PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

A deep-learning, vision-based framework for testing swarm algorithms using inexpensive mini drones

Sayani Sarkar, Nathan Johnson

Sayani Sarkar, Nathan Johnson, "A deep-learning, vision-based framework for testing swarm algorithms using inexpensive mini drones," Proc. SPIE 12124, Unmanned Systems Technology XXIV, 1212409 (31 May 2022); doi: 10.1117/12.2618137

Event: SPIE Defense + Commercial Sensing, 2022, Orlando, Florida, United States

A deep-learning, vision-based framework for testing a swarm algorithm using inexpensive mini drones

Sayani Sarkar[1] and Nathan Johnson[1]

Bellarmine University, 2001 Newburg Rd, Louisville, KY, USA

ABSTRACT

The ability to explore dangerous buildings or hostile landscapes using a swarm of inexpensive mini drones is relevant to many search and rescue or surveillance scenarios encountered by civilian first responders and military personnel. Swarms of mini drones, implementing various path planning algorithms, provide a unique solution in situations where there is the risk to human life or use of expensive Unmanned Aerial Vehicle technology would be cost-prohibitive or both. Although inexpensive, off-the-shelf drones contain stabilization circuitry and onboard cameras, they suffer restricted flying time and lack GPS systems. The limited capability of such drones has curtailed their use by researchers investigating practical search and genetic algorithms, and many researchers rely on simulation, rather than testing with actual drones. In this paper, we describe an ad hoc framework for testing swarm algorithms while taking the first step toward implementing swarm intelligence using low-cost, off-the-shelf drones and an inexpensive network router. We initially created a public dataset, MINIUAV, including images of Tello and TelloEdu mini-drones taken from our live drone video recordings and photos scraped from various internet resources. Using the images, we then trained a deep-learning-based YOLOv4-Tiny (You Only Look Once) object detector allowing us to implement a swarm intelligence rule where drones act collectively based on a swarm alignment rule. Our results show the object detector allows a drone to identify a neighboring drone with greater than 90% accuracy. Finally, the dataset used to train the object detector will be made available on request.

Keywords: Artificial neural networks, computer vision, object detection, unmanned aerial vehicles, YOLO, target following, swarm intelligence, path planning, mini drone

1. INTRODUCTION

Low-cost aerial drones provide a unique solution in situations where there is a risk to human life or the use of more expensive unmanned aerial vehicle (UAV) technology would be cost-prohibitive. The ability to deploy large numbers of camera-equipped, semi-autonomous UAVs has numerous military and civilian applications including intelligence, surveillance, reconnaissance and search and recovery. One might imagine a number of inexpensive drones, carried as standard gear for field operations and released when needed by an infantry unit over a battlefield or into an occupied building, or by firefighters or by police or a bomb squad to gather situational intelligence. Various military branches use UAVs both for surveillance and as offensive weapons, although these historically have consisted of large, remotely-piloted weapons and surveillance platforms with individual drone costs approaching $1 million. The U.S. military has conducted research using swarms of small UAVs in at least a few instances.[1] The U.S. Army distributed drones to units of the 82^{nd} Airborne Division, for example, giving a soldier the ability to fly mini-drones.[2] Without explicitly mentioning mini or micro drones, the DOD has published plans calling for the development of more autonomous drones for particularly dangerous missions.[3,4] Small, off-the-shelf drones, often termed *"mini drones,"* contain stabilization circuitry, onboard cameras, and little else. They suffer restricted flying time and lack GPS systems. The limited capability of such drones has curtailed their use by researchers investigating practical swarm and path planning algorithms, and many researchers rely on computer simulation, rather than testing with actual drones. In this work, we train a YOLOv4-Tiny object detector to recognize Tello EDU drone video images using the Darknet neural network framework

Further author information: Both authors contributed equally.
Dr.Sayani Sarkar: E-mail: ssarkar@bellarmine.edu, Telephone: +1(337)806-5484
Dr.Nathan Johnson: E-mail: njohnson3@bellarmine.edu, Telephone: +1(502) 386-2224

