Drone Detection & Classification with Surveillance ‘Radar On-The-Move’ and YOLO

Haifawi, Hani; Fioranelli, Francesco; Yarovoy, Alexander; van der Meer, Rob

DOI
10.1109/RadarConf2351548.2023.10149588

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 IEEE Radar Conference (RadarConf23)

Drone Detection & Classification with Surveillance 'Radar On-The-Move' and YOLO

Hani Haifawi*†, Francesco Fioranelli*, Alexander Yarovoy*, Rob van der Meer†
*Microwave Sensing, Signals and Systems (MS3) Group, Delft University of Technology, Delft, The Netherlands
†Robin Radar Systems B.V., The Hague, The Netherlands
{hani.haifawi, rob.vandermeer}@robinradar.com; {f.fioranelli, a.yarovoy}@tudelft.nl

***Abstract*—A new method to jointly detect and classify drones using a moving surveillance radar system ('*radar on-the-move*') and computer vision is presented. While most conventional counter-drone radar-based techniques focus on time-frequency distributions to obtain classification features, such approaches are limited in volumetric spatial coverage. To compensate for this, surveillance radars that offer full spatial coverage are used, but the determination of the best detection and classification approach to be applied on the resulting data is still an open challenge. In this paper a framework is proposed that combines deep learning approaches from computer vision, specifically the *You Only Look Once* (YOLO) network, with data from the moving surveillance radar produced by Robin Radar Systems B.V. This framework allows to jointly detect and label targets based on range-Doppler images generated in real-time. The method is validated on experimental data, with preliminary results on a small dataset showing precision, recall, mean average precision (mAP@0.5) and Area Under Curve (AUC) of over 99%.**

***Index Terms*—drone detection, drone classification, surveillance radar, YOLO**

2023 IEEE Radar Conference (RadarConf23) | 978-1-6654-3669-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/RADARCONF2351548.2023.10149588

I. Introduction

Drone tracking and classification systems have received significant attention due to the exponential growth of the Unmanned Aerial Vehicles (UAVs) market, and the related concerns for accidental or intentional misuses of such platforms. To address this need of reliable monitoring capabilities, radar systems can provide robust detection and classification in any weather or light condition, as well as the direct determination of range and velocity of targets [1]. For this reason, counter-drone radar systems have become a safety requirement in a wide array of public and private events and at critical locations such as airports or power plants. Drone detection and tracking systems that are anchored on moving vehicles have also gained remarkable interest due to an ever increasing need for protection against rogue UAVs that might appear anywhere near and around an asset to be protected.

The ability to distinguish drones from other types of targets is typically achieved in literature thanks to the fast-rotating propellers that induce unique micro-Doppler modulations. Conventionally, micro-Doppler signatures are obtained using radars with fixed beams with high dwell time on target [2]–[4]. While this allows to capture a continuous sequence of samples necessary to obtain good-quality time-frequency signatures, such systems suffer of limited spatial coverage due to their static nature, so that the targets of interest may not be always present in the radar beam. In order to achieve full spatial coverage in a cost-effective manner, surveillance radars that use rotating antennas are adopted [5], [6]. However, this may result in losing the most salient features for drone classification due to the limited time on target. This limitation can be solved by staring radar systems that ensure wide spatial coverage continuously at all time [7], [8], but at the cost of higher system complexity and a very large amount of data to be processed. In terms of classification algorithms, machine and deep learning methods have gained a lot of momentum given their high performances in many applications including drone-related ones [9]–[13]. These approaches aim to solve a classification problem, i.e. assigning labels to detected targets. Hence, a detector is firstly needed to find the targets of interest before the classification stage.

In this paper, an approach that jointly solves the detection and classification problem of drones with a single algorithm is presented. The 'You Only Look Once' (YOLO) [14] framework from computer vision is chosen given the real-time inference speed of its small models, while retaining high accuracy for small objects identification and bounding boxes prediction. Essentially, YOLO exploits deep learning (DL) for both detection (finding a bounding box for the detected object) and classification (assigning a label to the box), rather than using DL only for classification tasks [10].

