IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 November 2022, accepted 24 December 2022, date of publication 3 January 2023, date of current version 13 March 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3233841

RESEARCH ARTICLE

Swarm Reconnaissance Drone System for Real-Time Object Detection Over a Large Area

SUNGTAE MOON[1], JIHUN JEON[2], (Graduate Student Member, IEEE), DOYOON KIM[3], AND YONGWOO KIM[4], (Member, IEEE)

[1]School of Computer Engineering, Korea University of Technology and Education, Cheonan-si 31253, Republic of Korea
[2]Department of Electrical and Computer Engineering, Inha University, Incheon 22212, Republic of Korea
[3]Korea Aerospace Research Institute (KARI), Deajon 34133, Republic of Korea
[4]Department of System Semiconductor Engineering, Sangmyung University, Cheonan-si 31066, Republic of Korea

Corresponding author: Yongwoo Kim (yongwoo.kim@smu.ac.kr)

This work was supported in part by the Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science and ICT, Republic of Korea, under Grant 2020M3C1C1A01083163; in part by the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) under Grant 2022R1G1A1007415; and in part by the MSIT under the Information Technology Research Center (ITRC) Support Program through the Institute for Information & Communications Technology Planning & Evaluation (IITP) under Grant IITP-2021-0-02052.

ABSTRACT Recent developments in drone technology have led to the widespread use of unmanned aerial vehicles (UAVs). In particular, UAVs are often used in reconnaissance to detect objects such as missing persons in large areas. However, traditional systems use only one UAV to search for missing persons in a large area. In addition, object detection is performed after flight or manually because detection requires high computing power. In this paper, a reconnaissance drone system using multiple UAVs is proposed. The proposed multi-UAV reconnaissance system performs real-time object detection on each UAV. The real-time object detection results from each UAV are received by the ground control system (GCS) to stitch the images. To enable real-time object detection in individual UAVs, the filter pruning method is applied to the YOLOv5 model, and the model uses 40% fewer parameters than the existing baseline model. The lightweight YOLOv5 model achieves approximately 11.73 FPS on the Jetson Xaiver NX using a mission computer. Moreover, the proposed image stitching method enables image stitching by effectively matching features using additional information generated by UAVs. The UAV flight tests show that the proposed reconnaissance system can monitor and detect objects in real time over large areas.

INDEX TERMS Image stitching, network pruning, real-time object detection, swarm flight system.

I. INTRODUCTION

With the recent development of drone technology, drones are now used in various applications, such as reconnaissance systems in large, dangerous areas that are difficult for humans to directly search and analyze. Advances in artificial intelligence have dramatically improved object detection technology to find people or cars. However, since most missions are operated with a single drone, the scope and time of operation are limited. In addition, due to the performance limitation of unmanned aerial vehicles (UAVs), detecting objects in real time is difficult, making an immediate response impossible. These limitations have spurred research on swarm flights using multiple drones, which allow missions to be performed by dividing a large area and achieve cooperation by assigning drones different missions.

For a swarm reconnaissance system, a swarm operation system that simultaneously controls and manages multiple drones is needed. Based on the system, an image stitching algorithm that synchronizes the images received from the drones and merges them into a single matched image is necessary. The integrated image helps the user effectively understand and make decisions about the overall situation. Then, a real-time object detection algorithm is needed to detect missing persons or intruders. For object detection, deep learning algorithms have been used. However, the processing

The associate editor coordinating the review of this manuscript and approving it for publication was Zhongyi Guo.

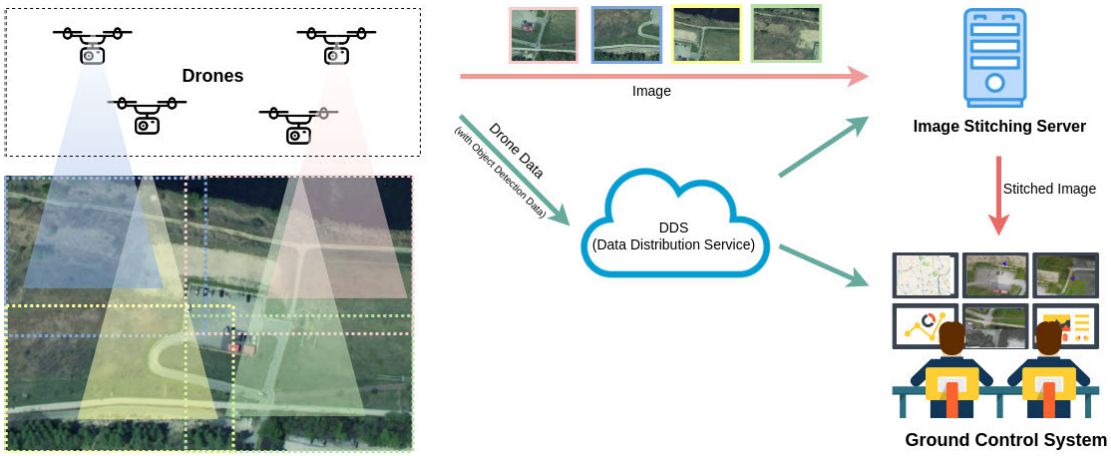

FIGURE 1. Proposed swarm reconnaissance drone system overview.

is performed outside the UAV or as postprocessing because of its high computational cost.

In this paper, a data distributed service-based swarm reconnaissance drone system is proposed in which the system simultaneously controls and operates multiple drones using safe and integrated commands, as shown in Fig. 1. The proposed system receives independent images from each drone and stitches the images while detecting objects inside the drones in real time. As a result, the ground control system (GCS) provides full situational awareness in real time. The proposed system was verified through an object detection test based on a stitched image obtained by drones.

The main contributions of this paper can be summarized as follows:

1) A real-time object detection method based on UAV images is proposed. A proposed swarm reconnaissance drone system is designed with the goal of processing 10 frames per second (fps) for the execution time of real-time object detection. To achieve the requirement of 10 fps in the Jetson Xavier NX system used in the drone, the proposed filter pruning method for the lightweight network is applied to achieve object detection performance.
2) Real-time image stitching is proposed for the swarm drone system. The proposed image stitching method effectively matches features using additional information generated by the UAVs.
3) Flight experiments of UAVs are conducted to verify the feasibility of the proposed methods.

The remainder of this paper is structured as follows. In Section II, related research on swarm UAV systems and object detection for UAV imagery is described. Section III describes the overall architecture of the proposed swarm reconnaissance UAV system with aerial image stitching and real-time object detection. Section IV briefly describes the experimental setup and results. In Section V, the conclusions of this paper are discussed.

II. RELATED WORKS

A. UAV SWARM

Swarm flight technology has the advantage of being able to quickly complete a mission by dividing the mission into a large number of smaller areas. As a result, many swarm flight studies are being actively conducted. Early swarm flight studies were mostly conducted through simulations because flight validation was difficult due to the high cost [1], [2], [3]. With the development and miniaturization of drone technology, swarm flight technology has been researched indoors. In particular, an image-based marker, a motion capture sensor, or a UWB sensor has been used to estimate the position indoors without GPS [4], [5]. However, indoor swarm flight technology cannot be expanded and developed outdoors due to the narrow operating range of the sensor. For outdoor position estimation, an extended Kalman filter that combines a GNSS sensor and an IMU sensor is generally used. However, in the case of outdoor swarm flight, GNSS sensors are insufficient because position accuracy within 1 m is required to avoid collisions between drones. Moon et al. [6] proposed an RTK-GPS-based position estimation algorithm to improve position accuracy and built a system that allows hundreds of drones to operate within 1 m of each other without colliding.

