UAV Pursuit-Evasion using Deep Learning and Search Area Proposal*

Sébastien Arola[1] and Moulay A. Akhloufi[2]

***Abstract*— Unmanned Aerial Vehicles (UAVs) are very popular and increasingly used in different applications. UAVs collaboration and Pursuit-Evasion are among the current interests of the research community. In this work, we propose the use of deep learning for vision-based UAVs Pursuit-Evasion. The proposed approach uses images captured by a UAV and deep learning to detect and track another UAV. An approach based on the deep YOLO algorithm is used to detect the object of interest (target UAV). We propose the use of a search area proposal (SAP) algorithm which predicts the current UAV location based on previous detection history data. The proposed approach was able to process videos at 30 fps and get high mAP for UAV detection. Additionally, we developed a high-level control algorithm based on the use of the detected bounding box coordinates. The bounding box size and position help compute the commands to send to the follower UAV. Tests were conducted in outdoor scenarios using quadcopter UAVs. The obtained results and the high mAP are promising and show the possibility of using this kind of vision-based deep learning and search area proposal (SAP) approach for UAVs Pursuit-Evasion scenarios.**

I. INTRODUCTION

UAVs (Unmanned Aerial Vehicles) have known an increase in popularity in recent years and are now widely used in many applications. An area that was once mostly limited to the military is now expanding into commercial and industrial applications with hundreds of uses of drones emerging everyday.

Today, UAVs have become central to the functions of various industries and organizations. Many of these applications use individual UAVs. However, the use of multiple UAVs and UAV swarms is attracting more interest from the research community leading to the exploration of topics such as UAV collaboration, multi-drones autonomous navigation, UAVs Pursuit-Evasion, etc.

In this work, we are interested in multiple UAVs Pursuit-Evasion. The strategy can be collaborative or competitive between multiple UAVs. The goal here is to use deep learning, search area proposal (SAP) and the captured images from one of the UAVs to detect and track/follow the second moving UAV in real-time.

*We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference number RGPIN-2018-06233].

[1]Sébastien Arola is with the Perception, Robotics, and Intelligent Machines Research Group (PRIME), Dept. of Computer Science, Université de Moncton, Moncton, NB, Canada and with UPSSITECH, Université Toulouse III Paul Sabatier, Toulouse, France sebastien.arola@univ-tlse3.fr

[2]Moulay A. Akhloufi is with the Perception, Robotics, and Intelligent Machines Research Group (PRIME), Dept. of Computer Science, Université de Moncton, Moncton, NB, E1A3E9, Canada moulay.akhloufi@umoncton.ca

Recent work has been proposed in the area of collaborative UAVs. Schmuck *et al.* [1] propose the use of multiple small UAVs to act as agents and collaborate in a monocular simultaneous localization and mapping (SLAM) application. Vemprala *et al.* [2] propose a vision based collaborative localization framework for multiple micro aerial vehicles (MAV). They use the images captured by the cameras of the MAVs to estimate the pose. Individual and relative pose estimations are combined to localize against surrounding environments. The algorithm uses feature detection and tracking to estimate a six degree-of-freedom pose. They tested their algorithms using simulations within Microsoft AirSim [3]. Tian *et al.* [4] proposed the use of fleets of UAVs to search for lost persons. The UAVs explore collaboratively an area under thick forest canopies where the use of GPS is unreliable. They propose the use of onboard computation and wireless communication. A laser-range finder equips each UAV and is used to estimate its position, location and plan its path. A 3D map of the terrain is built and all the individual maps are fused on a ground station. With the built maps, it is possible to recognize unexplored and already-searched spots. Chung *et al.* [5] review aerial swarm robotics approaches and techniques that allow each member of the swarm to communicate and allocate tasks among themselves, plan their trajectories, and coordinate their flight to efficiently achieve the overall objectives of the swarm. The paper reviews state of the art theoretical tools focusing on those developed and applied to aerial swarms. They present results covering trajectory generation, task allocation, adversarial control, distributed sensing, monitoring, and mapping. Other work deals more with pursuit-evasion involving unmanned aerial vehicles. Alexopoulos *et al.* [6] propose a simulation of a two-player pursuit-evasion games with UAVs in a three-dimensional environment. They present a game theoretic framework to solve the pursuit problem using two identical quad-rotors. Krishnamoorthy *et al.* [7] use a UAV to search a target moving on a road. Ground sensors are used to trig the search by a UAV. In [8], Alexopoulos *et al.* propose a cooperative pursuit-evasion games with UAVs. They present a game-theoretical solution and show its applicability using multiple real outdoors flight experiments with a hex-rotor UAV. Fargeas *et al.* [9] analyze the problem of path planning for a team of UAVs performing surveillance and pursuit. The UAVs rely on communicating with ground sensors to detect potential intruders. They propose a heuristic algorithm to coordinate the UAVs during surveillance & pursuit, and show its effectiveness through simulations. Fu *et al.* [10] propose a ground target pursuit algorithm using fixed-wing UAVs. They propose a method which generates waypoints and direct the

