An Image-based Classification Module for Building a Data Fusion Anti-drone System

Edmond Jajaga[1][0000-0003-1833-5856], Veton Rushiti[1][0000-0002-9396-1163], Blerant Ramadani[1][0000-0002-4453-9644], Daniel Pavleski[1], Alessandro Cantelli-Forti[2][0000-0002-6943-2632], Biljana Stojkovska[3][0000-0003-4435-2676] and Olivera Petrovska[1][0000-0003-3065-9853]

[1] Mother Teresa University, Mirçe Acev nr. 4, 1000 Skopje, North Macedonia
{name.surname}@unt.edu.mk
[2] Lab RaSS CNIT, Pisa, 56124 Pisa
alessandro.cantelli.forti@cnit.it
[3] Faculty of Computer Science and Engineering, Intelligent Systems Department, Ss. Cyril and Methodius University, Rugjer Boshkovikj 16, 1000 Skopje, North Macedonia
biljana.stojkoska@finki.ukim.mk

Abstract. Means of air attack are pervasive in all modern armed conflict or terrorist action. We present the results of a NATO-SPS project that aims to fuse data from a network of optical sensors and low-probability-of-intercept mini radars. The requirements of the image-based module aim to differentiate between birds and drones, then between different kind of drones: copters, fixed wings, and finally the presence or not of payload. In this paper, we outline the experimental results of the deep learning model for differentiating drones from birds. Based on the trade-off between speed and accuracy, the YOLO v4 was chosen. A dataset refine process for YOLO-based approaches is proposed. The experimental results verify that such an approach provide a reliable source for situational awareness in a data fusion platform. However, the analysis indicates the necessity of enriching the dataset with more images with complex backgrounds as well as different target sizes.

Keywords: anti-drone system, deep learning, YOLO, data fusion

1 Introduction

The human's multisensory system has been extensively studied in order to provide more accurate and more efficient machine decisions. This process includes integration of multi-source data and is called data fusion. In processing perspective, data fusion represents an area which includes a combination of batch and stream processing features. Namely, in data fusion systems, data is collected over time from continuous data streams and follows with continuous processing of a bunch of data. Thus, the system requires fast and lengthy performance. In situational awareness perspective, a data fusion system achieves refined position, identifies estimates and complete and timely assessments of situations, threats and their significance [1]. The final goal of using data fusion in multisensory environments is to obtain a lower detection error probability and higher reliability by using data from multiple distributed sources [2]. The same goal applies also for the deep learning (DL) approaches, which utilize convolutional neural networks (CNNs) for object detection and recognition. CNNs tend to look for meaningful features that can help to classify the images or, in the case of object detection, to draw the boundary boxes enclosing the target of interest [3].

Data fusion systems are especially important for the domain of Means of Air Attack (MoAA). One of the most developing MoAA category are the Unmanned Aerial Vehicles (UAVs) i.e., “drones”. Killer drones represent a real threat to people’s life and health. For example, just recently a drone of unknown origin crashed near Zagreb (Croatia) by flying undetected on a number of states. Fortunately no-one was injured. In order to facilitate the neutralization of killer-drones and minimize the risk for people and assets, a NATO SPS Anti-Drones project[1] has been focalized on the development of a new concept of an anti-drone system able to detect, recognize and track killer-drones. The project scope is to progress the state of the art exploiting mini-radar technology and signal processing, data processing and fusion subsystem, for improving the performance and eliminating the environmental impact (e.g., ECM pollution) in an urban environment. The system infrastructure includes a network of LPI (Low-probability-of-intercept) mini-radar with FMCW or noise-like waveform, and on-demand, fully digital, optical camera-integrated imaging capability, capable of working in all weather conditions, to be deployed and appropriately placed on the ground in the area of the asset to be protected. The optical part is essential to support correct classification and therefore identification of the threat and thus to eliminate false alarms.