Additionally, robust communication is required for stable swarm flight. Moon et al. [6] proposed a passive approach that minimizes the amount of communication by inputting the missions to the drones in advance for stable communication with hundreds of drones. However, the passive approach is not suitable for a swarm reconnaissance system because performing a dynamic mission is difficult. Cho et al. [7] proposed an efficient network configuration method based on a distributed network according to the mission to increase communication efficiency. However, their proposed method was only verified by simulations and only considered two dimensions, which is insufficient for application to a real environment. Moreas et al. [8] proposed proactive link maintenance mechanisms to create a self-organizing

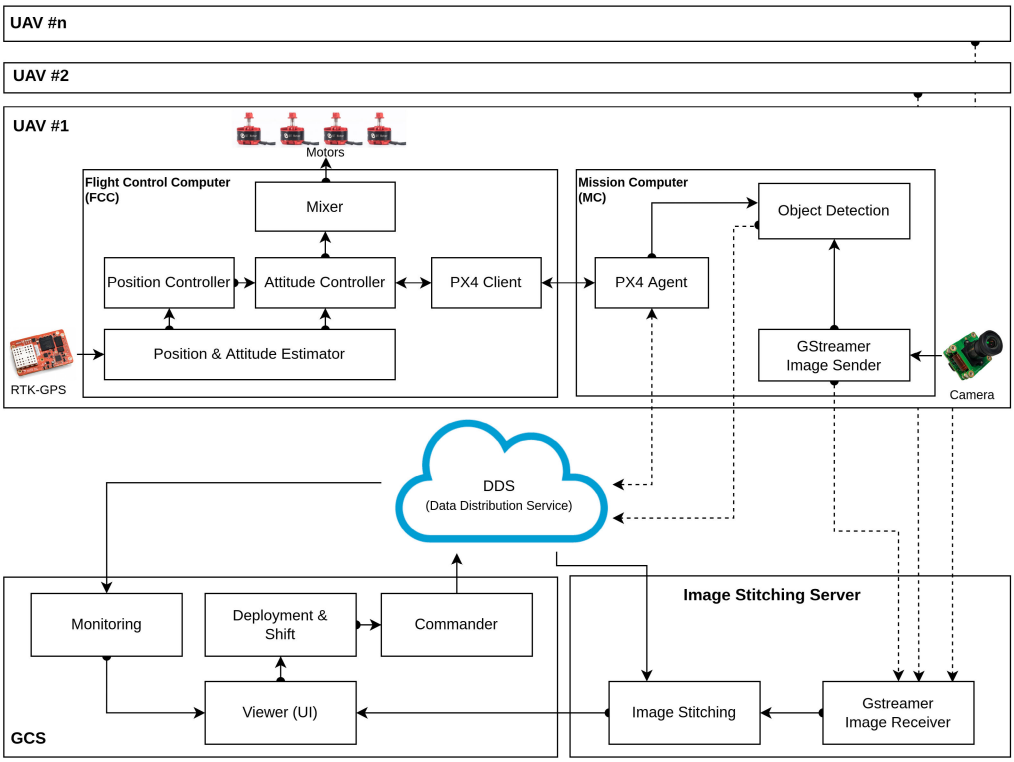

FIGURE 2. **Proposed swarm reconnaissance drone system architecture.**

flying network capable of providing network support for the UAV nodes already engaged in exploration and targeting tasks in surveillance missions.

In addition, a method that avoids drone collisions and minimizes the energy usage for swarm reconnaissance systems is needed. Kopfstedt et al. [9] performed optimization using the mixed integer quadratic programming (MIQP) method. However, since verification was performed through simulations, verification in a real environment is necessary. Moon et al. [6] used the Fair Hungarian method considering the battery condition to ensure that the battery consumption of the drones was even and verified their approach with 100 drones. However, this method is not suitable for a swarm reconnaissance drone system because performing a dynamic mission is difficult due to the passive mission approach. The Defense Advanced Research Projects Agency (DARPA) is developing a swarm collaboration system through the offensive swarm-enabled tactics (OFFSET) project [10]. However, OFFSET is not suitable for use in a swarm reconnaissance system that performs search missions for missing persons because it has focused on the development of a collaborative method involving both humans and unmanned vehicles.

When a swarm reconnaissance system is deployed, a variety of missions over a large area can be performed. An image stitching algorithm is necessary for full situational awareness. Yahyanejad et al. [11] proposed a hybrid approach that combines inaccurate information on the camera's position and orientation with image data using traditional approaches. Lucier et al. [12] tried image registration with satellite images and unprocessed aerial images captured with a UAV. However, these approaches stitch images with postprocessing. In addition, when a missing person is in the forest, traditional approaches are not sufficient to find image features.

B. OBJECT DETECTION FOR UAV IMAGERY

Recently, deep learning-based object detection algorithms have been extensively studied in the drone field. Object detection algorithms for drones are different from those for tasks such as autonomous driving. In the drone internal object detection algorithm, the object to be detected generally has a very small object size in the image. The small size of the object to be found means that there is little information about the characteristics of the object that can be expressed in the image, which makes training the deep learning network difficult.

To solve this problem, various studies have been conducted to detect small objects in object detection used in drones. Chen et al. [13] proposed an anchor-free-based RRNet network detector including a re-regression module. The proposed regression module creates a more accurate bounding box. They also proposed an adaptive resampling technique to increase the amount of training data. Yang et al. [14] proposed the ClusDet network to better detect small objects by classifying images based on clusters and performing object

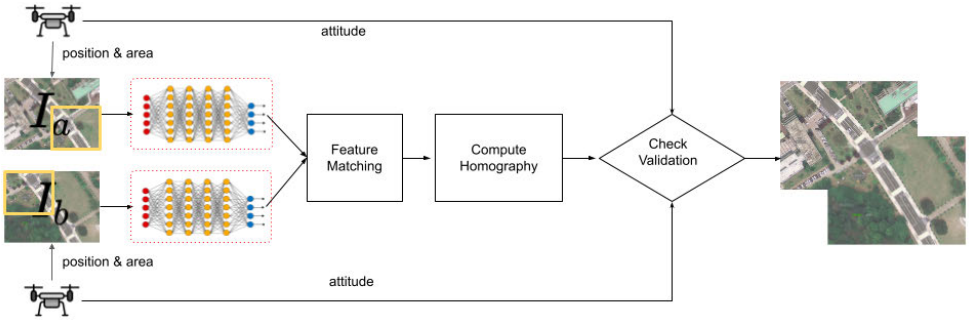

FIGURE 3. Proposed image stitching method.

detection on the clustered images. In addition, Liu [15] proposed multibranch parallel feature pyramid networks (MPFPNs) based on the Cascade-RCNN [16] network to improve the accuracy of detection of small objects in drone images.