UAV to the latest waypoint. They simulated three different scenarios to demonstrate the performance and efficiency of the proposed algorithm. In Camci *et al.* [11], the authors are interested in the control of quadcopters in pursuit-evasion scenarios. They propose the use of reinforcement learning to tune type-2 Takagi-Sugeno-Kang fuzzy logic controllers (TSK-FLCs). They benchmarked their approach in several scenarios and under different noisy conditions to show its performance.

In this work, we propose the use of vision-based deep learning object detection for the detection and tracking of a UAV in a Pursuit-Evasion scenario. The deep object detection approach uses vision data captured by a UAV to detect and follow another UAV. The algorithm is divided into two parts, the detection of the target UAV and the control of follower UAV navigation. The deep detection part uses historical detection data from a set of image sequences and inputs these data to a search area proposal (SAP) algorithm in order to locate the area with a higher probability of UAV presence. The following sections present the details of the proposed approach.

II. Deep Pursuit-Evasion Framework

The proposed deep Pursuit-Evasion framework uses images captured by a UAV and a deep learning network to detect and follow another UAV in a pursuit-evasion scenario. The detection coordinates are sent to a high level controller which translates each detection into its corresponding flight controls to keep the target UAV locked and close to the center of the image frame of the follower UAV. The proposed framework is presented in figure 1.

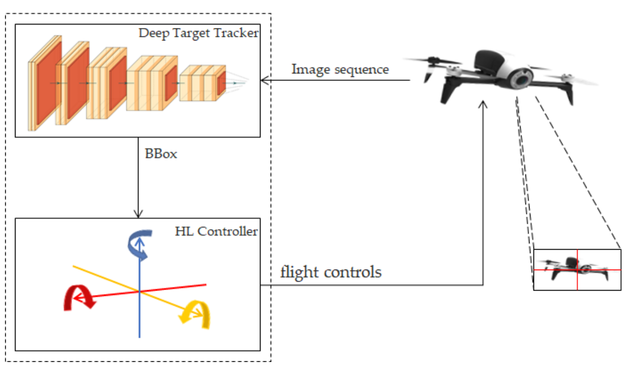

Fig. 1: Deep Pursuit-Evasion framework

A. Deep object detection of UAVs

To detect the target UAV we developed an approach based on the deep YOLO algorithm [12], [13]. The detection is performed with YOLO v2 [13], a deep convolutional neural network for object detection.

We use a UAV equipped with a camera to detect the target to track. The captured image is used to locate the target UAV and YOLO v2 [13] is used to detect the object of interest. YOLO v2 [13] is a popular 24 layers deep convolutional neural network built for object detection. YOLO v2 starts with images of size 224x224 pixels for the classifier training and then tune this classifier with 448x448 images. Anchor boxes are used and the object detector outputs a tensor with our class confidence. In this case we train it to recognize UAVs and non UAVs. This deep CNN shows very good results for our detection and recognition tasks. It is capable of a detection with more than 50% mAP score on a 30 fps video. Its ability to process video streams in real time is very interesting for our objective, since we needed a very fast detection to be able to estimate the target UAV position, compute the flight controls and send them to the follower UAV.