Unmanned Systems Technology XXIV, edited by Hoa G. Nguyen, Paul L. Muench, Brian K. Skibba,
Proc. of SPIE Vol. 12124, 1212409 · © 2022 SPIE · 0277-786X · doi: 10.1117/12.2618137

and then take the first steps toward implementing swarm behavior in an individual drone. Mini drones lack the computational power to run an onboard neural network, and we avoid the issue by using a Python control module running on a laptop or similar system; we capture the wireless video feed from the drones and pass the frames to the detection algorithm, which then creates bounding boxes around another drone in the field of view. The control module sends commands back to the drone via WiFi.

In Section 2, we examine research in several areas pertinent to implementing swarms of drones. Section 3 outlines experimental design and methodology. Section 4 includes a discussion of experimental results and recognition accuracy. Finally, Section 5 presents some discussion and a path forward.

2. BACKGROUND AND MOTIVATION

At this point, practical commercial use of swarm algorithms in robots (aerial or otherwise) has been essentially non-existent.[5] Much of the important research into swarm behavior involves computer simulation rather than using actual physical drones. For example, Arnold et al. examine various configurations of heterogeneous drones to find survivors following natural disaster.[6] Swarms offer great potential in their promise of complex emergent behaviors even when the individual swarm members are incapable of any sophisticated calculation on their own. Any system that uses actual small, off the shelf drones to implement a swarm requires the following capability:[5]

- Swarm Navigation and Alignment software – To assemble, fly in a pattern, avoid collision and navigate or search the environment.
- Object Detection and Identification - Using a camera, neural network model to identify other drones and objects in the environment.
- A network and control framework – To process video data from the drone and send flight and other commands back to the drone swarm

The following sections examine some of the prior research regarding swarms, object detection using neural networks and recent efforts to control mini-UAVs remotely with software.

2.1 UAV Swarms

Swarm research often includes two general areas, which are sometimes conflated in the literature. The first involves coordinated movement, gathering, moving without colliding, maintaining correct position, and so on. The second involves navigation, path-finding and search. Much swarm research is based on the initial work by Reynolds[7] and examination of swarming and flocking behavior of various animals, including birds, fish and, in particular, bees and ants. Groups of these animals show emergent behaviors far beyond the capabilities of any one individual. Birds fly in beautiful, collision free formations, a necessity for a swarm of UAVs, that emerge when individual members of the flock follow a few simple rules related to the location of their nearby fellows. The most popular approach to swarm behavior and morphology advocates using a set of rules including separation, alignment and cohesion, based on Reynold's BOID mode.[7–10]

Bees and ants, for example, exhibit even more complex behavior, searching for, finding and converging on food sources – behavior which again emerges from simple individual behavioral rules – and which could have great practical application for swarms of UAVs searching buildings or other complex spaces. Many such search algorithms are based on behaviors present in biology, in the foraging behavior of bees and ants for example, and excellent surveys are presented in Senanayake et al.[11] and Chung et al.[12]

2.2 Object Detection and Identification

The fastest and most accurate methods for detecting an object in pictures or video frames use convolutional neural networks.[13–15] In particular, algorithms based on YOLO (You only look once) have proved popular.[16] Although detection accuracy isn't always better than other CNNs, the frame processing rate, required computational resources and public availability make it a logical choice.[17] YOLOv4 is now available[18] while YOLOv3[19] has previously been used to recognize objects in drone video. Current versions of YOLO use the Darknet neural network framework[20] written in C with CUDA libraries. The current version of YOLOv5 utilizes PyTorch.[21] In 2021, Madasamy et al. used YOLOv3 to recognize objects using drone cameras in a real time environment.[22]