Another important characteristic of drone object detection is that high-performance processing is impossible due to the performance limitations of the internal system of the drone. Therefore, to enable object detection within the drone device, the complexity of the deep learning-based network should be low. Wu et al. [17] proposed a lightweight YOLOv3 network [18] for real-time object detection. Zhang et al. [19] proposed SlimYOLOv3, which balances the number of parameters, memory usage, and inference time for YOLOv3.

Ammar et al. [20] proposed a network operating at approximately 12 fps at 608 $\times$ 608 resolution using the YOLOv4 [21] network. Additionally, various YOLO series, such as YOLOX [22] using decoupled heads and YOLOv7 [23] using model reparameterization techniques, have appeared. Based on the YOLOv5 [24] network, a strip bottleneck module was proposed to enable real-time object detection, and the SPB-YOLO [25] network was proposed. As in previous research, when the network complexity decreases, the accuracy generally decreases, so an object detection network that optimizes complexity and accuracy is needed. In this paper, a filter pruning method based on YOLOv5 is applied to improve the detection accuracy of small objects and enable real-time object detection.

III. PROPOSED SWARM RECONNAISSANCE DRONE SYSTEM

The swarm reconnaissance drone system consists of a GCS, UAVs, an image stitching server, and an RTK-GPS base station, as shown in Fig. 2. The UAV system is divided into a flight control computer and a mission computer. For the flight control computer (FCC), a PX4 is used, which is based on open source software running on the NuttX real-time operation system. In the PX4, each module communicates with other modules through the micro object request broker (uORB) message-driven method. The uORB is designed following a publish-subscribe model. The FCC estimates and controls the position and attitude using various sensors, including RTK-GPS, which enables more precise position estimation than the GPS sensors. Using this system, position estimation at the centimeter level can be realized. The PX4 client module transmits messages to the mission computer (MC) by packet serialization. The PX4 agent receives a PX4 message from the PX4 client in the MC. The MC is developed based on ROS2, which is a distributed middleware system running on the Linux operating system. The mission computer focuses on image processing for object detection. The image sender module receives video streaming data from the camera and transmits it to the object detection module and image stitching server. For object detection, the data are transmitted without any image quality downgrade. However, when the image data are transmitted to the image stitching server, the image quality is downgraded because of the limitations of the LTE bandwidth.

The communication between a UAV and other parts of the system uses the data distribution service (DDS) of ROS2 to increase robustness. Therefore, the system can monitor the status and give commands with duplicate GCSs. The GCS receives many monitoring messages, including objects, to check the UAV status. In addition, the GCS receives the stitched image from the image stitching server. Users can check the current status and objects using the registered image. To increase the operating time, the proposed system uses the UAV shift algorithm. The algorithm replaces the UAV with another UAV on the ground if the battery status of the UAV is too low. All commands are passed through ROS2 messages via a commander module. The image stitching server collects video streaming data from the UAVs through gstreamer. After converting the video streaming data to an image, the image stitching module stitches the images.

A. AERIAL IMAGE STITCHING

Since each image is received separately from the different drones, it should be synchronized and integrated as a merged image through an image stitching algorithm for full situational awareness. The image stitching algorithm extracts feature points from an image, matches the feature points with those in the images of the other drones, and then registers the image through homography estimation.

The homography matrix indicates a relationship between the points of a reference image and the points of a target

image. The homography is defined as follows:

$$X' = HX \tag{1}$$

X denotes the feature points as $X = (x, y, 1)^T$, and X' denotes the feature points of the target images. The homography is basically estimated by applying 4 feature points of a reference image and 4 points of a target image via direct linear transformation (DLT). However, an error occurs when extracting feature points between the reference image and the target image, so an accurate homography cannot be extracted. Therefore, to extract the correct feature points and estimate the optimal homography, the outlier feature points must be removed using the RANdom SAmple Consensus (RANSAC) algorithm [26].

To extract image features in general, traditional feature extractors such as the SIFT [27] or SURF [28] algorithm are used. However, these algorithms are not sufficient to extract the features of images received from drones. When a drone is used to detect a missing person in a river or a forest, the image received from the drone has few feature points. Particularly in the case of swarm reconnaissance, the overlapping area required for matching is relatively small because reconnaissance must be performed over a wide area. Therefore, the feature points required for matching are insufficient. In addition, real-time image stitching from a high-definition image cannot be used because the extraction of feature points sharply increases the computational burden.

To overcome these difficulties of image stitching during swarm reconnaissance, a more powerful and faster image registration method is proposed using deep learning-based feature point extraction and drone status, as shown in Fig. 3. The deep learning-based feature extractor can replace traditional feature extractors with convolutional neural networks (CNNs) because they extract complex features that represent images in much greater detail, learn task-specific features and are much more efficient. The proposed image stitching algorithm uses the SuperPoint [29] and SuperGlue [30] algorithms.

For robust image registration, a homography estimation method is proposed that considers the distance from the center of the area including the feature point when extracting the feature point. The homography matrix is estimated in the DLT method using RANSAC in the same manner as in the normal method. The position transformation of the reference image feature points through the estimated homography is expressed as follows:

$$\hat{X}_i' = \hat{H} X_i \tag{2}$$

To determine the accuracy of the estimated homography, the sum of the errors between the n estimated feature points and the target image feature points is obtained as follows:

$$\epsilon = \sum_{i=1}^{n} |\eta(X_i' - \hat{X}_i')| \tag{3}$$

Generally, the error is expressed as the Euclidean distance between the position of the estimated feature point and the position of the target feature point. However, in the case of a swarm reconnaissance system operating in a small overlapping area within a wide area, the homography estimation accuracy is low even if the error is small. To solve this problem, the weight of the feature point error away from the center is reduced as follows:

$$\eta(X_i' - \hat{X}_i') = \frac{r_{max} + r_i}{r_{max}} \sqrt{(X_i' - \hat{X}_i')^T (X_i' - \hat{X}_i')} \tag{4}$$

where r_i is the distance between the i^{th} matched feature point and the center of the overlapping region and m represents the maximum value of r_{max}.

Additionally, in the case of a swarm reconnaissance system, image registration is incorrectly estimated because the number of feature points and the overlapping area are small. Therefore, the homography matrix must be analyzed and corrected. The homography can be represented in matrix form as follows:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}, \tag{5}$$

where the 2×2 matrix that has $h_{11}, h_{12}, h_{21}, h_{21}$ elements contains rotation and scale information. h_{13} and h_{23} contain information regarding translations along the x-axis and y-axis, respectively. h_{31} and h_{32} contain the perspective transformation information. In addition, (6) can detect distortion, inversion, and concavity issues during image registration when D is negative.

$$D = h_{11}h_{22} - h_{12}h_{21} \tag{6}$$

Equation (7) can be used to determine the degree of projection. In the case of the proposed system, the homography estimation was often incorrect due to the small overlapping area, and the validity of the homography can be confirmed through the P value. In this paper, the homography estimation was confirmed to be incorrect when the P value was greater than 0.002.

$$P = \sqrt{h_{31}^2 + h_{32}^2} \tag{7}$$

If the transformation is judged to be an abnormal homography transformation through the P and D values, then it is performed as an affine transformation rather than a projective transformation as follows:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