To the best of our knowledge, the usage of the YOLO framework and networks on radar range-Doppler plots for drone detection and classification is novel, as it has been primarily used only on optical images or videos. The system considered in this paper for the validation of the proposed approach is a surveillance radar with rotating antennas by Robin Radar Systems B.V., but with the additional complexity of its movement on the ground to patrol a certain area or asset of interest. The data used for classification via YOLO are range-Doppler plots generated by this radar, achieving multi-class and multi-instance object detection. The initial validation with experimental data reported in this paper showed promising results with precision, recall, and area under the precision-recall curve of over 99% in real-time.

The rest of the paper is organized as follows. Section II presents the radar system. The signal processing chain and YOLO-based detector are presented in Section III. The dataset is discussed in Section IV. Section V presents the experimental results, and the conclusion is drawn in Section VI.

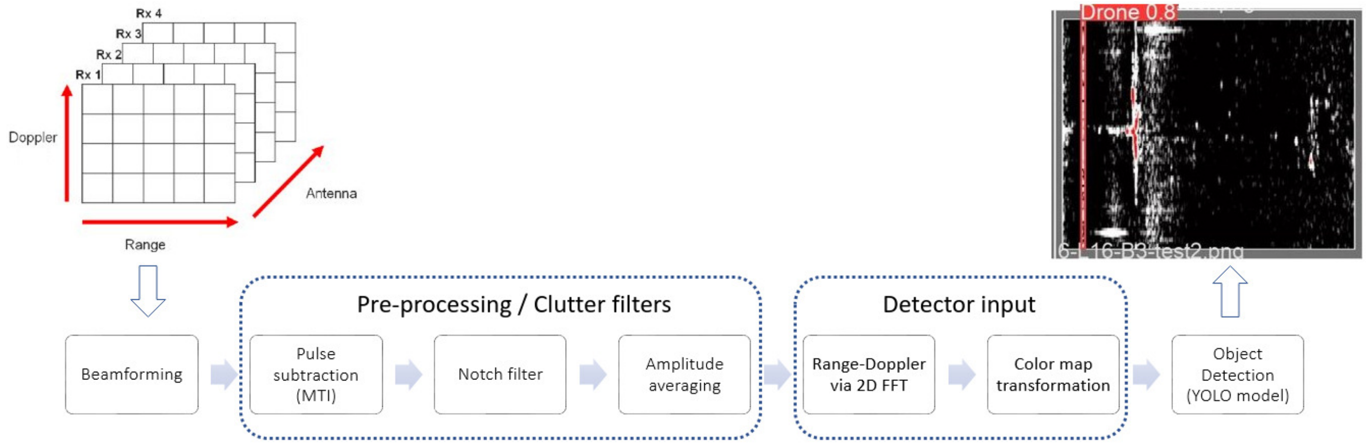

Fig. 1. Signal processing chain block diagram, from input radar data cube to an example of detection output generated by the YOLO.

II. Moving radar system characteristics

The radar system used in this paper is the *IRIS* FMCW Drone Radar developed by Robin Radar Systems B.V., operating at 9.25 GHz with a saw-tooth bandwidth modulation of 50 MHz [15]. The sampling rate is 15.625 MHz with a Sweep Repetition Frequency (SRF) of 4 kHz. A full rotation in the azimuth plane takes 2 seconds per antenna, and given that there are two antennas facing opposite directions, a full scan takes 1 second in total. The Coherent Processing Interval (CPI) consists of 100 consecutive sweeps that take 25 milliseconds and cover 4.5 degrees in azimuth. In the elevation plane, the radar has two phased array antennas, each with 8 elements. The radar was anchored on a moving platform on the ground and can detect, track and classify drones while cruising at approximately 50 km/h ('radar on-the-move').