2.3 Command and Control

At issue with very small drones, of course, is where and how to compute the object recognition, swarm and search algorithms, and where individual drone flight commands will originate. Mini drones are unlikely to be capable of the computation needed for a mission without extensive external direction. "Command and control" must be offloaded to a network-attached computer capable of executing the neural network and other algorithms in a timely manner.[23] At present there is no standard architecture for controlling a swarm of drones. Work on architecture, generally speaking, falls into two loose categories −− high level command architectures focused on overall mission planning and objectives, and lower-level communication and flight command protocols. Research at the Naval Postgraduate School proposes a taxonomy for military swarms as well as tactics for approaching and exiting target areas.[24] Abdelkader et al. discuss recent drone applications and propose a system architecture with central planning and distributed execution.[25] Purta et al. discuss using the Dynamic Data-Driven Application System (DDADAS)[26] to control a simulated swarm of UAVs. Communication and control architectures are reviewed by Campion et al.[27]

3. EXPERIMENTAL DESIGN AND METHODOLOGY

The budget-friendly, programmable DJI Ryze Tello EDU drone, was used to test our vision-based framework.[28] The Tello EDU weighs just 80 grams and occupies 98 × 92.5 × 41 cubic millimeters. Each drone costs about $140 U.S. and contains a 3-axis gyroscopic accelerometer, a magnetometer and an infrared altitude detection sensor. The drone has no GPS capability. The Tello Software Development Kit (SDK) allows two, 2.4 GHz network modes; an Access point (AP) mode and Station mode. In AP mode, the drone functions like a client to a wireless access point and in the Station mode the Tello acts as its own access point, accepting a connection from a computer or other device. Multiple drones may be connected to a single router using AP mode; unfortunately, Tello EDU drones are unable to send a video stream while in AP mode.

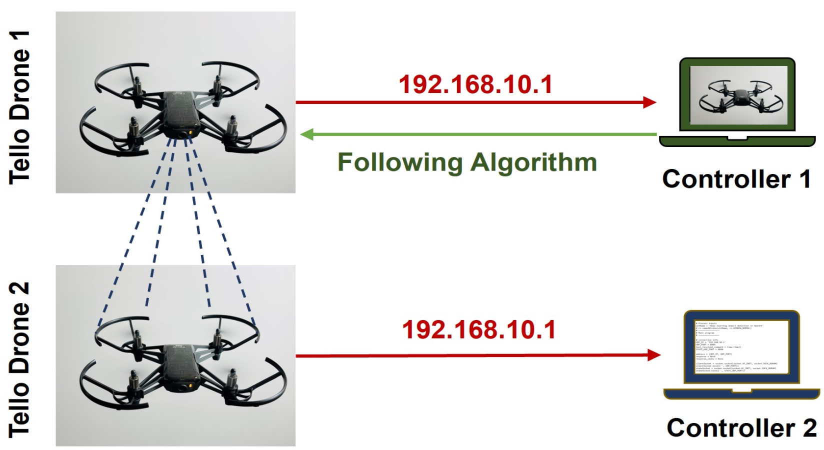


Figure 1. Controlling Tello drones in Station Mode.

The preferred vision based drone swarming framework would connect drones to a single router in AP mode. Because of the particular limitations of the Tello EDU drones regarding the video stream, each must connect in Station mode to a single controller, shown in Fig. 1. Each drone connects with a controller using a UDP socket with IP address "192.168.10.1" and port address 8889. After connection establishment, the drones used the YOLOV4-Tiny algorithm for target tracking and following. YOLOv4,[18] an improved version of YOLOv3, is a single stage object detector. YOLOv4-Tiny contains fewer convolutional layers than YOLOv4 and is ideal for mini drones using real-time processing. In Section 3.1, we discuss drone detection using the YOLOv4-Tiny algorithm; in 3.2, bounding box prediction; and in 3.3, drone movement based on bounding box tracking.

