Simulated Dataset for the Loaded vs. Unloaded UAV Classification Problem Using Deep Learning

Hamid Azad[1], Varun Mehta[2], Miodrag Bolic[1] and Iraj Mantegh[2]

***Abstract*— Detecting payloads on Uncrewed (or Unmanned) Aerial Vehicles (UAVs) is crucial for safety and security reasons. Deep learning methods can utilize changes in UAV appearance caused by payloads for detection, but collecting sufficient training data through real tests is costly and time-consuming. Therefore, simulation can be a more practical option. This paper presents the first synthetic air-to-air vision dataset for classifying loaded vs. unloaded UAVs. The dataset includes five types of aerial vehicles with attached and hanging payloads of different colors. It also incorporates three environmental conditions (sunny, rainy, and snowy) to diversify the background in recorded videos. Annotated frames and XYZ coordinates of the camera and drone are provided. To validate the dataset, a ResNet-34 network is trained with synthetic data and tested on real UAV flight data. The classification results on the test dataset confirm the effectiveness of the synthetic dataset for payload detection. The synthetic dataset and classification codes are publicly available on GitHub (`https://github.com/CARG-uOttawa/loaded-unloaded-drone-dataset`).**

Keywords - Drone, Uncrewed Aerial Vehicle, Unmanned Aerial Vehicle (UAV), Remotely Piloted Aircraft Systems (RPAS), UAV Payload, Counter UAV, Machine Learning, dataset

I. INTRODUCTION

Uncrewed or Unmanned Aerial Vehicles (UAVs), also known as drones and Remotely Piloted Aircraft Systems (RPAS), have attracted considerable attention recently. The U.S. Federal Aviation Administration (FAA) released its first approval of UAVs to be integrated into the nation's airspace in November 2013 [1]. Due to FAA Aerospace Forecast 2022-2042 [2], the number of registered UAVs with FAA by December 2020 was about 1.37 million (with an average of 10,300 per month during 2021). In Canada, this number was over 72,000 by December 2022, according to Transport Canada. Utilizing drones has grown dramatically in various commercial applications and governmental missions such as precision agriculture, cargo transport, search and rescue [3]. For a recent review of UAV technologies and applications, the reader is referred to [4]. However, the extensive usage of UAVs raises serious concerns about safety and security issues. The threats, such as transporting smuggled goods and other illegal substances, are among many concerns about UAVs. Thus, the classification problem between loaded and unloaded drones has received considerable attention in recent years.

A. Related works

Various sensors have been used for drone detection applications, including radar, Electoro-Optical (EO) sensors, and acoustic ones. The same sensors can generally be used for classification between loaded and unloaded drones. Many existing methods in the literature are based on classification approaches, using recorded radar data from drones. A majority of those considered the micro-Doppler signature from the UAV and extracted some discriminating features to classify between loaded and unloaded drones [5]–[11]. For example, in [12], [13], the authors calculated the short-time Fourier transform (STFT) of the radar signals in the collected dataset in the University College London (UCL) to analyze micro-Doppler signature of the micro-drone for different payloads. Then, they applied several classifiers on the features extracted from STFT to classify between loaded and unloaded drones. Among all references, only one paper has utilized acoustic data for loaded drone classification [3]. The study focused on the problem of remote detection of payload weight using the drone's acoustic fingerprint. It was shown that different weights cause changes in the speed of motors and thus the acoustic fingerprint. By using Mel-Frequency Cepstral Coefficients (MFCC) and SVM classifiers, they achieved a classification accuracy of around 98% for detecting specific payload classes carried by the drone.

The rapid progress in the field of computer vision and advances in deep learning algorithms make optical sensing one of the most attractive methods for drone detection and classification. By using cameras with different fields of view (FOV), with wide field coverage and narrow FOV for higher resolution and improved identification performance, various visual (image or video) data analysis methods have been proposed for drone detection/classification problems. Among all the references which have used the vision camera for drone classification, there are only a few that consider the payload detection problem.