To the best of our knowledge, this is the first attempt that proved data fusion by integrating image data with radar ones for differentiating drones from birds. This paper covers the optical subsystem and automatic recognition, in particular the ability to distinguish drones from birds and is organized as follows. Section 2 gives an overview of the system design. Section 3 describe details for the dataset generation methodology of the proposed approach. Section 4 examines the experimental results. In Section 5, relevant related work from the literature is presented. Finally, the paper is concluded in Section 6, with directions for future work.

2 System design

One of the main challenges of our system is establishing an efficient data fusion algorithm. Data fusion takes action in different levels of our system. In a higher-level perspective, as depicted in Fig. 1, the system should fuse together radar and camera data. We follow a similar approach to Liu et al. [4]. However, instead of integrating camera and acoustic data, our solution will combine Support Vector Machine (SVM) radar data (direction of arrival, range, angular coordinates, elevation and radar cross section) with You Only Look Once (YOLO) camera-based images. In lower levels, the data fusion takes place only within the modules of the corresponding data source. As per the optical

[1] https://antidrones-project.org/, last access 24.03.2022

part, the images provided by the camera are processed by DL methods to support the data fusion algorithm with additional confidence score for radar-detected targets.

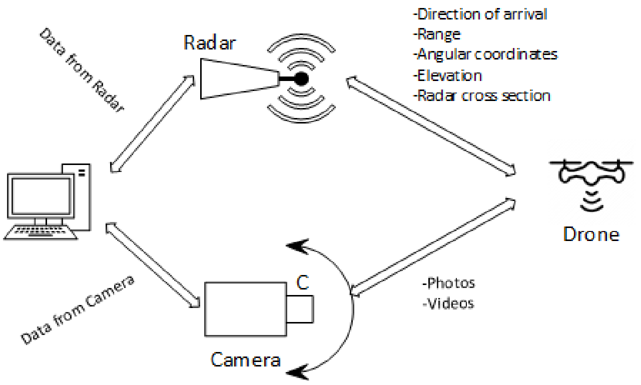

Fig. 1. Processing flow of drone detection

During the phase of literature review for object recognition approaches, a number of different DL approaches were considered. Namely, state-of-the-art ML frameworks, including: YOLO, TensorFlow and PyTorch, were examined. YOLO framework was chosen based on the project objectives, high accuracy and ability to detect objects in real-time by processing 67 FPS [5]. Moreover, it's power efficient compared to other DL detectors [6], open source, flexible network architecture, low hardware requirements i.e., minimum 4GB memory and is able to detect relatively small objects. The YOLO architecture model is mainly based on Darknet [7], which typically consists of 19 convolutional layers and 5 pooling layers.

3 Dataset Generation Methodology

Although measurement campaigns to verify the quality of the mini-radar have provided some static and dynamic optical images of drones also equipped with synthetic payloads, unfortunately, to date there is a lack of existing drones' dataset [2], [8]. It's even harder to have sufficient number of images of drones with payload. Furthermore, a very sensitive issue represents the quality of the images in terms of drone or bird size and positions, as well as the background characteristics. For this purpose, the researchers have considered different drone dataset generation techniques, e.g., the randomization method described in [9] or combining background-subtracted real images as described in [8]. Our focus was rather on building a methodology for more qualitative dataset.

The number of classes to be recognized by the model should also be considered, because it reflects on the performance of the model. Thus, in line with radar-based recognition fusion, we plan to consider five classes on the optical side, including: drone, bird, fixed-wing, copter and drone with payload. Following the lack of images and the purposes of the challenge, we have decided to firstly enrich the dataset for the first two classes (drone and bird), then to follow up with the next two classes (fixed-wing and copter) and finally detect and classify drones with payload.

To ensure better recognition results, the following criteria were considered during dataset selection:

- different types of drones: copters and fixed wing ones, as well as different models including: DJI Phantom, Inspire, Mavic, RTK 300, and Matrice,
- multiple drones shown in different positions and distances,
- different backgrounds and sizes of drones, and
- a number of fixed wing drones, which are currently classified as drones.