If the UAV attitude is changed by the wind during the reconnaissance operation, then projection transformation is needed, but since the reconnaissance is performed from a fixed position and the camera attitude is stabilized by the gimbal, the affine transformation alone is sufficient to carry out the mission. Therefore, even if a slight error occurs, image registration is possible even after the affine transformation. In addition, if there is no overlapping region or an error

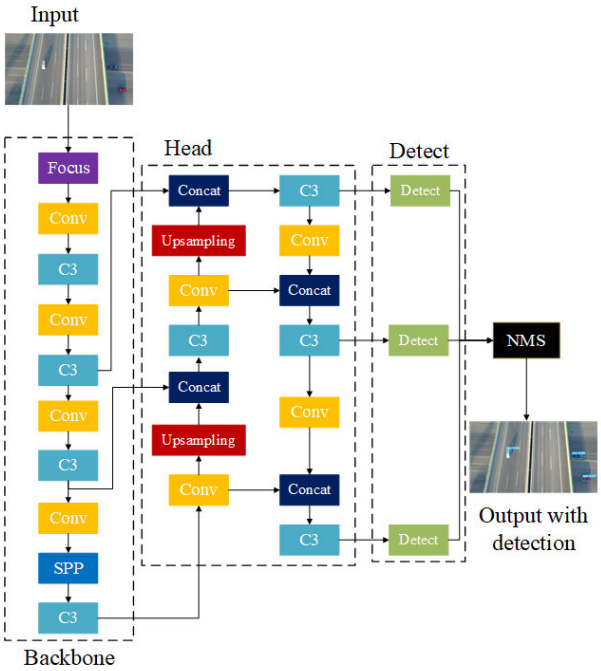

FIGURE 4. Architecture of the object detection network of the YOLOv5-large model.

occurs in the affine transformation, then image registration can be carried out using only the position and direction information of the UAV.

B. REAL-TIME OBJECT DETECTION

Deep learning-based object detection networks can be divided into two-stage object detection networks and one-stage object detection networks. The R-CNN [16] series is a representative two-stage object detection network technique, whereas the YOLO [18], [21] series is a representative one-stage object detection network method. In general, a two-stage object detection network performs better than a one-stage network, but its inference speed is slower. In this paper, YOLOv5 [24], which has excellent performance in one-stage networks, is selected as the default network for real-time object detection. To attain a fast inference time while maintaining the detection performance of the basic network, the weight of the network is reduced by extending the existing filter pruning method.

As shown in Fig. 4, YOLOv5 is largely divided into backbone, head, and detect. The backbone and head consist of the Conv, Focus, C3, and SPP modules. In addition, several versions of the network have been proposed for YOLOv5, such as s(small), m(medium), l(large), and x(xlarge), depending on the size of the network. In this paper, the large model is adopted as the baseline network. In addition, the previously developed filter pruning method that optimizes the amount of computation and parameters simultaneously [31] is applied in this study. Furthermore, by applying the quantization method supported by the TensorRT framework to the lightweight model for real-time object detection, the filter-pruned network can be used with only INT8 integer operations.

The target capacity filter pruning (TCFP) method [31] for real-time object detection developed in the previous study consists of three stages, as shown in Fig. 5. A detailed description of the TCFP method can be found in [31]. The TCFP method reduces both the computational cost and the number of parameters to optimize the object detection network. The first stage of the TCFP method, the sparsity learning stage, is the process of learning important connections (filters) in the network. This process is trained by adding a pruning loss function to the original loss function used in the YOLOv5 model.

In this paper, sparsity learning was performed on a pre-trained YOLOv5-large baseline network. The pruning process, which is the second stage of the proposed TCFP method, determines the insignificant filters from the sparsity learning results and prunes them. Finally, based on the pruned network architecture, a retraining process is performed after initialization. To determine the filter to be pruned in the sparsity learning stage, the indicator function of (9) is applied. t denotes a threshold, and γ is a scaling factor learned in the batch normalization layer. If the value is less than the threshold, then the channel is considered unimportant, and the corresponding channel and filter can be removed.

$$\theta(\gamma, t) = \begin{cases} 0, & \text{if } |\gamma| \le t \\ 1, & \text{if } |\gamma| > t \end{cases} \tag{9}$$

However, (9) is impossible to differentiate at threshold t, and the differential value in the remaining differentiable parts is zero. Therefore, a straight-through estimator as in (10) is used for backpropagation.

$$\frac{\partial \theta(\gamma, t)}{\partial \gamma} = \begin{cases} -1, & \text{if } \gamma \le t \\ 1, & \text{if } \gamma > t \end{cases} \tag{10}$$

During pruning in the sparsity learning stage, the amount of computation of the pruned network is given by (11).

$$F_{pruned} = \sum_{l=1}^{L} \left\{ F_l \left(\frac{\sum_{c=1}^{C_{l-1}} \theta(\gamma_{l-1,c}, t)}{C_{l-1}} \right) \left(\frac{\sum_{c=1}^{C_l} \theta(\gamma_{l,c}, t)}{C_l} \right) \right\} \tag{11}$$

where F_l is the amount of computation of the L-th convolutional layer, L is the number of layers, and C is the number of channels. On the right-hand side of (11), the first term represents the effect of the FLOPs reduced by the channel that disappeared from the previous layer, and the second term represents the effect of the FLOPs reduced by the filter that disappeared from the current layer. Similar to the amount of computation, the number of parameters after pruning can be expressed as (12).

$$P_{pruned} = \sum_{l=1}^{L} \left\{ P_l \left(\frac{\sum_{c=1}^{C_{l-1}} \theta(\gamma_{l-1,c}, t)}{C_{l-1}} \right) \left(\frac{\sum_{c=1}^{C_l} \theta(\gamma_{l,c}, t)}{C_l} \right) \right\} \tag{12}$$

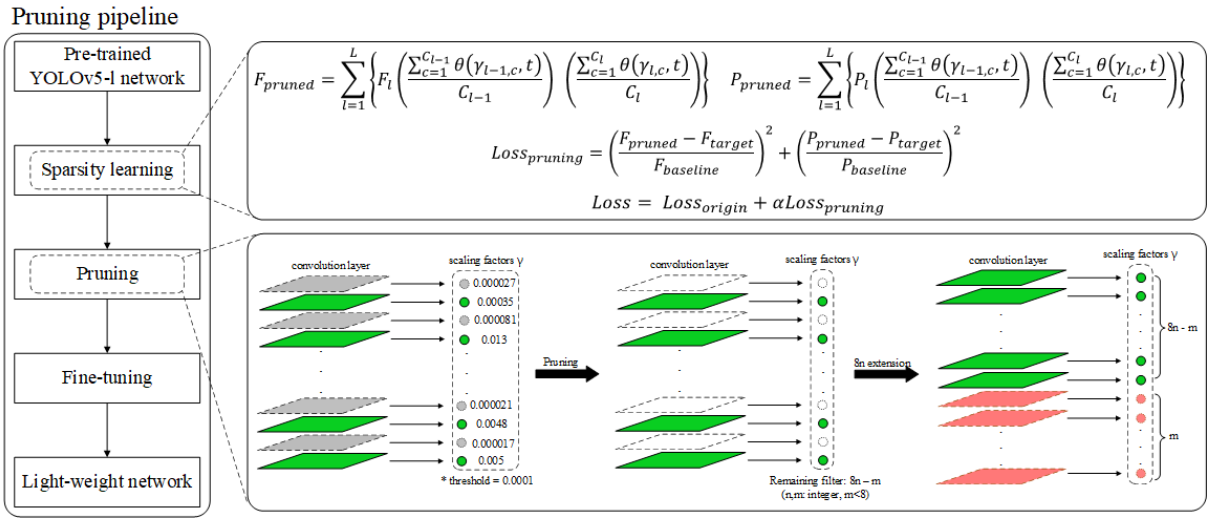

FIGURE 5. Filter pruning pipeline using the target capacity filter pruning (TCFP) method [31] for the lightweight YOLOv5 model.