3.1 Drone detection YOLOv4-Tiny algorithm

YOLO divides the input image into $S \times S$ grids of equal sizes where each grid predicts the object located at the center of the grid. Each grid identifies object bounding boxes with an associated confidence score, and boxes with the best confidence scores can be used for flight planning. The 4^{th} generation of YOLO (YOLOv4) was released in the early 2020 and is used for drone flight planning. YOLOv4-Tiny uses CSPDarknet53-Tiny networks for object detection.[29]

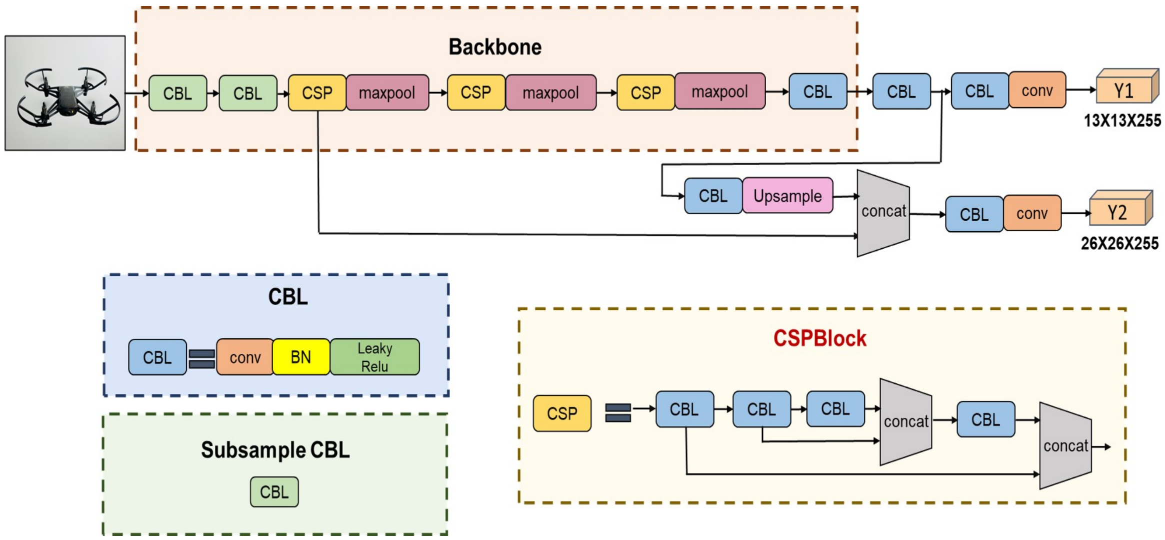


Figure 2. YOLOv4-Tiny network architecture based on CSPDarknet53-Tiny backbone and CSPBlocks.

The CSPDarknet53-Tiny has three convolution blocks and three CSPBlocks. Each convolution block consists of a convolution layer of kernel size 3, a down-sampled stride of 2, and a LeakyReLU activation function for simplified calculation. The network architecture of YOLOv4-Tiny is shown in Fig. 2. In the design, the CBL layer consists of a convolution layer, and a batch normalization function (BN), followed by LeakyReLU. In the CSPBlock, three CBL layers are connected back to back and concatenated with the previous CBL layer. The first two CBL layers are the sub sample of the main CBL layer. YOLOv4-Tiny uses two feature maps of size 13×13 and 26×26 to predict the bounding boxes based on the class names. In our model, we used a single class name called "TELLO."

3.2 Bounding box prediction

YOLOv4-Tiny uses complete intersection over union (CIoU) loss instead of Mean Square Error (MSE) loss for bounding box prediction.[30] The bounding box confidence score can be calculated by:

$$C_i^j = A_{i,j} * IoU \tag{1}$$

Where C_i^j is the confidence score at the i^{th} grid j^{th} bounding box. $A_{i,j}$ is an instance of the object. IoU is the prediction over union between the predicted bounding box and true box. The total loss L_{total} function of YOLOv4-Tiny contains a bounding box regression loss function ($Ł_{reg}$), a classification loss function ($Ł_{cls}$), and a confidence loss function ($Ł_{conf}$).