As per our dataset a number of open-source datasets were considered, including DroneNet [dronenet], Drone vs Bird [3], Skagen and Klim [skagen-klim] and other free web images. From DroneNet [dronenet] and Skagen and Klim [skagen-klim] datasets, 2395 and 1709 images are used, respectively. The annotations on these datasets are already in YOLO format. Around 200 of images found on the web were manually annotated using LabelImg[2], a graphical image annotation tool. The largest dataset that was used for our approach is Drone vs Bird [3] one.

In summary, the dataset contains a total of 14 549 images, consisting of 12 370 images with drones only (including 3 261 with only fixed-wings), 1 857 only birds and 322 images with annotated drones and birds. Regarding the size of the targets, they mainly fall between the sizes of 16^2 and 48^2, and over 96^2 (see Fig. 2).

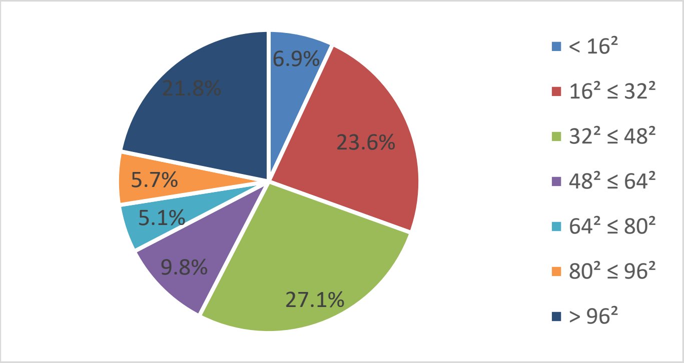

Fig. 2. Distribution of target sizes based on the annotations in the train and test data

As previously mentioned, a well-defined process of building the dataset was iteratively performed by the team. Namely, in order to match the YOLO format, a number of pre-processing steps were iteratively performed on each dataset, as depicted in Fig. 2. Each step is described in detail in the following subsections.

[2] https://github.com/tzutalin/labelImg, last access 19.03.2022

Image extraction and selection. If the dataset contained video, then the step of image extraction per frame was performed. For this purpose, *Free Video to JPG Converter* application[3] was utilized. The tool supports customized extraction of images per frame and per second. The images were extracted frame-by-frame. The image filename was constructed in the following format "`<video_filename> <frame_number>`". The image selection was done by human intervention manually i.e., by removing unimportant images, which were selected based on the following criteria:

- images without drone or bird, or
- the target being so small that it causes confusion to the prediction model, or
- the body of the target object being mostly behind another visible object.

The image selection step was also performed for image-based datasets.

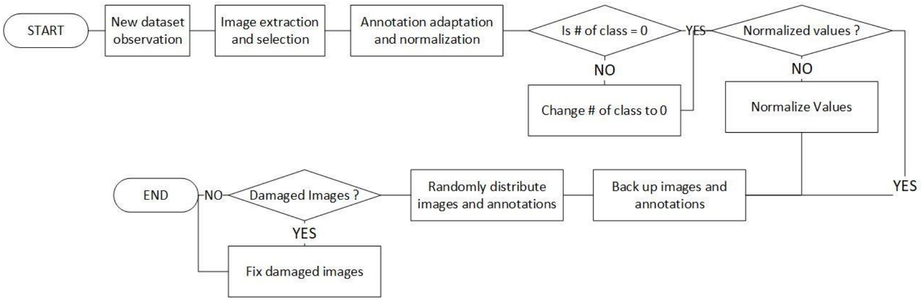

Fig. 3. The dataset refine process for using on YOLO-based models

Annotation adaptation and normalization. As YOLO framework requires, for each image, a single annotation file, in the next step the annotation adaptation task took place. Specifically, for this task a simple desktop application was developed. The application supports the following steps in order:

1. Load the folder images. Browse for the folder containing images (extracted from videos in the previous step).
2. Set annotated text file. Browse and open the annotation file containing the annotation list for every frame of the video. The general format and an example annotation of the Drone vs Bird dataset format consists like in Table 1.