III. Data processing pipeline

A. Signal processing chain

The signal processing chain is described in this section and depicted in Fig. 1. After the received signals are transformed into the radar data cube, beamforming in elevation is applied combining the data from the 8 receiver channels in order to increase the target Signal to Noise Ratio (SNR) at specific angles. Given the radar is both moving on the ground and mechanically rotating, care needs to be taken when applying clutter filtering techniques. Specifically, three different pre-processing steps are applied one after the other on the data:

- **Pulse subtraction:** this acts as a moving target indicator (MTI) or low-pass filter. It removes the weak static clutter centered around 0 Hz caused by the reflections received by the antenna back-lobe pointing in the opposite direction of the main lobe. Since these are mostly due to the mechanical setup of the radar, they yield a null Doppler shift as they move together with the vehicle.
- **Notch filter:** this is an adaptive autoregressive moving average filter that cuts a specific frequency band. Given the velocity of the platform and the orientation of the antenna is known a-priori, the cut-off frequency can be estimated in real-time. It removes the strongest dynamic clutter created from the ground reflections which is no longer static due to the movement of the sensor.
- **Amplitude averaging:** this processing step divides the absolute value of each range bin by the mean absolute value of all range bins in a single sweep. This helps attenuate the effect of background noise and remaining clutter in view of the subsequent steps.

After digital beamforming is applied and the sweeps are filtered as described above, a 2D Fast Fourier Transform (FFT) is used to obtain the range-Doppler plots. These are the inputs to the YOLO network. While traditional micro-Doppler approaches focus on time-frequency distributions such as spectrograms, due to the rotating nature of this radar, not enough samples are captured to perform this operation [3]. While spectrograms can offer rich micro-Doppler information [4], the Doppler signature of the drone's propellers can still be seen within the range-Doppler images [6], as in Fig. 2.

Conventionally, the absolute value of these range-Doppler matrices can be directly fed as input to neural networks for classification. However, given that YOLO was initially created for optical images, a further image processing step is applied to the range-Doppler matrices to generate visualisations such as the example shown in Fig. 2.

Essentially, this step is a colour transformation that maps the received signal power in the range-Doppler matrix to a color map designed to maximize the RGB contrast between the drone blades' signature (colored in white) and the background (colored in black), while retaining the drone bulk Doppler in red. The proposed color mapping is shown in Fig. 3.

This visualization facilitates the interpretation of the data by the YOLO network, making the contrast between different salient elements in the image more evident (e.g., the Doppler modulation due to the blades of the drones).

B. YOLO detector and its training

The chosen object detection & classification pipeline is YOLOv5s [14]. While there are deeper YOLO models available, due to the small size of the data set used and the relatively

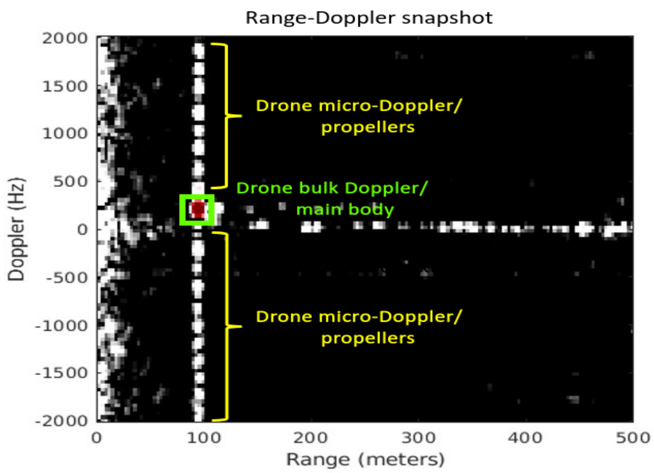

Fig. 2. Example of filtered range-Doppler containing a drone used as input for YOLO for detection and classification. The bulk Doppler and the micro-Doppler of the drone's propellers are both highlighted.

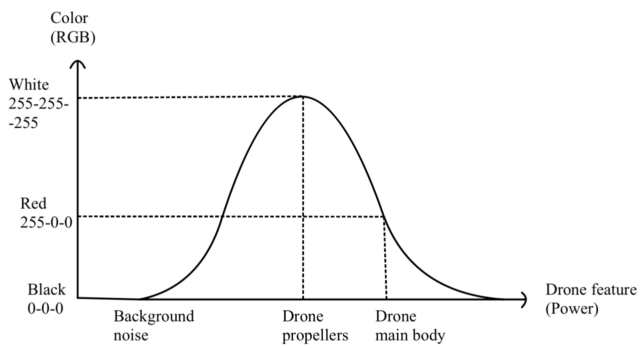

Fig. 3. Mapping of the range-Doppler matrix intensity values into the custom-defined RGB color map.