B. Prediction and tracking using a search area proposal (SAP)

In this work, we introduced the use of a search area proposal (SAP) to select the most interesting area for UAV detection based on historical detection data over time on a sequence of images as illustrated by figure 2. The developed search area proposal algorithm is based on particle filters, a popular approach used in computer vision target tracking [14], [15].

Particle filters are a set of algorithms used to solve a probabilistic filtering problem. The goal is to estimate the state of a dynamic system with only partial information and a noise applied to the sensor that captures this information. A set of particles are generated with a weight for each of them. The best particles are the ones that are the closest to the next state of the system. This way we can easily predict the next state of the system given previous states. These algorithms are very popular in object tracking applications when motion can undergo nonlinear trajectories. The term particle filter comes from statistics. The technique is also known as Monte Carlo Localization (MCL) in robotics [16] and as Condensation (Conditional Density Propagation) algorithm in computer vision [17]. A theoretical overview of particle filters can be found in [18].

The objective here is to combine particles filters with a deep CNN object detection to improve the detection performance and limits the candidates to specific areas of the image. As shown in figure 2, a sequence of images (stream) are fed to a search area proposal (SAP). The SAP uses previous detection data to predict the most probable position of the UAV in the current frame. Particle filters serve as a basis for our SAP algorithm. From a set of high probability particles we define a search area that encloses these best particles. This area is sent to a deep object detector, in our work we used YOLO v2, for the final detection of the object of interest. The resulting output is a bounding box and a confidence score for the detected object. The history of detection is kept to feedback the SAP for the next predictions. With these predictions computed by the SAP, we can reduce the search area (the region of interest around the predicted position). Using the proposed search area proposal YOLO (SAP-YOLO) leads to improvements in the performance of the target detection when it is far from the follower (above 5 meters). Without SAP, the deep CNN object detector (YOLO) struggles to detect the target when it is far away because the UAV size in the image is very small.

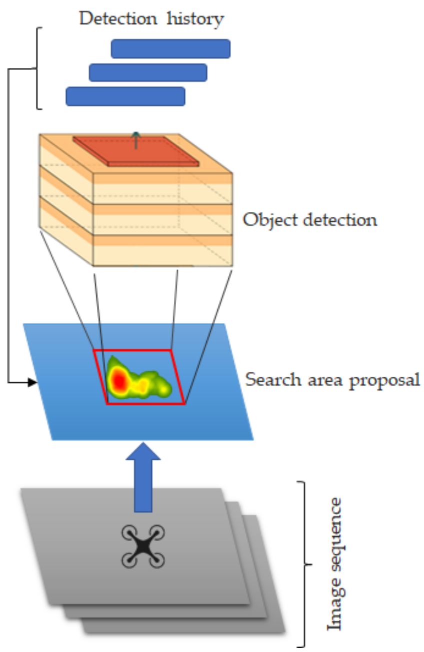

Fig. 2: YOLO with search area proposal (SAP-YOLO)

The extraction of the search area using SAP and its scaling to the input size of the deep CNN improves the detection accuracy. In figure 3, we can see the predicted area (small image in the left) where SAP estimated the UAV position. The green bounding box is the result of the detection in this area. The white bounding box is the box corresponding to the green box projected onto the whole frame.

Fig. 3: Example of a prediction given by SAP-YOLO

C. UAV control

When the follower UAV detects the target UAV, it needs to control its motion (follower) to track and follow it (target). We use the bounding box computer by the deep neural network to estimate the relative position of the target UAV and compute the necessary flight controls.