Seidaliyeva et al. [14] proposed a deep learning approach to the two-class drone classification problem (loaded vs. unloaded). In this reference, the authors collected their data using a DJI Phantom 2 and captured some videos from it in both loaded and unloaded cases. In addition, because of the shortage in the number of recorded frames, some photos from drone delivery services such as Amazon and DHL were taken from some public open sources. Unfortunately, there is no specific information about the type of payloads used

[1]H. Azad and M. Bolic are with the School of Electrical Engineering and Computer Science (SEECS), University of Ottawa, 800 King Edward, Ottawa, On., Canada {`hamid.azad`, `miodrag.bolic`}`@uottawa.ca`

[2]V. Mehta and I. Mantegh is with National Research Council Canada (NRC), Montreal, Qc., Canada {`Varunkumar.Mehta`, `Iraj.Mantegh`}`@cnrc-nrc.gc.ca`

2023 IEEE Sensors Applications Symposium (SAS) | 979-8-3503-2307-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/SAS58821.2023.10254046

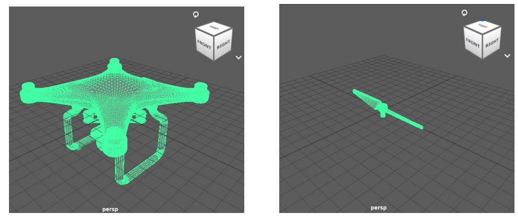
in the test and the dataset is not publicly available. The authors utilized a deep residual convolutional neural network (ResNet-34) for the binary classification problem between loaded and unloaded drones. The same authors proposed another solution based on YOLOv2 (You Only Look Once) for the two-class drone detection problem (loaded vs. unloaded) [15]. Both papers show acceptable results in the considered classification problem.

B. Simulated Dataset Generation

The vision camera is a commonly used sensor for drone detection and classification [16]. Deep learning (DL) methods have shown promising performance in various domains, but collecting and annotating a large amount of data for training remains a challenge. There is a scarcity of publicly available optical datasets for detecting payloads on UAVs. To address this, simulated datasets can be beneficial for quick investigations, with real data augmentation later in the training process. Simulations offer control over environmental conditions, unlike practical tests affected by factors like weather. AirSim©, an open-source simulation tool developed by Microsoft, enables the simulation of drones and cars in diverse environments.

AirSim© is based on Unreal Engine and provides an API for easy control of vehicles and data collection from sensors, including vision cameras [17]. Its "object detection" feature allows generating bounding boxes around desired objects in recorded videos. The simulated environment can be customized, offering control over weather and lighting conditions. These features facilitate the creation of comprehensive vision datasets for DL classification algorithms. By default, AirSim© provides a simple quadrotor model, limiting the dataset's diversity. However, it is possible to import custom drone models, although the process is not straightforward.

Contributions: To the best of authors' knowledge, this work is one of the few that applies computer vision and DL to the problem of payload detection on UAV platforms. To address the lack of an available dataset for loaded versus unloaded drone classification based on vision data, we have created the first publicly-available annotated air-to-air vision dataset. The dataset includes five diverse drone models in loaded and unloaded scenarios, captured in various environments and weather conditions. The recorded videos provide comprehensive training and evaluation data, with annotations and XYZ coordinates available for each frame. This dataset not only fills the data gap but also encourages the inclusion of other drone types in future research. However, this work and the prepared dataset focus on optical detection based on appearance features, excluding the kinematic and dynamic effects of the payload on UAV motion. The frame-by-frame classification problem relies on changes in the UAV's shape due to added payload, without extracting kinematic features from the videos. In addition to dataset preparation, we define various classification problems on the recorded dataset and train/test ResNet-34 (Residual Network with 34 layers) [18] to solve them (codes available via the provided link).



(a) Body object (b) Propeller object

Fig. 1: Prepared 3D objects to be imported into AirSim©

The classifier, trained on a portion of the synthetic dataset, is also applied to real-world test data from UAV flights. The favorable performance on unseen data demonstrates the richness and diversity of the prepared dataset.