simple Doppler modulations of the propellers, this small model shows the best trade-off between speed and performance. YOLO is a one-stage detector that looks at the entire input image when processed, compared to two-stage detectors that use sliding windows. The architecture consists of a backbone, a neck and a head. The backbone is made of Convolutional Neural Networks (CNNs) used to obtain the feature maps via the receptive field based on multiple convolutions and pooling operations. The neck aims to mix and combine high-level features with low-level ones (feature pyramids), which are then propagated to the head. The head is therefore responsible for the output, i.e. predicting the bounding boxes.

Nevertheless, one of the main issues of deep learning approaches applied to radar data remains the small size of datasets. To address this issue, the network weights were initialized using transfer learning from a pre-trained model for 300 epochs over the COCO dataset [16]. The model was then trained over range-Doppler plots of drones for an additional 150 epochs, using the Stochastic Gradient Descent (SGD) and Adam as optimizers.

Moreover, data augmentation was included during the training stage to help mitigate overfitting and improve the performance metrics where possible. It should be noted that radar images embed the kinematics of targets that should not be altered, otherwise, the augmented propellers' Doppler modulations risk to be not realistic and not encountered in true experimental data. Thus, the data augmentation methods used in this approach are flip, translation, mosaic, hue, saturation, and value, denoted as 'albumentations' [17]. While the library contains over 60 data augmentation methods, techniques that preserve the kinematics coherence were chosen. For example, the flip is only done from left to right or up and down, but the image is never rotated because the Doppler signatures would then be horizontal instead of vertical, which is never encountered in real world. The transformation of images is done stochastically, such that the model never sees the same image twice.

A second step to help the model generalize better over unseen data is including weights decaying. At each epoch, a specific percentage of the weights is removed in order to prevent the model from learning too much from the specific training data, similarly to L2-regularization.

As discussed also in Section V, it was noted that the model initially yielded false alarms due to wind-turbines in the field of view that had similar Doppler modulations due to the rotation of their blades. By fine-tuning the training hyperparameters for this specific application using a genetic algorithm implemented by the YOLOv5 creator, the model became more robust against these incorrect detections [18].

IV. Dataset Generation and Composition

The data was recorded using pre-defined test scenarios with a Autel Evo I drone and the radar moving back and forth at different velocities. The IRIS Drone Radar was anchored on a moving vehicle, and the tests were taken with driving velocities ranging from 0 km/h up to 50 km/h. It should be noted that the data was collected in a clutter-rich area with a plethora of other targets, such as wind turbines in the background, vegetation moving with the wind, birds, trains, ships, cars, cyclists, and other objects that can occur in real-life environments and may yield false alarms. However, in this feasibility test the drone was always located relatively close to the radar, up to 500 meters, hence the data had good SNR. It is expected that data with lower SNR and less visible Doppler modulations of the propellers will reduce the performance in terms of precision, and this will be investigated in future work.

To label data, a bounding box is drawn around the range and Doppler bins containing the drone. Given the fast rotating propellers, usually all Doppler bins are covered by the propellers' signature, whereas with the available range resolution the drone tends to appear as a point-like target in range, occupying only a few range bins. The label then consists of 5 elements: the class ID and the bounding box coordinates (i.e., top left corner abscissa and ordinate coordinates, box height and box width).

The final model was trained using 188 drone range-Doppler images, and an equal amount of plots containing non-drone targets as described in Table I. As shown, the data was split as 60-20-20% between training, validation, and test sets to verify the performances of the model during training, but also the scalability over unseen data. It should be noted that to separate the training and test sets and to obtain natural data diversity, the training and validation data sets were collected in June, while the test set was collected in December. This separation ensures that during testing the model did not see data correlated to the validation and training set.