$$Ł_{total} = Ł_{reg} + Ł_{cls} + Ł_{conf} \tag{2}$$

$$Ł_{reg} = 1 - IoU + \frac{d^2(b, b^{gt})}{C^2} + \frac{\upsilon^2}{(1 - IoU) + \upsilon'} \tag{3}$$

The bounding box regression loss function can be expressed as:

$$\upsilon = \frac{4}{\pi^2}(tan^{-1}\frac{\omega^{gt}}{h^{gt}} - tan^{-1}\frac{\omega}{h})^2 \tag{4}$$

Where $d^2(b, b^{gt})$ is the Euclidean distance between the center points of the predicted bounding box (b) and the true bounding box (b^{gt}); C is the minimum diagonal distance of the box that can contain the predicted and true bounding boxes; ω^{gt} and h^{gt} are the width and height of the true bounding box; and ω and h are the width and height of the predicted bounding boxes. The classification loss function is:

$$\text{Ł}_{cls} = -\sum_{i=0}^{S\times S}\sum_{j=0}^{B} M_{i,j}^{obj} \sum_{c\in classes} [p_i(c)log(\hat{p}_i(c)) + (1 - p_i(c))log(1 - (\hat{p}_i(c)))] \tag{5}$$

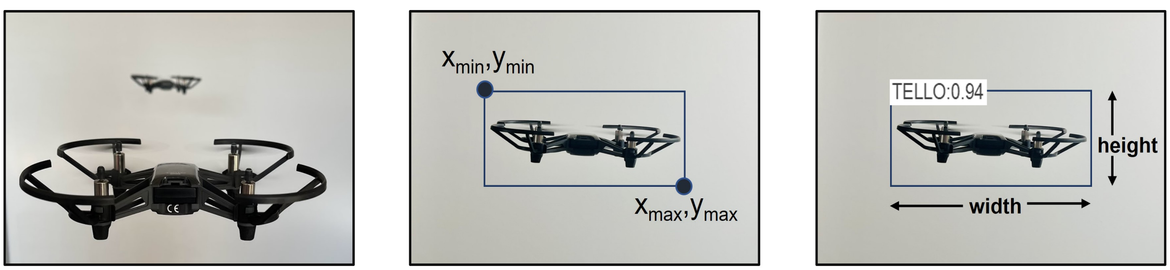
The confidence loss function can be expressed as:

$$\begin{aligned}\text{Ł}_{conf} = -\sum_{i=0}^{S\times S}\sum_{j=0}^{B} M_{i,j}^{obj} \sum_{c\in classes} [C_i log(\hat{C}_i) + (1 - C_i)log(1 - (\hat{C}_i))] - \\ \sum_{i=0}^{S\times S}\sum_{j=0}^{B} (1 - M_{i,j}^{obj}) \sum_{c\in classes} [C_i log(\hat{C}_i) + (1 - C_i)log(1 - (\hat{C}_i))]\end{aligned} \tag{6}$$

Where $S \times S$ is the grid size, and B is the number of bounding boxes in the grid. $M_{i,j}$ is the object function. The value of $M_{i,j}$=1 if the object located at i^{th} grid j^{th} bounding box, otherwise it is 0. $\hat{p}_i(c)$ is the predicted probability, and $p_i(c)$ is the true probability. $\hat{C}_i$ is the confidence score of the predicted box, and C_i is the confidence score of the true box. Bounding boxes with confidence level greater than 60% were used for drone flight planning.

3.3 Drone tracking and mission planning

A drone is able to identify the leading drone based on the bounding box provided by YOLOv4-Tiny. The distance between the drones is derived from the size of the bounding box relative to the image frame.



(a) (b) (c)

Figure 3. Drone vision based swarming. (a) Drone is following another drone using alignment rule, (b)Drone vision with bounding box coordinates, (c) Drone vision with detection confidence of the the drone ahead.