The rest of the paper is organized as follows. The procedure for preparing the simulated dataset is explained in the next section, and some information about the collected dataset will be given. In section III, a brief explanation of the deep network that is utilized to solve the classification problem is presented. Then, in section IV, some samples of the recorded dataset, as well as the results of applying the DL-based method, will be given. Also, the results of both simulated and practical data are presented in this section. Finally, the conclusion will be given.

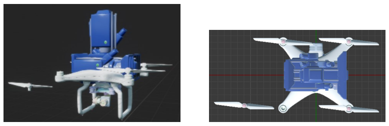
II. Simulated dataset using AirSim

As was explained in the introduction, there are no public vision datasets for the loaded vs. unloaded drone classification problem. AirSim© provides a platform for simulating flying drones in different environments. There is only one simple quadcopter available as the default model that can be utilized in the simulations. However, as an open-source tool, it is possible to import other drone models into AirSim© (although it is not straightforward work). In this paper, various models of drones in both loaded and unloaded forms have been imported to this environment in order to accomplish some flying scenarios and collect the required data. In this section, the procedure of creating the simulated air-to-air dataset, including the process of importing new drone models into AirSim©, will be explained. Besides, some statistics about the simulated dataset and an explanation of uploaded files on GitHub will be presented.

A. Procedure of importing a custom drone model into AirSim

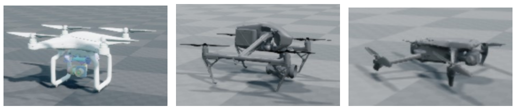
To import a specific UAV model into AirSim©, the 3D model needs to undergo some preparation. Since most 3D models of drones consist of multiple components, only the body and one propeller need to be extracted from the original 3D model using 3D design tools like Autodesk Maya [19]. These components should be combined into a single object, and the resulting 3D models should be exported as *fbx* files for use in AirSim simulations. For example, Fig. 1 shows the two prepared 3D objects (body and propeller) for the DJI Phantom drone.

After importing these *fbx* files into AirSim©, the drone model can be constructed by duplicating the default drone

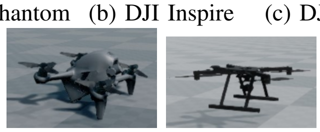


(a) Prespective view (b) Top view

Fig. 2: Adding the last propeller to the new drone model in AirSim©



(a) DJI Phantom (b) DJI Inspire (c) DJI Mavic



(d) DJI FPV (f) Quadrotor

Fig. 3: Different drone models simulated in the dataset

model and replacing the *body* and *propeller* parts with the imported objects. In the blueprint editing window, the prepared objects can be used to replace the corresponding parts of the drone model. Fig. 2 shows adding the last propeller to the new drone model in AirSim© (the blue 3D parts are just some AirSim© internal shapes to show the sensors mounted on the drone). It should be noted that the propellers are not static for the drone simulated using the above procedure. All the propellers would be controlled and rotated during hovering or flying scenarios.

Based on the explained procedure, 5 types of drones have been imported into AirSim© in our dataset (Fig. 3) including 4 DJI models [20] and also, a generic drone model. From this point on, this model is called Quadrotor in the paper, and its width with widely spread propellers is assumed to be around 70 cm. These drones are selected based on their size category. In other words, DJI Phantom and Quadrotor can be considered as two models in the large-size category. DJI Inspire is a sample of the mid-size drones and, DJI Mavic and FPV models can be categorized in the small-size group.

B. Payload simulation

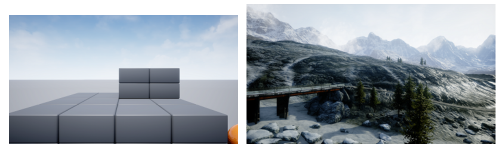
To enhance dataset diversity, various forms of payloads were simulated and incorporated into the drone models, including box-shaped payloads in two colors (black and orange) attached to the drones, as well as hanging payloads. Importing loaded drones followed a similar procedure to that of importing the drone itself. However, the payload was drawn separately using the *Curves/Surfaces* option in Autodesk Maya and then merged with the body object to create a complete 3D model of the loaded drone. Fig. 4 showcases examples of simulated loaded drones featuring different payload forms and colors.