TABLE I
DATASET COMPOSITION AND SPLITTING

Data	Training	Validation	Test
Number of drone images	188	62	62
Number of total images	376	124	124
Ratio compared to whole dataset	60%	20%	20%

V. RESULTS

In this section the results generated by the proposed approach are presented and discussed. In terms of performance metrics, in computer vision recall can be interpreted as the area of overlap between the ground truth bounding box and the predicted one, divided by the area of the detected bounding box. Similarly, precision measures the same area of overlap, but this time it is divided by the ground truth object. Lastly, the mean Average Precision (mAP) is a weighted mean of precision scores at multiple thresholds, i.e. the area under precision-recall curve. Each new threshold represents the increment in the recall score compared to the prior threshold. The result is then averaged for each class. Thus, mAP highlights a trade-off between precision and recall that considers both false positives and false negatives, making it a very popular metric in all object detection applications. Moreover, mAP@0.5 means the average precision is computed for Intersection over Union (IoU) scores of 0.5, while mAP@$0.5 : 0.95$ represents the mAP taken for IoU values ranging from 0.5 up to 0.95 with a step of 0.05.

A. Handpicked hyperparameter space

The performances of two optimizers were analyzed: SGD and Adam. Additionally, the model was tested using handpicked hyperparameters, and the results are juxtaposed with those obtained by fine-tuning the model via genetic algorithms. The results are summarized in Table II.

One main difference between SGD and Adam is the emergence of false alarms. During training, SGD creates false detections over wind turbines that yield wide Doppler modulations due to their fast-rotating blades, as seen in Fig. 4. Adam on the other hand avoids such false alarms, as seen in Fig. 5. This is further reflected from the performance metrics, where Adam tends to have higher precision than SGD. Thus, while one simple solution to distinguish drones from wind turbines would be to verify the reflectivity and size of the target, the false alarms can be successfully removed by analyzing multiple optimizers for the training.

Fig. 4. Example of output of the proposed model, with detections obtained using SGD optimizer. A false alarm induced by a wind turbine is reported, together with a true detection.

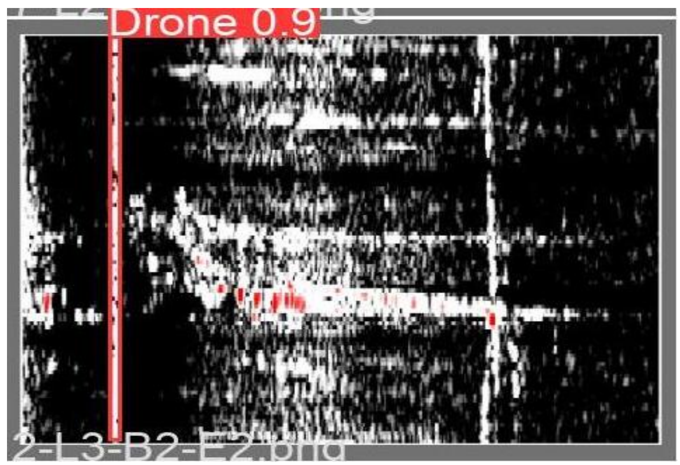

Fig. 5. Example of output of the proposed model for the same range-Doppler plot of Fig. 4, but using Adam as optimizer. The choice of this optimizer did not yield false detections.

B. Hyperparameters fine-tuning via genetic algorithms

The proposed solution to further adapt the model for the desired application is by fine-tuning the hyperparameters. While there are multiple methods to explore the hyperparameter space, in this paper genetic algorithms [18] were investigated and the new model results are summarized in Table II. The best combination of the hyperparameter space is chosen based on the best mAP@0.5 obtained. A mutation for the proposed dataset took approximately 5 minutes, and a total of 600 mutations were used for each optimizer. Thus, approximately 5 days were required to obtain the new hyperparameters' values.

Fig. 6 shows an example of results generated with the hyperparameter space from the genetic algorithm, without the false detection from the wind turbine. While Adam is less computationally expensive and did not yield false alarms as compared to SGD, these examples highlight the importance of the optimizer choice and proper hyperparameters initialization. Table II shows the improvements of the model obtained via

TABLE II
PERFORMANCE DIFFERENCES BETWEEN HAND-PICKED HYPERPARAMETERS AND THOSE OBTAINED VIA GENETIC ALGORITHM.

