream Backbone for Improved Faster R-CNN Object Detection and Aerial Tracking from UAV Images," *Remote Sensing*, Vol. 13, No. 9, 2021, Paper 1670. https://doi.org/10.3390/rs13091670

[25] Albaba, B. M., and Ozer, S., "SyNet: An Ensemble Network for Object Detection in UAV Images," *2020 25th International Conference on Pattern Recognition (ICPR)*, Inst. of Electrical and Electronics Engineers, New York, 2021, pp. 10,227–10,234. https://doi.org/10.1109/ICPR48806.2021.9412847

[26] Han, S.-C., Zhang, B.-H., Li, W., and Zhan, Z.-H., "Moving Object Detection for Airport Scene Using Patterns of Motion and Appearance," *Journal of Aerospace Information Systems*, Vol. 18, No. 11, 2021, pp. 852–859. https://doi.org/10.2514/1.I010902

[27] Ying, J., Li, H., Yang, H., and Jiang, Y., "Small Aircraft Detection Based on Feature Enhancement and Context Information," *Journal of Aerospace Information Systems*, Vol. 20, No. 3, 2023, pp. 140–151. https://doi.org/10.2514/1.I011160

[28] Huang, G., Zhang, X., Zhao, R., Li, W., Liang, B., and Xie, J., "Efficient Small-Object Detection in Airport Surface Based on Maintain Feature High Resolution," *Journal of Aerospace Information Systems*, Vol. 19,

No. 4, 2022, pp. 305–316.
https://doi.org/10.2514/1.I011004

[29] Dolph, C. V., Ippolito, C., Glaab, L. J., Logan, M. J., Tran, L. D., Danette Allen, B., Alam, M., Li, J., and Iftekharuddin, K., "Adversarial Learning Improves Vision-Based Perception from Drones with Imbalanced Datasets," *Journal of Aerospace Information Systems*, Vol. 20, No. 8, 2023, pp. 1–19.
https://doi.org/10.2514/1.I011185

[30] McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A., "Communication-Efficient Learning of Deep Networks from Decentralized Data," *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Vol. 54, JMLR: W&CP, Fort Lauderdale, Florida, 2017, pp. 1273–1282.

[31] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V., "Federated Optimization in Heterogeneous Networks," *Proceedings of Machine Learning and Systems*, Vol. 2, March 2020a, pp. 429–450.

[32] Zhang, H., Xie, Z., Zarei, R., Wu, T., and Chen, K., "Adaptive Client Selection in Resource Constrained Federated Learning Systems: A Deep Reinforcement Learning Approach," *IEEE Access*, Vol. 9, July 2021, pp. 98,423–98,432.
https://doi.org/10.1109/ACCESS.2021.3095915

[33] Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., and Zhou, Y., "A Hybrid Approach to Privacy-Preserving Federated Learning," *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, Assoc. for Computing Machinery, New York, 2019, pp. 1–11.
https://doi.org/10.1145/3338501.3357370

[34] Li, T., Sahu, A. K., Talwalkar, A., and Smith, V., "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine*, Vol. 37, No. 3, 2020, pp. 50–60.
https://doi.org/10.1109/MSP.2020.2975749

[35] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," arXiv preprint arXiv:1412.6980, 2014.

[36] Cao, Y., He, Z., Wang, L., Wang, W., Yuan, Y., Zhang, D., Zhang, J., Zhu, P., Van Gool, L., Han, J., et al., "VisDrone-DET2021: The Vision Meets Drone Object Detection Challenge Results," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Inst. of Electrical and Electronics Engineers, New York, 2021, pp. 2847–2854.
https://doi.org/10.1109/ICCVW54120.2021.00319

P. Wei
Associate Editor