The drone is centered in the image frame and surrounded by the bounding box as shown in fig. 3(b). The change in relative bounding box size is used to determine the distance between the lead and following drone. The distance is expressed by,

$$distance = \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2)} \tag{7}$$

Where $x_{min}, y_{min}, x_{max}, y_{max}$ are the bounding box coordinates. This distance and a tolerance level based on the image frame height and width are then used as a reference in calculating the Proportional-Integral-Derivative(PID) control system to generate a control signal. The PID controller evaluates the drone sensor data and drives the drone accordingly. The PID control function can be derived by:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)d(t) + K_d \frac{\partial e(t)}{\partial t} \tag{8}$$

Table 1. Drone flying command with SDK 2.0.

Command	Degree of Freedom	Task planning
takeoff/ land	heav (z)	auto takeoff/ land
up/ down χ	heav (z)	ascend/ descend in range $20cm \leq \chi \leq 500cm$
left/ right χ	roll (ϕ)	fly left/right in range $20cm \leq \chi \leq 500cm$
forward/ back χ	pitch (θ)	fly forward/ backward in range $20cm \leq \chi \leq 500cm$
cw/ ccw χ	yaw (ψ)	rotate χ° clockwise/counter-clockwise where $\chi = 1{-}360^{\circ}$
streamon/ streamoff	5MP, 30fps, 82.6° view	enable/ disable video stream

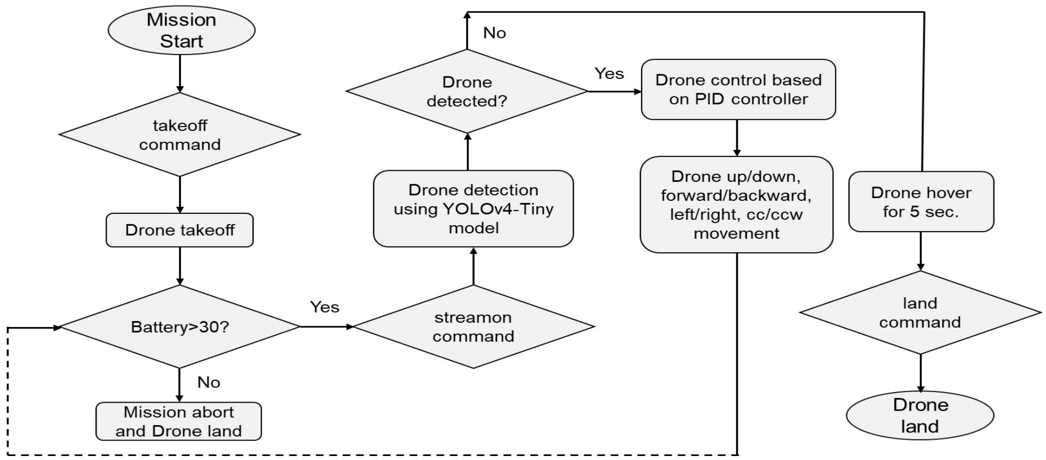


Figure 4. A general framework for Tello drone control using object detection.

Where $e(t)$ is the difference between s(t) and y(t). The s(t) value represents the set point and the y(t) value, the measured process value. The u(t) value is the control signal.

The drones are controlled based on the SDK commands listed in Table 1 and shown in Fig. 4. The mission is an iterative process of drone recognition using YOLOv4-Tiny and calculation of control signals used to control forward/backward, right/left, and up/down movements. Python libraries are used to connect to the drone, set up a video stream and parse the video into appropriate frames.

4. EXPERIMENTAL RESULTS

In this design, we used the Darknet deep learning framework to train a YOLOv4-Tiny model.[20] The model was trained on an NVIDIA GeForce RTX 3070 with 16 GB RAM and an Intel Core i7 CPU. The operating system was Windows 10 and included installations of OpenCV 3.3, CUDA 11.0 and CUDNN 8.3.