Model	Set	Hyperparameters tuning	Optimizer	Precision	Recall	mAP@0.5	mAP@0.5:0.95
1	Validation	Hand picked	SGD	98.3%	98.4%	99.4%	75.2%
1	Test	Hand picked	SGD	99.9%	98.4%	99.4%	76.6%
2	Validation	Genetic algorithm	SGD	98.2%	100%	99.5%	75.7%
2	Test	Genetic algorithm	SGD	100%	99.7%	99.5%	75.1%
3	Validation	Hand picked	Adam	100%	98%	99.5%	75.8%
3	Test	Hand picked	Adam	99.6%	98.4%	99.5%	76.2%
4	Validation	Genetic algorithm	Adam	100%	98.2%	99.5%	76.5%
4	Test	Genetic algorithm	Adam	99.9%	100%	99.5%	77.3%

genetic algorithms, especially for the results obtained over the test set. Nonetheless, while the false alarms were successfully suppressed for SGD, the mAP at multiple thresholds slightly decreased when the hyperparameters were initialized using genetic algorithms. For Adam meanwhile, all the performance scores improved. This may be due to Adam being an adaptive optimizer which is more difficult to initialize, but also because more mutations were required to find an optimal hyperparameter space. Hence, the genetic algorithms were able to find a better hyperparameter space to yield the best overall results.

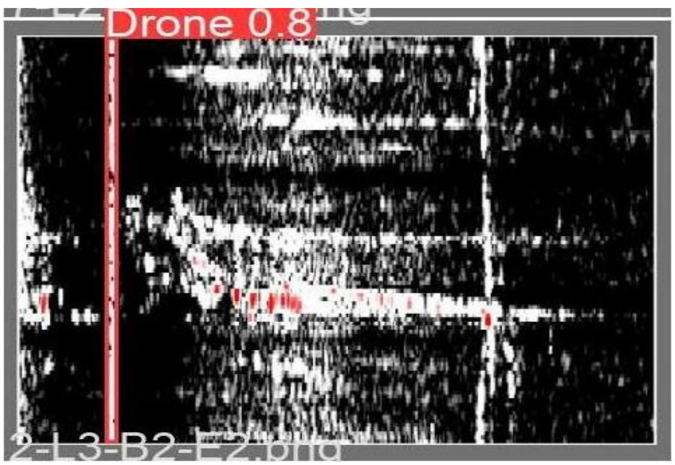

Fig. 6. Example of output of the proposed model for the same range-Doppler plot of Fig. 4 and 5, but using SGD optimizer and genetic algorithm. The genetic algorithm combined with SGD did not yield false detections.

C. Precision-Recall and ROC curves

The Receiver Operating Characteristic (ROC) curve is a popular tool to visualize the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR). However, the output of object detection frameworks such as YOLO is a bounding box surrounding the object of interest. Hence, the true negatives in this application represent all possible bounding boxes that were correctly not detected within an image, which would be in the order of thousands. Thus, the FPR is non-representative in object detection applications, which in turn limits the practicality of ROC curves. The Precision-Recall (PR) curve is proposed as a valid alternative [19].

The PR curve analyzes the model's false positive rate (i.e. the precision score) at multiple probability of detection thresholds (i.e. the recall score). Based on this curve, the Area Under the Precision-Recall Curve (AUC-PR) is computed, which represents the mean Average Precision (mAP). The AUC-PR constitutes one of the main ways to evaluate the performances of a model, given that a high AUC (or mAP) represents a high probability of detection and low false positive rate. In Fig. 7 the PR curve shows that the precision stays high as the recall score also increases, and the curve is almost optimal. The total AUC is 99.5%, which represents the mAP@0.5.

This high AUC score reflects the high target SNR, and further research to investigate the scalability and performance of this method on more challenging datasets is necessary.