We build a visual servoing based on the obtained data. The controls sent to the UAV are meant to keep the target in the center of the image. With this high level controls we can follow the UAV when it flies in the following directions: right, left, up, down and forward. The UAV built-in controller takes care of the remaining adjustments to keep the trajectory and motion smooth. Figure 4 shows the center of the image (video frame), the target position and the vectors used to estimate the controls.

Du and Dv are the values we need to minimize to reduce the position error of equation (1):

$$E\left(t\right) = s\left(t\right) - s^{*}\left(t\right) \tag{1}$$

with $s\left(t\right)$ the target position at instant t and $s^{*}\left(t\right)$ the desired position at instant t (in our case, the image center).

$s\left(t\right)$ and $s^{*}\left(t\right)$ are defined by equations (2) and (3):

$$s\left(t\right) = \left(u_{center} + Du \;\; v_{center} + Dv\right)^{t} \tag{2}$$

and

$$s^{*}\left(t\right) = \left(u_{center} \;\; v_{center}\right)^{t} \tag{3}$$

Thus, we need to minimize the error defined by equation (4):

$$E\left(t\right) = \left(Du \;\; Dv\right)^{t} \tag{4}$$

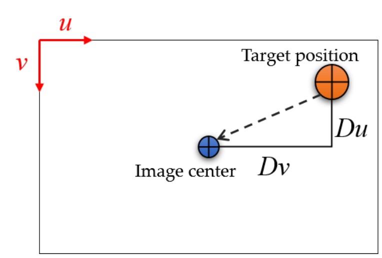

Fig. 4: Target UAV position representation within the image reference frame of the follower UAV

While the estimation of the 4 direction UAV controls (left, right, up and down) are straight forward, estimating forward motion when no 3D data is available can be challenging. To solve this problem, we developed a simple estimation based on the size of the UAV. The YOLO v2 deep convolutional neural network is good enough to estimate the closest bounding box to the target. The obtained coordinates give a good estimate of the size of the UAV. This size is compared with a reference size and when a scale is below a defined threshold, a move forward control is generated. This way the follower UAV moves forward to be close to the tracked UAV until the estimated target UAV size is within a defined tolerance. This approach requires an initialization step before the flight. For our experiments we use a reference size of the UAV positioned at 30 cm.

III. Dataset

To train the YOLO convolutional neural network to detect UAVs we need to develop a specific dataset of UAV images especially with horizontal UAV views (back view, side view and front view). We used two UAVs: Bebop 2 [19] and

Mambo [20] to capture multiple image sequences in indoor and outdoor conditions. Each frame was annotated with a bounding box that serves as a reference for training. Annotations follow a specific format. Various information are described in an XML format such as bounding box coordinates, associated file name, etc.

We created a first dataset of 6000 images of different sizes and tested the performance of object detection using YOLO v2. The majority of the first images were indoor. To refine the detection, we collected additional images mostly outdoor. 2500 new images were added to the previous dataset. The complete dataset contains 8500 images of two different UAVs taken at different angles and in different conditions (e.g. changes in outdoor illumination condition: sunny, cloudy, etc.)

IV. Experimental results

A. Object detection results

To test our solution we developed a prototype in python using keras [21] and tensorflow [22]. We trained YOLO v2 on our data and after 150 epochs of training on the dataset we achieved a mAP score of 0.96. This score shows the high performance of the proposed network for specifically detecting UAVs. Since we only deal with detecting UAVs, the number of classes is reduced, which explains the higher detection performance. The loss curve is given in figure 5. We can see that the loss decreases quickly and we obtain a low loss after only 80 epochs.