C. Simulated Environments

To utilize imported drones in AirSim©, it is necessary to create custom environments as the pre-built binary files on

(a) Attached box payload (b) Hanging payload

Fig. 4: Two forms of loaded drones in the simulated dataset



(a) Blocks (b) Landscape mountains

(c) City park - Lite

Fig. 5: Different environments used in the simulated dataset

the AirSim© website are incompatible. In our dataset, we have chosen three environments: a simple *Blocks* environment, *Landscape Mountains*, and *City Park Environment-Lite* [21]. The *Blocks* environment represents the simplest case with a primarily sky background, except when the drone is near the blocks. The *Landscape Mountains* environment provides a natural setting with trees, rocks, and a lake in the background. Lastly, the *City Park* environment offers a blend of natural elements, such as trees, along with man-made objects like buildings. By simulating various backgrounds in the dataset, we account for their potential impact on the performance of vision-based classifiers. A sample scene of all these environments can be seen in Fig. 5.

D. Statistics about the simulated dataset

Considering the approach described above, the specification of the simulated dataset will be explained in this part. In practical situations, various types of drones may be used to carry the desired payload. Thus, in the published dataset, five types of drones, including 4 DJI models and the generic Quadrotor model, have been imported into AirSim© environment. Considering other simulation conditions explained in the previous parts, a brief explanation of dataset specification can be seen in Table I. In each case, a video of around 1000 frames with a resolution of 1920x1280 pixels and a field of view (FOV) of 82 degrees has been recorded, and the annotation file (using the "object detection" feature of AirSim©) has been provided. Fig. 6 shows an example of the shape of Quadrotor in both loaded and unloaded cases.

As mentioned in the previous sections, air-to-air video recording of the observed drone helps to have various view angles of it (compared to the case when the camera is on the ground and there is only a bottom view angle of the drone). To illustrate this advantage, Fig. 7 shows the captured image

TABLE I: Dataset explanation

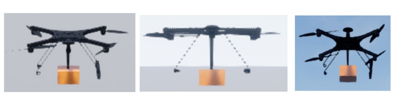
Types of drones	DJI Phantom, DJI Inspire, DJI Mavic, DJI FPV, Quadrotor
Environment	Simple block Env., Mountain landscape City park
Weather condition	Sunny, Rainy, Snowy
Payload condition	Unloaded, Attached load (box), Hanging load

Fig. 6: An example of the shape of Quadrotor in the collected dataset for the loaded and unloaded cases (all the images are resized to 255×255 pixels)

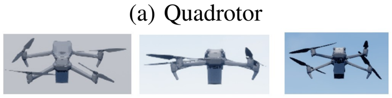
of two types of drones for various height differences between the camera and the observed drone.

The effect of weather on the recorded videos for sunny, rainy, and snowy weather can be seen in Fig. 8. It can be observed that in rainy and snowy weather, it is probable that the drops fall in front of the camera and cover some parts of the recorded scene, including the drone.

Fig. 9 shows the distribution of the collected dataset for DJI Phantom and DJI FPV as samples of two different size classes (for six scenarios and a total of around 6000 frames, including loaded and unloaded drones in various weather conditions in one of the environments). In both subfigures,

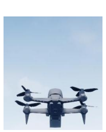


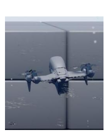
(a) Quadrotor



(b) DJI Mavic

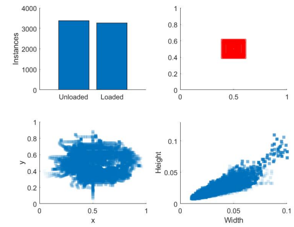
Fig. 7: Captured image of two types of drones for various height differences between the camera and the observed drone (left: up view, middle: same height, right: bottom view)