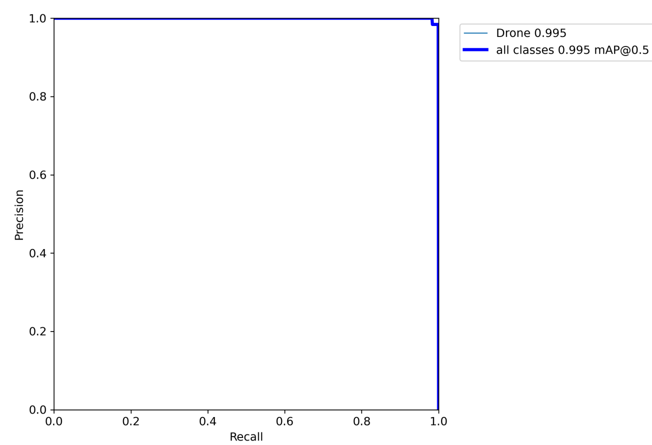

Fig. 7. PR curve of the model trained using Adam and genetic algorithm, highlighting the area under the precision-recall curve (or mAP).

D. Discussion and comparison with traditional algorithms

In these preliminary results, the proposed method was shown to accurately detect and label drones with high precision and recall scores based on range-Doppler plots. Moreover, the YOLO inference time is 1.7ms per image with approximately 0.2ms pre-processing time using a Nvidia GeForce RTX 3080 graphics card with 10 gigabyte total memory.

To the best knowledge of the authors, no similar computer vision method was investigated for the specific application of jointly detecting and classifying drones using surveillance radars' data. In the literature, a detector is firstly used to obtain the targets of interest, after which a classifier is used to assign the detected targets' labels. Thus, two different algorithms with different performance metrics and outputs are used to detect and label targets, while the proposed approach with YOLO solves this problem jointly. This makes the direct benchmarking of the method not straightforward.

For a drone with an experimental radar cross-section of approximately $-18\ dBm^2$, the same value as the Autel Evo II drone used in this project, the probability of detection using a constant false alarm rate detector for a Swerling case IV is given as below [20]:

$$cst = \frac{1}{1 + \frac{SNR}{2}} \tag{1}$$

$$P_D = \big(1 - cst \cdot (1 - cst) \cdot log(P_{FA})\big) \cdot P_{FA}^{cst} = 78.08\% \tag{2}$$

The classification however is done based on plots & tracks, compared to YOLO which labels targets based on range-Doppler plots. The current deep learning model used by Robin Radar has a precision score of 80.65% and a recall score of 97.60%. However, these benchmarks were exhaustively analysed for many test scenarios with multiple types of drones, different experimental conditions and a static radar, while the proposed YOLO model was tested for a radar on-the-move using the same drone. This calls for additional tests with more diverse data to be performed as future work.

VI. Conclusion

In this paper, a novel approach for joint detection and classification of drones is proposed. It uses the YOLO framework on range-Doppler plots captured with a surveillance 'radar on-the-move', i.e. a radar with a rotating antenna that is mounted on a ground moving platform. Different from the conventional approaches using micro-Doppler signatures from a staring radar, the proposed approach is shown to be effective when operating on single range-Doppler images corresponding to relatively short observation times (or CPI).

The approach has been validated with experimental data collected by the *IRIS* FMCW radar developed by Robin Radar Systems B.V., on a small dataset with a single drone flying in the scene of interest where other moving targets were present, including birds, targets on the ground, and wind turbines. Results in terms of precision, recall, and mean average precision (mAP@0.5) over 99% were obtained, showing the potential of the method and the opportunity to perform a broader verification on a larger dataset. This work validates the potential of applying deep learning methods for joint object detection and classification tasks in radar applications.

Acknowledgments

The authors are grateful to colleagues at Robin Radar Systems and TU Delft who supported the master thesis project of Hani Haifawi and related research.

References

[1] P. Wellig, P. Speirs, C. Schüpbach, R. Oechslin, M. RenkerM U. Boeniger and H. Pratisto (2018). Radar Systems and Challenges for C-UAV. 1-8.

[2] P. Molchanov, R.I.A. Harmanny, J.J.M. de Wit, K. Egiazarian, & J. Astola, (2014). Classification of small UAVs and birds by micro-Doppler signatures. International Journal of Microwave and Wireless Technologies, 6(3-4), 435-444.

[3] J.J.M. de Wit, R.I A. Harmanny and G. Prémel-Cabic, "Micro-Doppler analysis of small UAVs," 2012 9th European Radar Conference, 2012, pp. 210-213.