4.1 Dataset creation

In this work, we created a custom dataset, MINIUAV, including 500 images of Tello EDU drones. The images were resized 416×416. We used the LabelImg annotation tool[31] to create a single "TELLO" class. We applied data augmentation techniques like rotation, flipping and HSV shift to increase our data set from 500 to 1000 samples. The dataset is available on request.

4.2 Evaluation Criteria

We evaluated standard performance metrics for the MINIUAV dataset including precision, recall, F-1 score, and mAP (mean average precision) with a 25% confidence threshold and 50% IoU threshold. The formulae for performance metrics are shown below:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F_1 score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{11}$$

Where TP is the number of true positives, FP is the number of false positives, TN is the number of true negative cases and FN is the number of false negative instances.

Network	Interference time (ms)	Precision (%)	Recall (%)	$F_1 Score$(%)	mAP (%)
YOLOv3	180	86	82	88	87.43
YOLOv4	240	93	92	92	92.84
YOLOv4-Tiny	28	94	92	93	93.23

Table 2. Performance of YOLOv4-Tiny model

The MINIUAV dataset was divided into training (80%), testing (10%) and validation(10%) partitions. Performance metrics are shown in Table 2. To compare the performance of the proposed network with its YOLO counterparts, we trained and tested both YOLOv4 and YOLOv3 on the MINIUAV dataset and evaluated performance metrics for each. YOLOv4-Tiny has a comparable inference time with 93% accuracy compared to other models. The drone can detect and follow the leader drone at a distance 175 cm or less.

5. CONCLUSIONS AND FUTURE WORK

In this work, we have trained a YOLOv4-Tiny object detector and subsequently used it to analyze a drone video stream thereby allowing one drone to recognize another and follow. Our results show the object detector allows a drone to identify a neighboring drone in real time with 93% accuracy. In addition, we have shown that current object detection algorithms and drone hardware technologies are of sufficient maturity to implement simple swarm behavior.

Future work includes implementation of more complex swarm behaviors with multiple drones, path finding and navigation algorithms. Minor improvements in the capability of our drone, such as the ability to connect multiple drones to a single router, would substantially speed development efforts.

REFERENCES

[1] "Drone swarms." https://www.defensenews.com/video/2018/01/04/drone-swarm-test-in-california/. Accessed on 02.06.2022.

[2] "Army buys 9,000 mini-drones, rethinks ground robots - breaking defense breaking defense - defense industry news, analysis and commentary.." https://breakingdefense.com/2019/06/army-buys-9000-mini-drones-for-squads-rethinks-ground-robots-for-2020/. Accessed on 02.06.2022.

[3] Army, U., "Us army unmanned aircraft systems roadmap 2010-2035," *UAS Center of Excellence(ed.),(Alabama). Accessed* **27** (2010).

[4] Fahey, K. and Miller, M., "Unmanned systems integrated roadmap 2017–2042," *Department of Defense* (2017).

[5] Schranz, M., Umlauft, M., Sende, M., and Elmenreich, W., "Swarm robotic behaviors and current applications," *Frontiers in Robotics and AI* **7**, 36 (2020).

[6] Arnold, R., Mezzacappa, E., Jablonski, M., Abruzzo, B., and Jablonski, J., "Experimentation for optimization of heterogeneous drone swarm configurations: terrain and distribution," in [*Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*], **11746**, 1174625, International Society for Optics and Photonics (2021).

[7] Reynolds, C. W., "Flocks, herds and schools: A distributed behavioral model," in [*Proceedings of the 14th annual conference on Computer graphics and interactive techniques*], 25–34 (1987).

[8] Çintaş, E., Özyer, B., and Şimşek, E., "Vision-based moving uav tracking by another uav on low-cost hardware and a new ground control station," *IEEE Access* **8**, 194601–194611 (2020).