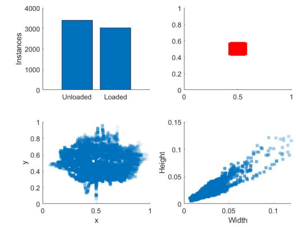


(a) Sunny (b) Rainy (c) Snowy

Fig. 8: The effect of weather condition on the captured images



(a) DJI Phantom



(b) DJI FPV

Fig. 9: Distribution of the collected dataset including the number of training images (top-left), Normalized bounding box shape for various frames (top-right), and scatter plots of objects' locations (bottom-left) in the frames and their sizes (bottom-right)

the top-left plot shows the histogram of the loaded and unloaded classes in the collected dataset. As can be seen, for both drones (and also, for other drones in the dataset), we have a uniform distribution for the number of samples for each class. The bottom-left figure shows the scatter plot of the location of the drone in the normalized frame region. This plot shows the drone is placed almost in all parts of this region. The top-right subfigure shows the normalized ground-truth bounding box width and height for various frames. This information shows the size of the drone in the dataset. In this figure, the normalized bounding boxes, with respect to the size of the image frame (1080×1920 pixels) for all the frames, are plotted. In addition, the size of both drones can be compared, and as it can be seen, on average, DJI Phantom has a bigger size than DJI FPV. Finally, the bottom-right subplot shows the scatter plot of the size of the bounding box and the size of the drone along the vertical and horizontal axis.

III. Classification Method

Although various forms of classifiers can be applied to the prepared dataset, the ResNet-34 [18] architecture is selected in this paper. The reason of selecting this architecture is that it showed acceptable performance in the same problem in [15]. ResNet-34 is a 34-layer deep convolutional neural network in the category of residual networks. To address the gradient vanishing problem as one of the challenges in the training phase of the network, ResNet structure was proposed. The building block of such networks consists of a connection between the output of one layer to the input of an earlier one. The structure of the residual block can be seen in Fig. 10. The basic structure of ResNet-34 is inspired by the famous VGG nets [22]. It consists of several layers with a kernel size of 3 by 3. In more details, it is composed of a total of 34 layers, which includes one convolutional and pooling layer, along with four other layers following the same pattern. Each layer utilizes a 3×3 convolution operation with feature map sizes of 64, 128, 256, and 512, respectively. There are also batch normalization step with the above identity connections, and also a fully-connected (fc) layer with softmax as the last layer. Since there are only two classes here, the last fc layer has changed to the case with two

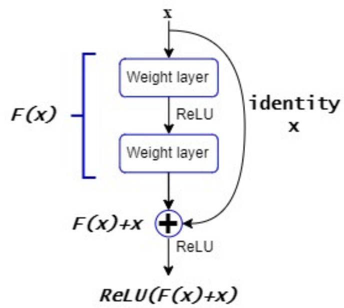


Fig. 10: A building block of residual network

outputs (probability of each class, i.e., loaded and unloaded drone). For a detailed description of the architecture, the reader is referred to [18].

IV. Simulation results

Due to the vast simulation conditions considered in the collected dataset (including the drone type, as well as payload and environmental conditions), various problems can be defined. Based on the portion of the dataset one considers for the problem and the training and testing phases, some of the suggested problems are:

- Loaded-vs-unloaded drone binary classification just for a single type of drone (the simplest case) in just one environmental condition or in the case of various environment/weather condition
- Loaded-vs-unloaded drone classification for the general case of several drone models in the dataset
- Classification among various payload forms for the single and/or multiple types of drones (unloaded vs. $\text{payload}_{\text{attached}}$ vs. $\text{payload}_{\text{hanging}}$)

This paper has considered and solved some of these problems using the deep network structure explained in the previous section. The hyperparameters of ResNet-34 have been set based on the values reported in [15]. The number of epochs and batch size are defined to be 25 and 64, respectively. Also, the value of the learning rate has been set to 0.03. Finally, the Stochastic Gradient Descent (SGD) optimizer has been used due to the better results reported using it compared to other optimizers. Using these settings, the results of classification for some of the defined problems in the previous part are given in Table II. For each problem, the data is divided into training, validation, and testing sets, with proportions of 70%, 15%, and 15%, respectively. To determine the number of frames in each row, as previously mentioned, approximately 1000 frames are simulated under each specific condition. For instance, in the second row of the table, there are a total of 6000 frames (3 weather conditions x 2 classes x 1000 frames per condition). These frames are further allocated as follows: 4200 frames for training, 900 frames for validation, and 900 frames for testing. Similar to previous works on the loaded vs. unloaded drone classification using vision data [14], [15], ResNet-34 shows highly accurate results on the binary loaded vs. unloaded drone classification problem.