Equations (11) and (12) can be used to create a loss function for pruning, which is given by (13).

$$Loss_{pruning} = \left(\frac{F_{pruned} - F_{target}}{F_{baseline}}\right)^2 + \left(\frac{P_{pruned} - P_{target}}{P_{baseline}}\right)^2 \tag{13}$$

In the above formula, F_{target} and P_{target} represent the amount of computation and the number of parameters of the target network, respectively. By adding the pruning loss function to the loss function of the existing object detection network and proceeding with sparsity learning, a network with the desired number of parameters and FLOPs can be created. The overall loss function in object detection of YOLOv5 is given by (14).

$$Loss = Loss_{origin} + \alpha Loss_{pruning} \tag{14}$$

$Loss_{origin}$ is the loss function used in YOLOv5, $Loss_{pruning}$ is the loss function applied by (13), and α is a hyperparameter. The $\alpha Loss_{pruning}$ is empirically set to 8 at the start of sparsity learning.

For example, if the number of target parameters and the number of FLOPs are set to 70% of the existing network, then the initial $Loss_{pruning}$ value before sparsity learning is 0.18. In this case, the α value will be 8 divided by 0.18. In a network trained by sparsity learning, a lightweight network can be created by pruning channels and filters according to the batch normalization layer with γ less than the threshold t applied in the indication function. The threshold t is set to 0.0001.

In the previous study [31], experiments confirmed that the greatest improvement in inference speed was obtained when the number of channels was extended by a multiple of 8 based on the architecture of the pruned network. Therefore, in this paper, an 8n extension that makes the number of channels a multiple of 8 is applied based on the architecture of the pruned network. After pruning, the network is retrained to restore the mAP performance of the pruned network.

IV. EXPERIMENTAL RESULTS

A. REAL-TIME IMAGE STITCHING

To evaluate the performance of the image stitching method proposed in this paper, 3 datasets were used to represent different environments with different image feature points. Dataset #1 was extracted from a heliport with many feature points. Datasets #2 and #3 were drawn from a playground with few feature points at elevations of 10 m and 30 m from each other, respectively. Simple objects were added in an environment lacking feature points, the overlapping area was set to 30% or less, and stitching was then performed. The drone used in the experiment was equipped with a DJI Osmo Pocket 2 camera, and a built-in gimbal was used to eliminate image blur caused by the jello phenomenon. The image stitching server operates with an Intel(R) Xeon(R) Silver CPU and is equipped with 4 Nvidia Titan RTX GPUs.

The image stitching was compared with the results using the image stitching algorithm cv::Stitcher provided by OpenCV. Each feature point extraction was performed via SIFT, SURF, SuperGlue, and this work. In test1 (Dataset #1), which had sufficient image feature points, all algorithms stitched images without error, as shown in Fig. 6. However, different stitching results were obtained in tests 2 4, which had few feature points.

As shown in Fig. 6 (a) and (b), SIFT- and SURF-based image stitching have difficulty in accurate extraction in an environment lacking feature points, and inaccurate matching frequently occurs. As shown in Fig. 6(c), when SuperGlue was used, more than twice as many key points were extracted than when the existing SIFT or SURF was used, and accurate matching was possible. In addition, even though a heavy network based on deep learning was constructed, the processing

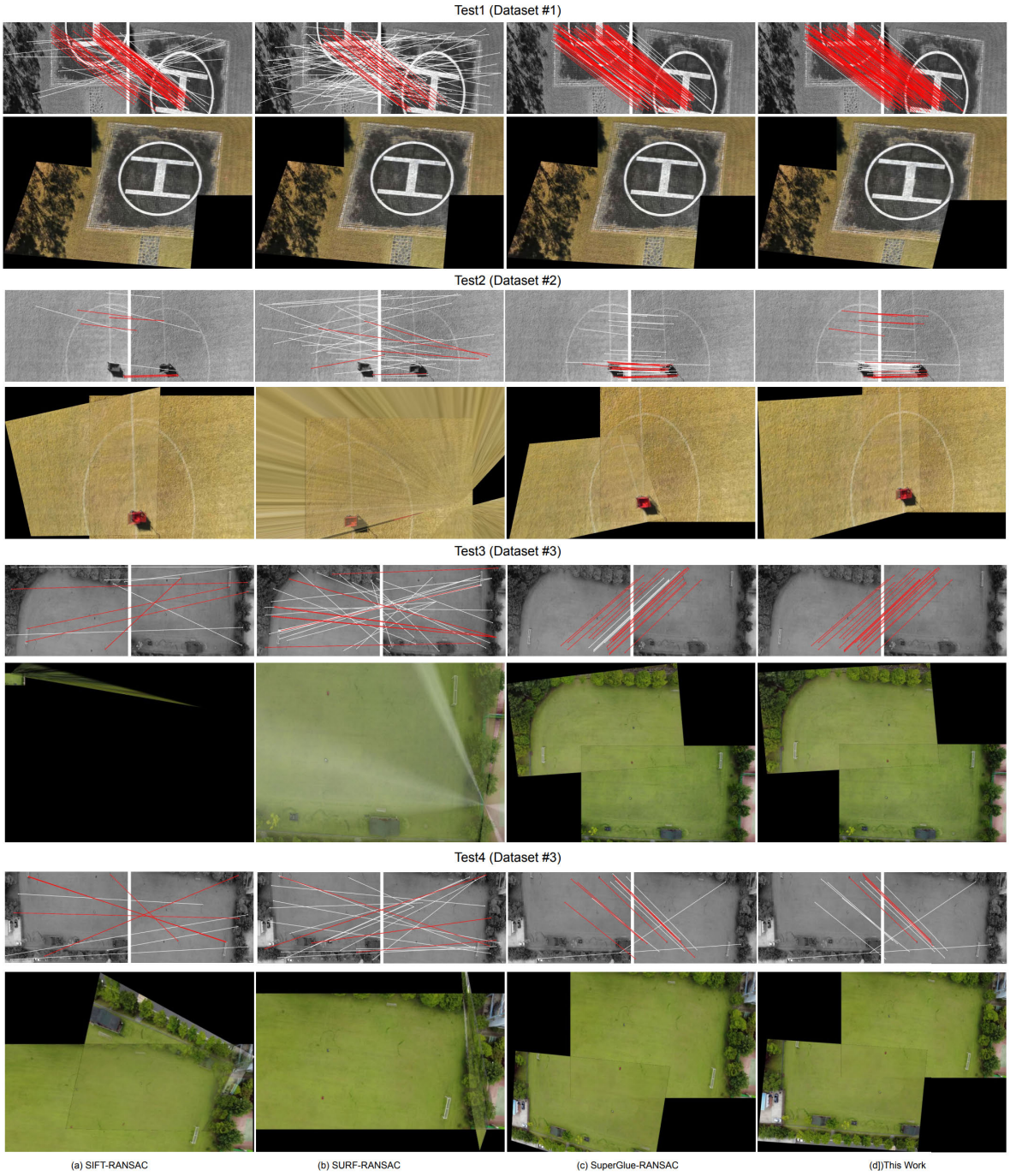

FIGURE 6. Image stitching result.