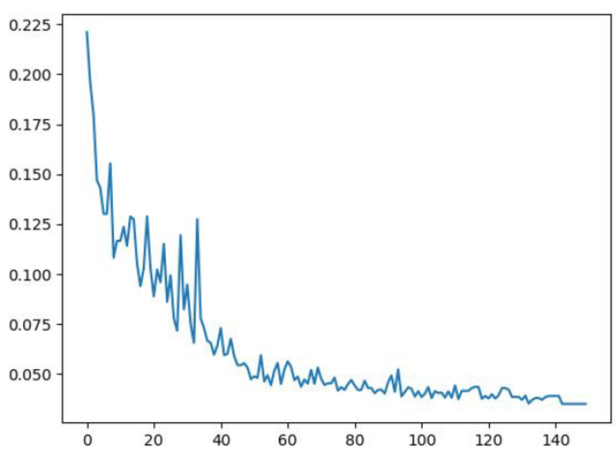

Fig. 5: Loss learning curve for object detection

To evaluate the performance of the detection we compute the Intersection Over Union (IOU) measure. Figure 6 shows the computation of the IOU. The intersection (green area) between the ground truth (green box) and the current detection (red box) is divided by the union of the ground truth and the detection (orange area).

The IOUs are computed on annotated videos not previously used for learning. Figure 7 shows one frame of the annotated test video and the bounding box of the detection.

Figure 8 shows the evolution of this IOU value during an outdoor video test. This video captures a target UAV

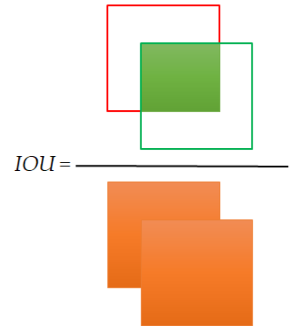

Fig. 6: Intersection Over Union computation

Fig. 7: UAV detection in outdoor tests

maneuvered manually, followed by a follower UAV which uses the deep object detection (YOLO v2) approach to detect and follow the first UAV. We can see that we obtain an IOU value between 0.7 and 0.91, which is more than acceptable as it means that the majority of the time more than 70% of the detected bounding box covers the reference bounding box. We can see that the IOU value is a bit lower in the middle of the graph, it corresponds to a UAV located farther and it is more difficult to detect.

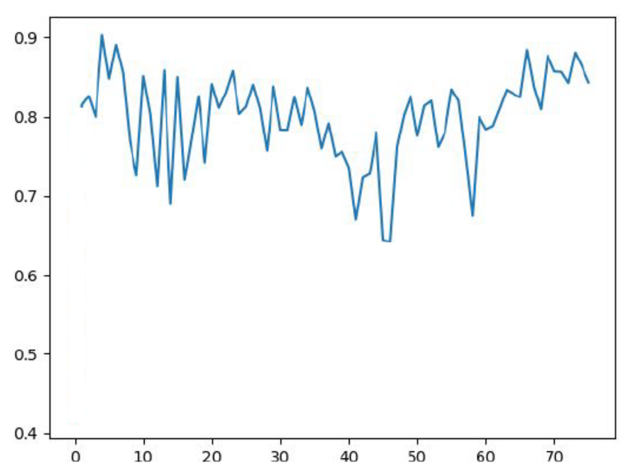

Fig. 8: Evolution of IOU measures during an outdoor test

We also conducted tests to evaluate the limits of the detection when the UAV is at a farther position. Three tests were carried out indoor at different distances: 1m, 4m and 6m. The confidence level for the detected class decreases with the increase in the distance (from 0.83 for 1m to 0.35 for 4m). These tests were made indoor with the UAV dropped on the floor so we could measure the distance with more precision (Figure 9).