Finally, it can be observed from this table that, as we expected, better classification results are achieved for the simpler cases. The classification accuracy decreases with an increase in the complexity of the background and environment (compare rows 3 and 4), weather conditions (compare rows 1 and 2, or 6 and 7), or a decrease in the size of the observed drone (compare rows 2 and 9).

TABLE II: Classification results using ResNet-34

	Case	Acc.
1	Quadrotor, Blocks env., Sunny (Payload with orange color)	1
2	Quadrotor, Blocks env., All weather (Payload with orange color)	0.9980
3	Quadrotor, Blocks env., Sunny	1
4	Quadrotor, City park env., Sunny	0.9916
5	Quadrotor, Blocks & City park env., Sunny	0.9941
6	DJIMavic, Blocks env., Sunny	1
7	DJIMavic, Blocks env., All weather	0.9883
8	DJIMavic, Blocks & City park env., Sunny	0.9609
9	DJIFPV, Blocks env., All weather	0.9644

A. *Results on practical data*

In addition to the simulation results, we also applied the ResNet-34 network, trained on simulated data, to some recorded practical data. Our practical data was captured using a quadrotor flying at various distances (from 20 to around 100 meters) from the camera (see Fig. 11). To have a loaded drone, a box-shaped payload (with dimensions of $7.8 \times 9 \times 15cm^3$) was attached underneath the drone. To test the classifier's performance in this study, around 1,800 frames of the recorded videos were selected for each loaded and unloaded case. The drone occupied an average of 20 pixels in width in the selected frames (the camera resolution was 1080×1920 pixels). It should be noted that the camera was mounted on a chasing drone, and thus we encountered an air-to-air test case. We achieved an accuracy of around 98% and 97% for classifying the unloaded and loaded drones, respectively. These results demonstrate the effectiveness of our simulated dataset for training the classifier network. The recorded practical data will be published soon.

V. Conclusion

This research introduced the first synthetic dataset specifically designed for the loaded versus unloaded classification of drones. By incorporating diverse types of drones, environments, payloads, and weather conditions, the collected dataset exhibits the necessary variation required for effective network training. Moreover, the proposed approach utilized the ResNet-34 deep network to address several classification problems within the dataset. The trained network, using the synthetic data, demonstrated promising results in accurately classifying loaded and unloaded drones within real-world scenarios.

The development of this synthetic dataset and the subsequent successful classification using deep learning techniques pave the way for future research in counter-drone systems.

(a) Unloaded

(b) Loaded

Fig. 11: A sample frame of the practical recorded video from unloaded and loaded drones, and the zoom of the detected drone

It opens up possibilities for further exploration of detection and classification algorithms, as well as the integration of real-world data to validate the effectiveness of the trained network. Initial results of practical data validate the proposed approach. However, as future work, importing more drone types and payload shapes can enhance dataset richness for research and practical applications.

ACKNOWLEDGMENT

This work is supported by the Canadian Safety and Security Program, which is led by Defence Research and Development Canada's Centre for Security Science (DRDC CSS), in partnership with Public Safety Canada. The Canadian Safety and Security Program is a federally-funded program to strengthen Canada's ability to anticipate, prevent/mitigate, prepare for, respond to, and recover from natural disasters, serious accidents, crime, and terrorism through the convergence of science and technology with policy, operations, and intelligence. Partial funding is provided from the NSERC Discovery Grant RGPIN-2020-04417, and complementary support has also been provided by National Research Council Canada through *Artificial Intelligence for Logistics* and *Integrated Aerial Mobility* programs.