Table 1. Input annotation's format.

Format	`framenum num_objs_in_frame obj1_x obj1_y obj1_w obj1_h obj1_class`
Example	`34 1 1 241 55 43 drone`

[3] https://www.dvdvideosoft.com/products/dvd/Free-Video-to-JPG-Converter.htm, last access 19.03.2022

The annotation adaptation task outputs a text file for each image with a filename the same as the picture with the format described in Table 2. For multiple objects present in a single frame multiple rows were appended to the text file.

Table 2. YOLO annotation format.

Format	`obj1_classnumber obj1_x obj1_y obj1_w obj1_h`
Example	`0 1 241 55 43`

Since we are dealing with a huge number of files a validation function is needed to check for valid pairs of image files with corresponding annotation files. For this purpose, the output folder files generated from the last step, consisting of a list of couples (photo and text files), are firstly loaded and then get checked for invalid couples.

Furthermore, YOLO expects object annotations to be in the normalized format. For this purpose, a simple conversion tool was also developed to check and normalize the annotations.

Image reduction. Since the images on successive frames are very similar and as such add little information to the model, we decided to reduce the dataset by removing every third image of each video file. This task was performed manually by using Windows Explorer feature to arrange files three per each row followed up by selecting the third column and removing files.

Train and test images distribution. Finally, as per our solution the dataset should be organized into train and test folder, based on the specified 75-25 percentage ratio. A Power Shell (PS) script was utilized to support this feature.

Fix object class, annotations and images. In different versions of our dataset the class of drone and bird was interchangeably set as 1 and 0. To ensure class consistency a PS script was executed on the dataset folders.

Furthermore, a simple tool was developed to also fix some conversion inconsistencies within annotation lines. In fact, double spaces and commas were replaced with single space.

Since our dataset was placed on open repositories, such as Google Drive, a small number of images got damaged after the process of distributing them into train and test folder. For this reason, before each training process, all the images were scanned for defects with the open-source tool Bad Peggy[4].

The final step before a training session was the backup of files. Namely, both train and test folder were occasionally backed up. Sometimes, the backup of files was performed before distribution of files into train and test folders.

As per the experimental set up, the default configurations of YOLOv4 model, based on Darknet [7], were utilized. Some Darknet code was modified and compiled to

[4] https://github.com/coderslagoon/BadPeggy, last access 19.03.2022

support the correct output format of the Challenge. The Google Colab Pro[5] platform was used for the training and validation of the model. It supports faster GPUs, more memory and longer runtimes as specified in the free version. With Colab one can import the image dataset, train the image classifier and evaluate the model.

Following the iterative process of improving the dataset, our model was trained with its different versions. During the last training around 6000 iterations were made and the whole training process lasted about 8 hours. The training resulted with mAP of 68%.