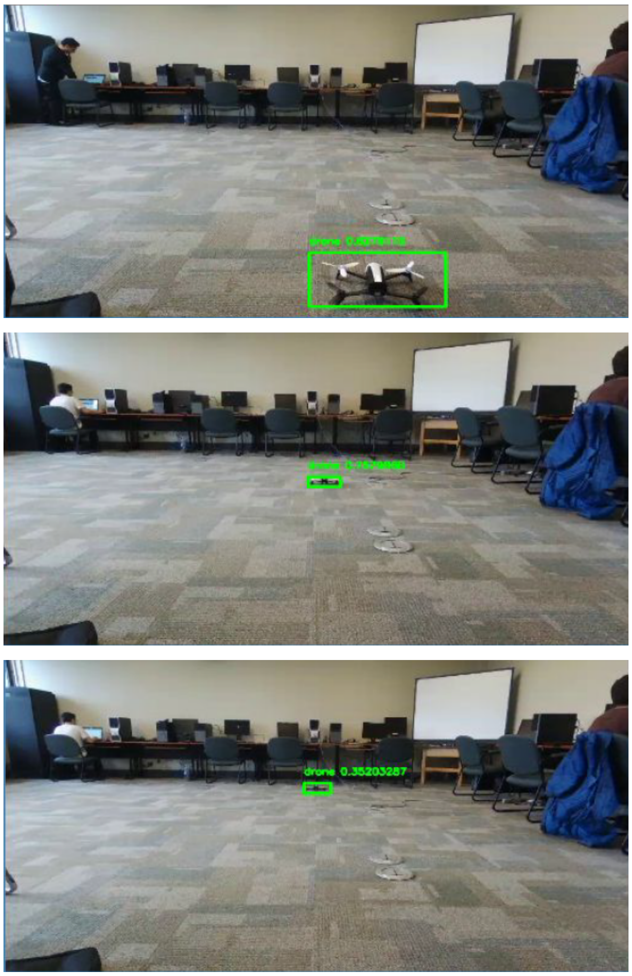

Fig. 9: Detection of the UAV at a distance of (up to down) 1m, 4m and 6m

B. UAV tracking with search area proposal (SAP)

Search area proposal (SAP) algorithm was used to predict the next search area and improve the detection. We tested it on the same annotated video above to compare the obtained IOUs with and without SAP. Figure 10 shows the evolution of the IOU measure over the video frames. We can see that when the UAV is close (start and end of the curve), the detection has a lower performance. This is because the UAV is closer and takes almost all the frame leading to a more difficult prediction of its next position. However, we can see that in the middle, when the UAV is farther, the IOU measure improves and get values between 0.8 and 0.9 compared to 0.7 without SAP. This shows that SAP is very useful when the target is small (far away and located at more than 4 meters in our experiments). However, the performance degrades when the target is at a closer distance (less than 2 meters). Between these distances the result is almost the same with and without a search area proposal (SAP).

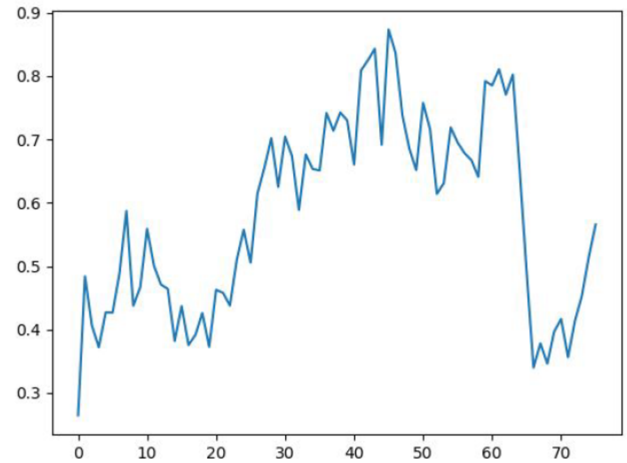

Fig. 10: Evolution of IOU measures with SAP-YOLO

V. CONCLUSION

In this work, we propose the use of deep learning to develop a vision-based UAVs Pursuit-Evasion.

A deep convolutional neural network (CNN) is used to detect objects of interest (UAV) and estimate the necessary controls for the follower UAV in order to keep the target UAV within its field of view and the closest possible to the center of the image frame. YOLO v2 was used as the UAV detector, since it was the best performing in complex outdoor conditions and faster enough to enable the processing at a rate of 30fps for a real-time tracking of the UAV. To follow the leader UAV, we developed a high-level control algorithm based on the use of the detected bounding box coordinates. The bounding box size and position help compute the command to send to the follower UAV among 4 directions (left, right, up, down) and a forward motion.