REFERENCES

[1] FAA. (2021, Nov.) Press release – faa releases unmanned aircraft systems integration roadmap. [Online]. Available: http://www.tc.faa.gov/its/worldpac/uas/UAS_Roadmap_2013.pdf

[2] ——. (2021, July) Faa aerospace forecasts. [Online]. Available: https://www.faa.gov/sites/faa.gov/files/2022-06/Unmanned_Aircraft_Systems.pdf

[3] O. A. Ibrahim, S. Sciancalepore, and R. Di Pietro, "Noise2weight: On detecting payload weight from drones acoustic emissions," *Future Generation Computer Systems*, vol. 134, pp. 319–333, 2022.

[4] F. Ahmed, J. Mohanta, A. Keshari, and P. S. Yadav, "Recent advances in unmanned aerial vehicles: A review," *Arabian Journal for Science and Engineering*, vol. 47, no. 7, pp. 7963–7984, 2022.

[5] Y. Zhao, X. Zhang, and F. Fioranelli, "Initial results of radar-based classification of commercial drone carrying small payloads," in *2019 International Radar Conference (RADAR)*, 2019, pp. 1–4.

[6] J. S. Patel, C. Al-Ameri, F. Fioranelli, and D. Anderson, "Multi-time frequency analysis and classification of a micro-drone carrying payloads using multistatic radar," *The Journal of Engineering*, vol. 2019, no. 20, pp. 7047–7051, 2019.

[7] J. J. De Wit, D. Gusland, and R. P. Trommel, "Radar measurements for the assessment of features for drone characterization," in *2020 17th European Radar Conference (EuRAD)*, 2021, pp. 38–41.

[8] S. Rahman, D. A. Robertson, and M. A. Govoni, "Radar signatures of drones equipped with heavy payloads and dynamic payloads generating inertial forces," *IEEE Access*, vol. 8, pp. 220 542–220 556, 2020.

[9] ——, "Radar signatures of drones equipped with liquid spray payloads," in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–5.

[10] L. Pallotta, C. Clemente, A. Raddi, and G. Giunta, "A feature-based approach for loaded/unloaded drones classification exploiting micro-doppler signatures," in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–6.

[11] H. V. Sethuraman, "Radar-based classification of unmanned aerial vehicles (uavs) carrying payloads," Master's thesis, Delft University of Technology, 2021.

[12] F. Fioranelli, M. Ritchie, H. Griffiths, and H. Borrion, "Classification of loaded/unloaded micro-drones using multistatic radar," *Electronics Letters*, vol. 51, no. 22, pp. 1813–1815, 2015.

[13] M. Ritchie, F. Fioranelli, H. Borrion, and H. Griffiths, "Multistatic micro-doppler radar feature extraction for classification of unloaded/loaded micro-drones," *IET Radar, Sonar & Navigation*, vol. 11, no. 1, pp. 116–124, 2017.

[14] U. Seidaliyeva, M. Alduraibi, L. Ilipbayeva, and A. Almagambetov, "Detection of loaded and unloaded uav using deep neural network," in *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, 2020, pp. 490–494.

[15] U. Seidaliyeva, M. Alduraibi, L. Ilipbayeva, and N. Smailov, "Deep residual neural network-based classification of loaded and unloaded uav images," in *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*, 2020, pp. 465–469.

[16] F. Dadboud, V. Patel, V. Mehta, M. Bolic, and I. Mantegh, "Single-stage uav detection and classification with yolov5: Mosaic data augmentation and panet," in *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2021, pp. 1–8.

[17] Airsim. [Online]. Available: https://github.com/microsoft/AirSim/

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] Autodesk maya. [Online]. Available: https://www.autodesk.ca/en/products/maya/overview/

[20] SZ DJI Technology Co., Ltd. [Online]. Available: https://www.dji.com/

[21] Unreal marketplace. [Online]. Available: https://www.unrealengine.com/marketplace/en-US/store/

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.