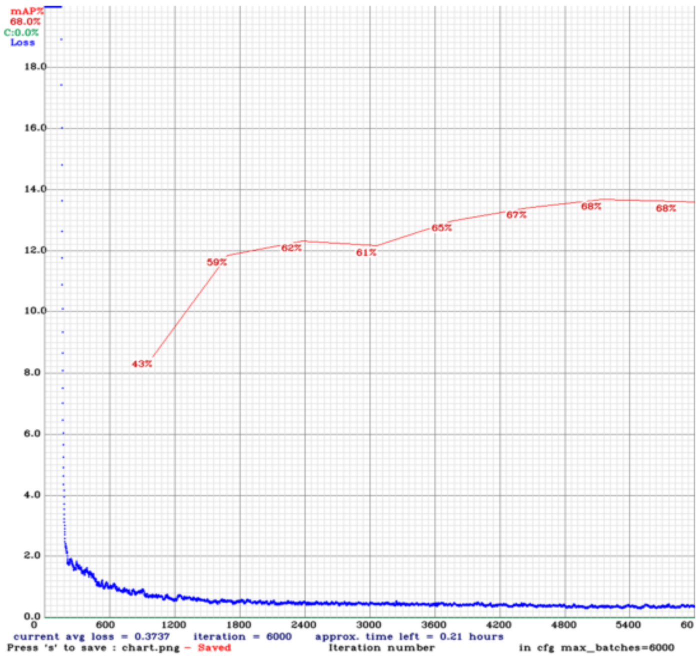

Fig. 4. mAP of the training performance of the dataset

4 Experimental Results

Experimental results are evaluated for drone detections and classification by discarding bird detections. In order to describe more correctly the prediction accuracy of our model, the *Precision* metrics are analyzed. Additionally, to better describe the detections of the proposed approach, the *Recall* metrics are utilized.

In order to get a broader perception of the performance of the described approach, a number of test sequences were selected to include the following characteristics: many static objects, complex backgrounds, different target sizes, moving camera and near/far targets. The videos were chosen from the Drone vs. Bird Challenge dataset as they best address these constraints, and also provide ground truth annotations of drone objects.

[5] https://colab.research.google.com/, last access 20.03.2022

Table 3 lists the selected test sequences with corresponding characteristics, the number of present ground truth objects (#GT) as compared to the number of submitted detections (#Det), resulting recall (#Rec) and precision (#Prec).

Table 3. Description of the test video sequence set with comparison of detection results in terms of number of submitted detections and resulting recall.

Sequence	Characteristics	#GT	#Det	#Rec	#Prec
dji_matrice_210_sky	moving cam; multi-rotor drone; clear sky view; short length;	1318	1470	99.77	92.41
dji_mavick_close_buildings	moving cam; multi-rotor drone; non-sky view; long length;	1501	1034	66.60	100.00
dji_phantom_landing_custom_fixed_takeoff	moving cam; multi-rotor drone; cloudy sky; short length;	2613	2606	92.10	99.63
parrot_disco_zoomin_zoomout	moving cam; fixed-wing drone; clear sky view; short length;	665	252	50.89	81.43

The first test and third sequence (dji_matrice_210_sky and dji_phantom_landing_custom_fixed_takeoff) are not very challenging for the proposed model, because they have a sky view and thus the retrieved results are near ideal. Based on the trained model, which contains a high number of images with clear and cloudy sky view, the high score of recall and precision has turned out as expected. Namely, a static street light has generated a number of false alarms in the first sequence, which has reduced the precision to 92.41%. Unlike this, the performance of the third sequence has resulted with better FPs and thus 99.63% precision, but with greater FNs i.e., recall of 92.10%.

The view of the second sequence (dji_mavick_close_buildings) represents a drone moving on land background. The duration of the sequence is lengthy and the drone appears in every frame, which has resulted with more FNs. Thus, the recall has dropped to 66.60%, which means 1/3 of GTs are missed. However, the precision has remained perfect, because there were no FPs.

A fixed-wing drone is demonstrated in the fourth sequence (parrot_disco_zoomin_zoomout). Following the clear sky, the resulting precision is perfect. Similar to the second sequence, the recall is again decreased by missing a half of the GTs.

5 Related Work

For getting a better insight about anti-drone YOLO-based approaches, a number of state-of-the-art ones were analyzed. Namely, the following approaches were analyzed:

- Aker et al. [8] describe an end-to-end object detection method to predict the location of the drone in the video frames. The scarce data problem for training the network has been solved by an algorithm for creating an extensive artificial dataset.
- In [10], authors describe an autonomous UAV detection and tracking platform. Namely, a Tiny YOLO detector is integrated into a hunter drone for detecting and chasing another drone.
- Wu et al. [2] propose a video-based detection of drones. To support their approach, they have developed a dataset consisting of 49 videos.
- A combined multi-frame DL detection technique, where the frame coming from the zoomed camera on the turret is overlaid on the wide-angle static camera's frame, is described in [11].
- Lei and Huang [6] have proposed a solution for detecting fixed-wing intruders with YOLOv3.