A search area proposal (SAP) approach was also developed to predict the most probable UAV location in the current video frame based on the history of detection data. This SAP algorithm combined with the deep object detection algorithm (SAP-YOLO) improved the detection of farther located UAVs.

Tests were conducted in outdoor scenarios using quadcopter UAVs. The obtained results and the high mAP are promising and show the possibility of using this kind of vision-based deep learning approach for UAVs Pursuit-Evasion scenarios.

Future work includes, improving the detection and prediction using the proposed search area proposal YOLO (SAP-YOLO) to enable the tracking of multiple-sized UAVs and optimizing the algorithms for faster processing.

References

[1] P. Schmuck and M. Chli, "Multi-uav collaborative monocular slam," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 3863–3870.

[2] S. Vemprala and S. Saripalli, "Monocular vision based collaborative localization for micro aerial vehicle swarms," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2018, pp. 315–323.

[3] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[4] Y. Tian, K. Liu, K. Ok, L. Tran, D. Allen, N. Roy, and J. P. How, "Search and rescue under the forest canopy using multiple UAS," in *Proc. International Symposium on Experimental Robotics (ISER)*, 2018.

[5] S. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, Aug 2018.

[6] A. Alexopoulos, T. Schmidt, and E. Badreddin, "A pursuit-evasion game between unmanned aerial vehicles," in *2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, vol. 02, Sep. 2014, pp. 74–81.

[7] K. Krishnamoorthy, D. Casbeer, and M. Pachter, "Minimum time uav pursuit of a moving ground target using partial information," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2015, pp. 204–208.

[8] A. Alexopoulos, B. Kirsch, and E. Badreddin, "Realization of pursuit-evasion games with unmanned aerial vehicles," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2017, pp. 797–805.

[9] J. C. L. Fargeas, P. T. Kabamba, and A. R. Girard, "Cooperative surveillance and pursuit using unmanned aerial vehicles and unattended ground sensors," *Sensors*, vol. 15, no. 1, pp. 1365–1388, 2015.

[10] X. Fu, H. Feng, and X. Gao, "UAV mobile ground target pursuit algorithm," *Journal of Intelligent & Robotic Systems*, vol. 68, no. 3, pp. 359–371, Dec 2012. [Online]. Available: https://doi.org/10.1007/s10846-012-9690-9

[11] E. Camci and E. Kayacan, "Game of drones: Uav pursuit-evasion game with type-2 fuzzy logic controllers tuned by reinforcement learning," in *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on*. IEEE, 7 2016, pp. 618–625.

[12] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 779–788. [Online]. Available: https://doi.org/10.1109/CVPR.2016.91

[13] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 6517–6525. [Online]. Available: https://doi.org/10.1109/CVPR.2017.690

[14] M. Akhloufi, A. Regent, and R. Ssosse, "3D target tracking using a pan and tilt stereovision system," in *Proc. of SPIE, Defense, Security, and Sensing, Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications X*, vol. 8713, 29 April - 3 May, 2013, pp. 8713–38.

[15] M. Akhloufi, "Pan and tilt real-time target tracking," in *Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications VII*, ser. SPIE, vol. 7668, Orlando, FL, USA, April 2010, pp. 7668–19.

[16] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, May 1999, pp. 1322–1328 vol.2.

[17] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, "Using the condensation algorithm for robust, vision-based mobile robot localization," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999, pp. 588–594.

[18] A. Doucet, "On sequential simulation-based methods for bayesian filtering," Signal Processing Group, Department of Engineering, University of Cambridge, Tech. Rep., 1998.

[19] Parrot SA, "Bebop 2," https://www.parrot.com/us/drones/parrot-bebop-2parrot-bebop-2-details, online; accessed March 2019.

[20] ——, "Mambo FPV," https://www.parrot.com/ca/drones/parrot-mambo-fpv, online; accessed March 2019.

[21] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/