[4] R.I.A. Harmanny, J.J.M. de Wit and G. Prémel-Cabic, "Radar micro-Doppler feature extraction using the spectrogram and the cepstrogram," 2014 11th European Radar Conference, 2014, pp. 165-168.

[5] H. Sun, B.-S. Oh, X. Guo and Z. Lin, "Improving the Doppler Resolution of Ground-Based Surveillance Radar for Drone Detection," in IEEE Transactions on Aerospace and Electronic Systems, vol. 55, no. 6, pp. 3667-3673, Dec. 2019.

[6] B.-S. Oh, X. Guo and Z. Lin, A UAV classification system based on FMCW radar micro-Doppler signature analysis, Expert Systems with Applications, Volume 132, 2019, Pages 239-255, ISSN 0957-4174.

[7] M. Jahangir, C. J. Baker and G. A. Oswald, "Doppler characteristics of micro-drones with L-Band multibeam staring radar," 2017 IEEE Radar Conference (RadarConf), 2017, pp. 1052-1057.

[8] M. Jahangir and C. J. Baker, "Robust Detection of Micro-UAS Drones with L-Band 3-D Holographic Radar," 2016 Sensor Signal Processing for Defence (SSPD), 2016, pp. 1-5.

[9] M. Ritchie, F. Fioranelli, H. Borrion and H. Griffiths, (2017), Multistatic micro-Doppler radar feature extraction for classification of unloaded/loaded micro-drones. IET Radar Sonar Navig., 11: 116-124. https://doi-org.tudelft.idm.oclc.org/10.1049/iet-rsn.2016.0063

[10] S. Rahman and D.A. Robertson, (2020), Classification of drones and birds using convolutional neural networks applied to radar micro-Doppler spectrogram images. IET Radar Sonar Navig., 14: 653-661. https://doi-org.tudelft.idm.oclc.org/10.1049/iet-rsn.2019.0493

[11] S. Björklund, "Target Detection and Classification of Small Drones by Boosting on Radar Micro-Doppler," 2018 15th European Radar Conference (EuRAD), 2018, pp. 182-185.

[12] D. A. Brooks, O. Schwander, F. Barbaresco, J. Schneider and M. Cord, "Temporal Deep Learning for Drone Micro-Doppler Classification," 2018 19th International Radar Symposium (IRS), 2018, pp. 1-10.

[13] A. Huizing, M. Heiligers, B. Dekker, J.J.M. de Wit, L. Cifola and R.I.A. Harmanny, "Deep Learning for Classification of Mini-UAVs Using Micro-Doppler Spectrograms in Cognitive Radar," in IEEE Aerospace and Electronic Systems Magazine, vol. 34, no. 11, pp. 46-56, 1 Nov. 2019.

[14] Horvat, Marko & Jelečević, Ljudevit & Gledec, Gordan. (2022). A comparative study of YOLOv5 models performance for image localization and classification.

[15] Robin Radar Systems B.V., IRIS Counter Drone Radar. 2021. URL: https://www.robinradar.com/ iris-counter-drone-radar (visited on 18/07/2022).

[16] T-Y. Lin, M. Maire, S. Belongie, L.D. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, (2014). Microsoft COCO: Common Objects in Context.

[17] A. Buslaev, V.I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A.A. Kalinin, Albumentations: Fast and Flexible Image Augmentations. Information 2020, 11, 125.

[18] H. Haifawi, "Automatic Classification of Unmanned Aerial Vehicles with Radars On-The-Move", MSc thesis, TU Delft, August 2022, available at http://resolver.tudelft.nl/uuid:db11a4b1-d38f-4d62-80a4-0bea33793cdb

[19] J. Davis and M. Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (ICML '06). Association for Computing Machinery, New York, NY, USA, 233–240. https://doi-org.tudelft.idm.oclc.org/10.1145/1143844.1143874

[20] M.A. Richards, Fundamentals of Radar Signal Processing, Second Edition. McGraw-Hill Education, 2014, ISBN: 9780071798327, URL: https://books.google.nl/books?id=qGZCAQAACAAJ