speed was improved by using a GPU. However, the homography estimation was often wrong due to the narrow overlapping area. The method proposed in this paper enabled more accurate homography estimation by adding feature weights according to positions, as shown in Fig. 6(d). A comparison of the image registration results is shown in Table 1.

B. REAL-TIME OBJECT DETECTION

The VisDrone-2019 [32] dataset was used to train the object detection network and evaluate the performance of the network. The VisDrone dataset consists of 6,471 training images and 1,610 test images. Each image has 3 channels, and information about the bounding box is provided in the form of a text file.

The batch size required for training all networks, including sparsity learning, was fixed at 32. In addition, the image size was fixed to 1376 $\times$ 1376 during training and evaluation. Other training and evaluation procedures were performed following the default settings in [32]. The inference time was measured after fixing the batch size to 1 on the Nvidia

TABLE 1. Comparison of image stitching performance.

Dataset	Method	Matching features	Valid matching features	Computing time (msec)
Test1 (Dataset #1)	RANSAC + SIFT	107	60	151
	RANSAC + SURF	133	28	162
	RANSAC + SuperPoint	128	105	45
	This work	128	124	46
Test2 (Dataset #2)	RANSAC + SIFT	12	8	121
	RANSAC + SURF	30	6	144
	RANSAC + SuperPoint	45	19	36
	This work	45	24	60
Test3 (Dataset #3)	RANSAC + SIFT	10	5	95
	RANSAC + SURF	31	9	76
	RANSAC + SuperPoint	18	13	33
	This work	18	9	36
Test4 (Dataset #3)	RANSAC + SIFT	11	7	107
	RANSAC + SURF	21	7	73
	RANSAC + SuperPoint	14	7	36
	This work	14	4	37

Jetson Xavier NX platform used as the mission computer. The PyTorch framework was used, and an additional network lightweighting process was performed through the TensorRT framework, which quantizes to INT8 to improve the inference time.

As shown in Table 2, the performance of the existing pruning methods and the proposed pruning method was evaluated for three different ratios. The first row of Table 2 shows the baseline results of the YOLOv5 model without pruning. Liu's method [33] cannot control the pruning ratio of the parameters and FLOPs at the same time. Therefore, pruning was performed based on the parameters. The HFP method [34] and the proposed method were trained with the target of pruning both parameters and FLOPs by 30%, 40% and 50% each.

When the parameters were pruned by 40% using Liu's method, both the number of parameters and the number of FLOPs were greater than those for the network in which both the parameters and FLOPs were pruned by 40% by the proposed filter pruning method, but the mAP (0.5:0.95) was 0.2% lower. In addition, it was difficult to effectively increase the inference speed of the network because the numbers of FLOPs of the three networks pruned by Liu's method were smaller than the FLOPs of the network pruned by 30% by the proposed method, and the FPS was lower. In the HFP method, the target pruning rate was meaningless, and the network was excessively pruned. Considerable effort was required to find the appropriate hyperparameter for obtaining a pruned network according to the target pruning rate.

When training was conducted with the target of 30% pruning through the proposed filter pruning method, the parameters and FLOPs were pruned by 32%, reaching the target. Additionally, the inference speed was improved by 24.3% compared with the baseline network time, and the mAP (0.5:0.95) drop was 0.6%. When the parameters and FLOPs were each pruned by 40%, a YOLOv5 network was obtained in which the parameters and FLOPs were pruned by 40%, and mAP (0.5:0.95) was decreased by 1%. In this case, the inference speed was improved by 42.2% compared with the YOLOv5 baseline network FPS. Last, when the parameters and FLOPs were each pruned by 50%, a YOLOv5 network was obtained in which the parameters and FLOPs were pruned by 51.2% and 46%, and mAP (0.5:0.95) decreased by 0.9%. The inference speed was improved by 58.7%. In addition, the inference speed performance after quantization with INT8 using the NVIDIA TensorRT framework improved at a similar rate to that of the FP16 precision.

FIGURE 7. Photograph of the UAV system.

By eliminating a large number of parameters and FLOPs, the proposed pruning method can reduce memory usage and power consumption. In addition, the proposed pruning method can improve the inference time performance at the same time with minimal accuracy drop when running an object detection network on an embedded device with hardware constraints mainly used in UAVs.

C. INTEGRATED OPERATION EXPERIMENTS

The UAV platform was designed to replace the currently employed mission board and used a general quadcopter frame equipped with an open-source PX4 system [35] that had an IMU (an LSM303D integrated accelerometer/magnetometer and L3GD20 gyroscope) and a barometer (MS5611, TE Connectivity), as shown in Fig. 7. In addition, it was equipped with an RTK-GPS sensor (Piksi, Swift Navigation) for