[9] Braga, R. G., da Silva, R. C., Ramos, A. C., and Mora-Camino, F., "Uav swarm control strategies: A case study for leak detection," in [*2017 18th International Conference on Advanced Robotics (ICAR)*], 173–178, IEEE (2017).

[10] Schranz, M. et al., "April 2020. swarm robotic behaviors and current applications," *Frontiers in Robotics and AI* **7**.

[11] Senanayake, M., Senthooran, I., Barca, J. C., Chung, H., Kamruzzaman, J., and Murshed, M., "Search and tracking algorithms for swarms of robots: A survey," *Robotics and Autonomous Systems* **75**, 422–434 (2016).

[12] Chung, S.-J., Paranjape, A. A., Dames, P., Shen, S., and Kumar, V., "A survey on aerial swarm robotics," *IEEE Transactions on Robotics* **34**(4), 837–855 (2018).

[13] Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S., "A survey of the recent architectures of deep convolutional neural networks," *Artificial intelligence review* **53**(8), 5455–5516 (2020).

[14] Sarkar, S., Totaro, M. W., Kumar, A., and Elgazzar, K., "Real-time object processing and routing for intelligent drones: A novel approach," *Computer* **53**(12), 80–89 (2020).

[15] Sarkar, S., Totaro, M. W., and Elgazzar, K., "Intelligent drone-based surveillance: application to parking lot monitoring and detection," in [*Unmanned Systems Technology XXI*], **11021**, 1102104, International Society for Optics and Photonics (2019).

[16] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You only look once: Unified, real-time object detection," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 779–788 (2016).

[17] Sahin, O. and Ozer, S., "Yolodrone: Improved yolo architecture for object detection in drone images," in [*2021 44th International Conference on Telecommunications and Signal Processing (TSP)*], 361–365, IEEE (2021).

[18] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M., "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934* (2020).

[19] Redmon, J. and Farhadi, A., "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767* (2018).

[20] Redmon, J., "Darknet: Open source neural networks in c." http://pjreddie.com/darknet/ (2013–2016).

[21] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems* **32** (2019).

[22] Madasamy, K., Shanmuganathan, V., Kandasamy, V., Lee, M. Y., and Thangadurai, M., "Osddy: embedded system-based object surveillance detection system with small drone using deep yolo," *EURASIP Journal on Image and Video Processing* **2021**(1), 1–14 (2021).

[23] Sarkar, S., Khare, S., Totaro, M. W., and Kumar, A., "A novel energy aware secure internet of drones design: Esiod," in [*IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*], 1–6 (2021).

[24] Giles, K. and Giammarco, K., "A mission-based architecture for swarm unmanned systems," *Systems Engineering* **22**(3), 271–281 (2019).

[25] Abdelkader, M., Güler, S., Jaleel, H., and Shamma, J. S., "Aerial swarms: Recent applications and challenges," *Current Robotics Reports* **2**(3), 309–320 (2021).

[26] Purta, R., Dobski, M., Jaworski, A., and Madey, G., "A testbed for investigating the uav swarm command and control problem using dddas," *Procedia Computer Science* **18**, 2018–2027 (2013).

[27] Campion, M., Ranganathan, P., and Faruque, S., "A review and future directions of uav swarm communication architectures.," in [*EIT*], 903–908 (2018).

[28] Ryze Tech, D., "Tello edu," *URL http://www.dji.com/product/tello* (2018).

[29] Xu, P., Li, Q., Zhang, B., Wu, F., Zhao, K., Du, X., Yang, C., and Zhong, R., "On-board real-time ship detection in hisea-1 sar images based on cfar and lightweight deep learning," *Remote Sensing* **13**(10), 1995 (2021).

[30] Jiang, Z., Zhao, L., Li, S., and Jia, Y., "Real-time object detection method based on improved yolov4-tiny," *arXiv preprint arXiv:2011.04244* (2020).

[31] Tzutalin, D., "Labelimg," *GitHub Repository* **6** (2015).