Each implementation has its own pros and cons. In general, none of the described approaches consider fusing optical data with radar ones. Moreover, even though we currently recognize birds and drones, our dataset next versions will further recognize drones with payload, which is not the case in the approaches. In particular, the solution presented in [8], detects the only drone in the scene and problems occur when the network mixes up a bird with the drone. The rest of the approaches are limited to a single drone class, except [11] who include other classes like: airplane, bird and background. But it does not provide further details about the dataset. The approaches were analyzed on the following different aspects of particular interest.

Network architecture. The YOLO architecture model is mainly based on Darknet [7], which typically consists of 19 convolutional layers and 5 pooling layers. In general, each approach has applied specific fine-tuning techniques for achieving better performance. For example, the classifier model used in [11] uses 64 x 64 size of the input layer, while vector classification is performed by 2 consecutive fully connected layer with 512 neurons with 0.5 dropout between them [11]. To raise the performance of our approach, we are considering the network modifications in future works.

Dataset. The dataset quantity and quality differ in the approaches, as well as image sizes. In particular, Aker et al. dataset, consisting of 676 534 images with 850 x 480 resolution, combines real drone and bird images with different background videos. Wyder et al. use a synthetically generated dataset[6] of 10,000 images from autonomous drone flying sequences, manually annotated. The same number of images has been generated and used by Lei and Huang for their solution. Our dataset consists of lesser number of images, as we strive to build a high-quality dataset. The proportion of the training versus validation dataset typically ranges between 70-80% and 30%-20%, respectively.

An artificial dataset generation algorithm is described in Aker et al. It describes the process of generation and reduction of the images. However, it does not include the process of image annotation extraction and conversion, as well as image checking for errors.

[6] https://osf.io/jqmk2/, last access 18.03.2021

Annotations. The annotations typically include information about coordinates of the center of the boxes with respect to the grid cell, the width and height in proportion to the whole image, and a confidence score of the detected object within the bounding box. In general, the approaches utilize the existing dataset annotations, or as in our case utilizing several parts from different datasets, and enriching them with new manually labeled and annotated images. Wu et al. have used Kernelized Correlation Filters (KCF) tracker [12] to auto label detected objects. A study about different types of annotation errors examined in a YOLO-based detector is described in [13]. In our approach we have used manual annotation as well as converting to YOLO-based format.

Classes. For better performance results the approaches have mainly considered a single class, i.e., drone, as specified in their model. However, Aker et al. use two classes drones and birds, while Unlu et al. have used four classes in their solution. As previously mentioned, for this paper we have used two classes and will use other ones for differentiating between drone models and carrying or not a payload.

Accuracy. The precision and recall of the approaches are satisfactory, with more than 89% and 85%, respectively. Our approach has resulted with more than 92% and 50% precision and recall, respectively. The autonomous Tiny YOLO-based approach [10] has performed with 77% accuracy in cluttered environments in eight frames per second.

6 Conclusion

Fast and robust detection and recognition is required for the anti-drone domain, because drones have ability to fly with high speed and for a short time can cause huge damage to human lives. A lot of research efforts has been dedicated by the image processing community. In particular, the findings of this paper suggest that a methodological approach should be well-defined for the dataset improvement lifecycle. The iterative process includes continuous check, validation and image variety of the dataset. Instead of infusing vast number of images into the dataset, which can cause model confusion, the dataset improvement process should ensure high quality images, which on the other side can lead to a reduced number of false alarms and missed detections.