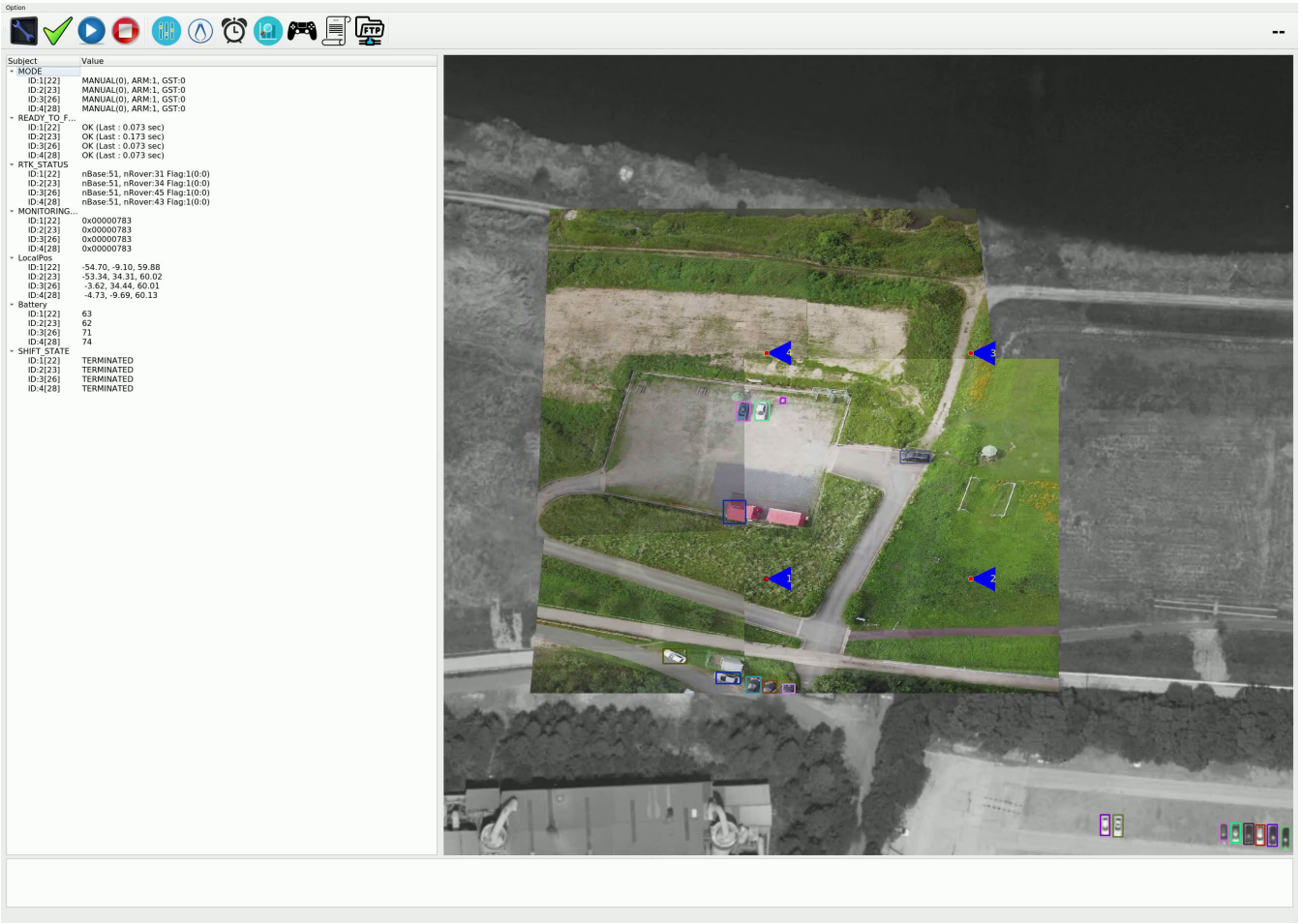

FIGURE 8. Experimental results of the swarm reconnaissance drone system.

TABLE 2. Comparison of mAP, the number of parameters, and inference time.

Model	Method	pruning rate (Parameters/FLOPs)	mAP (0.5)	mAP (0.5:0.95)	Params(M)	GFLOPs	FPS	
							Pytorch (FP16)	TensorRT (INT8)
YOLOv5-large	Baseline	-	45.4	26.7	46	528.8	4.6	9.24
	[33]	30%/	43.9	25.5	32	351.0	4.79	9.44
	[34]	30%/30%	44.2	25.7	26	322.1	5.71	10.83
	This work	30%/30%	44.7	26.1	31	359.3	5.72	11.08
	[33]	40%/	43.8	25.5	27	323.7	5.07	10.07
	[34]	40%/40%	44.2	25.7	20	275.6	5.96	11.28
	This work	40%/40%	44.2	25.7	27	316.6	6.54	11.73
	[33]	50%/	44.1	25.6	22	299.2	5.49	10.47
	[34]	50%/50%	44.3	25.5	20	266.8	6.37	11.51
	This work	50%/50%	44.3	25.8	22	285.7	7.3	12.01

accurate positioning. As a result, the UAV was able to estimate the position within 5 cm. The mission board included an NVIDIA Jetson Xavier NX platform running the Robot Operating System version 2 (ROS2) and a camera including a gimbal (DJI Osmo Pocket 2). The mission board was set to 20 W mode to deliver up to 21 TOPS (Tera Operations Per Second) for running object detection in the UAV. To reduce the weight, a carrier board was developed to integrate all sensors. Table 3 shows the specifications of the drone system.

The target of the experiments was to detect people and cars in a designated area. Once the area for the surveillance missions and the number of drones were determined, the GCS informed each drone where to move to cover the area. Then, the drones moved to their fixed positions and transmitted

TABLE 3. Specifications of the UAV system.

Description	UAV
weight	2706 g
payload	2 kg
flight time	20 min
dimension	650 mm
propellor	13 inch

images and object detection results. The GCS stitched the images and displayed the detection results.

The experiment was conducted at Daejeon Drone Park in Korea, and 4 UAVs took off from the same location and moved to locations designated by the ground station to hover at an altitude of 50 m. The flight proceeded for 10 min, and people and vehicles were detected, as shown in Fig. 8. This experimental result video is available at [36].

Images from each UAV were stitched into the merged image and mapped with satellite maps using latitude and longitude information. As a result of the experimental operation, image matching was performed in real time while detecting cars and persons even if the object was small.

V. CONCLUSION

In this paper, a swarm reconnaissance drone system was proposed for real-time object detection over a large area. To optimize the object detection model, a filter pruning method that optimizes parameters and inference time simultaneously was applied to the swarm reconnaissance system. As a result, the lightweight model with the filter pruning method enabled real-time object detection within the drone, and the ability of the swarm reconnaissance system to perform missions in a large area was confirmed. In addition, a stitching method was proposed to stitch drone images in the GCS. The swarm reconnaissance drone system can be used in various applications, such as monitoring forest fires or searching for missing people. Experiments in various environments are planned to increase the robustness of the system and stability. In future work, a robust image stitching algorithm will be studied to continue image stitching while moving drones. Additionally, an autonomous movement coordination algorithm for drone swarms in exploratory area surveillance missions can also be explored.

REFERENCES

[1] D. J. Bennet, C. R. MacInnes, M. Suzuki, and K. Uchiyama, "Autonomous three-dimensional formation flight for a swarm of unmanned aerial vehicles," *J. Guid., Control, Dyn.*, vol. 34, no. 6, pp. 1899–1908, Nov. 2011, doi: 10.2514/1.53931.

[2] H. Guo, J. Pang, L. Han, and Z. Shan, "Flight data visualization for simulation & evaluation: A general framework," in *Proc. 5th Int. Symp. Comput. Intell. Design*, vol. 1, 2012, pp. 497–502.

[3] C.-L. Huo, T.-Y. Lai, and T.-Y. Sun, "The preliminary study on multi-swarm sharing particle swarm optimization: Applied to UAV path planning problem," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 1770–1776.

[4] S. Moon, D. Cho, S. Han, D. Rew, and E.-S. Sim, "Development of multiple AR.Drone control system for indoor aerial choreography," *Trans. Jpn. Soc. Aeronaut. Space Sci., Aerosp. Technol. Jpn.*, vol. 12, pp. a59–a67, Jan. 2014. [Online]. Available: https://www.jstage.jst.go.jp/article/tastj/12/APISAT-2013/12_TJSAS-D-14-00027/_article

[5] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, "Cooperative grasping and transport using multiple quadrotors," in *Distributed Autonomous Robotic Systems* (Springer Tracts in Advanced Robotics), A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M. A. Hsieh, L. E. Parker, and K. Støy, Eds. Berlin, Germany: Springer, 2013, pp. 545–558, doi: 10.1007/978-3-642-32723-0_39.

[6] S. Moon, D. Lee, D. Lee, D. Kim, and H. Bang, "Energy-efficient swarming flight formation transitions using the improved fair Hungarian algorithm," *Sensors*, vol. 21, no. 4, p. 1260, Feb. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/4/1260

[7] J. Cho, J. Sung, J. Yoon, and H. Lee, "Towards persistent surveillance and reconnaissance using a connected swarm of multiple UAVs," *IEEE Access*, vol. 8, p. 157906–157917, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9178811/

[8] R. S. Moraes and E. P. Freitas, "Distributed control for groups of unmanned aerial vehicles performing surveillance missions and providing relay communication network services," *J. Intell. Robot. Syst.*, vol. 92, pp. 645–656, Nov. 2017, doi: 10.1007/s10846-017-0726-z.

[9] T. Kopfstedt, M. Mukai, M. Fujita, and C. Ament, "Control of formations of UAVs for surveillance and reconnaissance missions," *IFAC Proc. Volumes*, vol. 41, no. 2, pp. 5161–5166, 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016397622

[10] K. Giles and K. Giammarco, "Mission-based architecture for swarm composability (MASC)," *Proc. Comput. Sci.*, vol. 114, pp. 57–64, Jan. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050917317994

[11] S. Yahyanejad, D. Wischounig-Strucl, M. Quaritsch, and B. Rinner, "Incremental mosaicking of images from autonomous, small-scale UAVs," in *Proc. 7th IEEE Int. Conf. Adv. Video Signal Based Surveill.*, Aug. 2010, pp. 329–336.

[12] Lucier and W. Jordan, "Automatic UAV image registration using feature detection and matching with satellite imagery," M.S. Thesis, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2018. [Online]. Available: https://dspace.mit.edu/handle/1721.1/119920

[13] C. Chen, Y. Zhang, Q. Lv, S. Wei, X. Wang, X. Sun, and J. Dong, "RRNet: A hybrid detector for object detection in drone-captured images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 100–108.

[14] F. Yang, H. Fan, P. Chu, E. Blasch, and H. Ling, "Clustered object detection in aerial images," 2019, *arXiv:1904.08008*.

[15] Y. Liu, F. Yang, and P. Hu, "Small-object detection in UAV-captured images via multi-branch parallel feature pyramid networks," *IEEE Access*, vol. 8, pp. 145740–145750, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9162036/

[16] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," 2017, *arXiv:1712.00726*.

[17] Q. Wu and Y. Zhou, "Real-time object detection based on unmanned aerial vehicle," in *Proc. IEEE 8th Data Driven Control Learn. Syst. Conf. (DDCLS)*, May 2019, pp. 574–579. [Online]. Available: https://ieeexplore.ieee.org/document/8908984/

[18] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[19] P. Zhang, Y. Zhong, and X. Li, "SlimYOLOv3: Narrower, faster and better for real-time UAV applications," 2019, *arXiv:1907.11093*.

[20] A. Ammar, A. Koubaa, M. Ahmed, A. Saad, and B. Benjdira, "Vehicle detection from aerial images using deep learning: A comparative study," *Electronics*, vol. 10, no. 7, p. 820, Mar. 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/7/820

[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[22] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding Yolo series in 2021," 2021, *arXiv:2107.08430*.

[23] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.

[24] G. Jocher et al., Jan. 5, 2021. *Ultralytics/YOLOv5: V4.0—Nn.SiLU() Activations, Weights & Biases Logging, Pytorch Hub Integration*. [Online]. Available: https://zenodo.org/record/4418161

[25] X. Wang, W. Li, W. Guo, and K. Cao, "SPB-YOLO: An efficient real-time detector for unmanned aerial vehicle images," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Apr. 2021, pp. 099–104. [Online]. Available: https://ieeexplore.ieee.org/document/9415214/
[26] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981, doi: 10.1145/358669.358692.
[27] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Feb. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.
[28] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Computer Vision—ECCV* (Lecture Notes in Computer Science), A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Germany: Springer, 2006, pp. 404–417.
[29] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 337–33712.
[30] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4937–4946.
[31] J. Jeon, J. Kim, J.-K. Kang, S. Moon, and Y. Kim, "Target capacity filter pruning method for optimized inference time based on YOLOv5 in embedded systems," *IEEE Access*, vol. 10, pp. 70840–70849, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9815045/
[32] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 7380–7399, Nov. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9573394/
[33] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2736–2744. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2017/html/Liu_Learning_Efficient_Convolutional_ICCV_2017_paper.html
[34] L. Enderich, F. Timm, and W. Burgard, "Holistic filter pruning for efficient deep neural networks," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 2596–2605. [Online]. Available: https://openaccess.thecvf.com/content/WACV2021/html/Enderich_Holistic_Filter_Pruning_for_Efficient_Deep_Neural_Networks_WACV_2021_paper.html
[35] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 6235–6240.
[36] S. Moon. *Swarm Reconnaissance Drone System*. YouTube. Accessed: Aug. 30, 2022. [Online]. Available: https://youtu.be/fcFZ1k68bd0

SUNGTAE MOON received the B.S. degree from Chonnam National University, Gwangju, South Korea, in 2005, the M.S. degree from the Gwangju Institute of Science and Technology (GIST), Gwangju, in 2007, and the Ph.D. degree in aeronautical engineering from KAIST, in 2021. From 2007 to 2010, he was at Agency for Defense Development (ADD), where he developed a mission computer for aircraft. From 2011 to 2012, he was at the National Security Research Institute (NSRI) and worked in the area of security in embedded systems. From 2012 to 2022, he was a Senior Researcher with the Artificial Intelligence Research Division, Korea Aerospace Research Institute (KARI). Since 2022, he has been with the Korea University of Technology and Education (KOREATECH), Cheonan-si, South Korea, where he is currently an Assistant Professor with the School of Computer Science and Engineering. His current research interests include swarming flight systems, navigation algorithms, and object detection based on deep learning.

JIHUN JEON (Graduate Student Member, IEEE) received the B.S. degree in electronics engineering from Inha University, where he is currently pursuing the M.S. degree in electrical and computer engineering. His research interests include deep learning, computer vision, and systems-on-chip design.

DOYOON KIM received the B.S. degree from Hanseo University, Chungnam, South Korea, in 2013, and the M.S. degree from Sejong University, Seoul, South Korea, in 2015. Since 2015, he has been a Senior Researcher with the Aerospace Engineering Research Division, Korea Aerospace Research Institute (KARI). His current research interests include swarming flight systems, navigation and control algorithms, and model-based design.

YONGWOO KIM (Member, IEEE) received the B.S. and M.S. degrees from Inha University, Incheon, South Korea, in 2007 and 2009, respectively, and the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2019. From 2009 to 2017, he was a Senior Engineer with LX Semicon Company Ltd., Daejeon. From 2019 to 2020, he was a Senior Researcher with the Artificial Intelligence Research Division, Korea Aerospace Research Institute, Daejeon. Since 2020, he has been with Sangmyung University, Cheonan-si, South Korea, where he is currently an Assistant Professor with the Department of System Semiconductor Engineering. His current research interests include image/video processing algorithms, superresolution, and deep learning hardware architecture for vision processing.