To date, there is not enough evidence of approaches for detecting killer drones in far distances by combining different data sources. Namely, as can be observed by the results of this paper and based on the described related works, we can conclude that the image processing algorithms do not perform well enough in cases when the background of the view is complex and the distance of the drone is far. For this aim, as per future work of our approach, we propose that the drone detection and recognition should include other technology (i.e., radar RF) and data fusion techniques complemented with optical-based recognition. This will support higher system accuracy and reliability by eliminating the identified obstacles.

Aknowledgments. This work was funded by NATO SPS Programme, approved by Dr. A, Missiroli on 12 June, 2019, ESC(2019)0178, Grant Number SPS.MYP G5633. Total project Grant is Euro 398.000. Total project duration is 36 months, project kick-off date: 25 September 2019. NATO country Project Director Dr. A. Cantelli-Forti, co-director Dr. O. Petrovska, and Dr. I. Kurmashev. We thank Dr. Claudio Palestini, officer at NATO who oversees the project. We thank Prof. Biljana Stojkoska from the Ss. Cyril and Methodius University, for her scientific contribution and tutoring to the authors (Prof. Biljana is not a member of the project and had no access to the instrumental data). We thank author O. Petrovska for funding acquisition.

References

1. White, Frank.: Data Fusion Lexicon. San Diego, Calif, USA, Code 420, (1991).
2. Wu, M., Xie, W., Shi, X., Shao, P., &amp; Shi, Z.: Real-time drone detection using deep learning approach. In International Conference on Machine Learning and Intelligent Communications, pp. 22-32. Springer, Cham. (2018).
3. Coluccia A, Fascista A, Schumann A, Sommer L, Dimou A, Zarpalas D, Méndez M, de la Iglesia D, González I, Mercier J-P, Gagné G, Mitra A, Rajashekar S.: Drone vs. Bird Detection: Deep Learning Algorithms and Results from a Grand Challenge. Sensors,21(8):2824, (2021).
4. Liu, H., Wei, Z., Chen, Y., Pan, J., Lin, L., &amp; Ren, Y.: Drone detection based on an audio-assisted camera array. In: IEEE Third International Conference on Multimedia Big Data (BigMM), pp. 402-406. IEEE. (2017)
5. Bochkovskiy, A., Wang, C. Y., &amp; Liao, H. Y. M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020).
6. Lai, Y. C., &amp; Huang, Z. Y.: Detection of a Moving UAV Based on Deep Learning-Based Distance Estimation. Remote Sensing, 12(18), 3035. (2020).
7. Redmon, J., &amp; Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263- 7271. (2017).
8. Aker, C., & Kalkan, S.: Using deep networks for drone detection. In: 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1-6. IEEE (2017)
9. Marez, Diego., Samuel, Borden., and Lena, Nans.: UAV detection with a dataset augmented by domain randomization. In: International Society for Optics and Photonics, Geospatial Informatics *X*, vol. 11398, p. 1139807, (2020).
10. Wyder, P. M., Chen, Y. S., Lasrado, A. J., Pelles, R. J., Kwiatkowski, R., Comas, E. O., ... &amp; Lipson, H.: Autonomous drone hunter operating by deep learning and all- onboard computations in GPS-denied environments. PloS one, 14(11), e0225092, (2019)
11. Unlu, E., Zenou, E., Riviere, N., & Dupouy, P. E.: Deep learning-based strategies for the detection and tracking of drones using several cameras. IPSJ Transactions on Computer Vision and Applications, 11(1), 1-13 (2019).
12. Henriques, J. F., Caseiro, R., Martins, P., & Batista, J.: High-speed tracking with kernelized correlation filters. IEEE transactions on pattern analysis and machine intelligence, 37(3), 583-596, (2014).
13. Koksal, A., Ince, K. G., & Alatan, A.: Effect of annotation errors on drone detection with YOLOv3. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 1030-1031, (2020).