Classification of Low-Resolution Flying Objects in Videos Using the Machine Learning Approach

Ivo STANCIC[1], Lucija VEIC[2], Josip MUSIC[1], Tamara GRUJIC[1]
[1]*Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, 21000 Split, Croatia*
[2]*Student at University of Split, 21000 Split, Croatia*
istancic@fesb.hr

***Abstract*—A challenge of detecting and identifying drones has emerged due to the significant increase in recreational and commercial drones operating range, payload size, and overall capabilities. Consequently, drones may pose a risk to airspace safety or violate non-flying zone in the vicinity of vulnerable buildings. This paper presents an initial study for a machine-learning classification system applied to flying objects visible with a low resolution that is too distant from the camera to be efficiently classified by other methods. The original dataset in form of labeled high-resolution videos containing low-resolution drone, bird, and airplane objects was collected and carefully prepared. Computationally inexpensive features based on object shape and trajectory descriptors were recommended and tested with several ML models. The accuracy of the best-proposed model tested on our dataset was 98%. The results of this study demonstrate that Machine Learning classification seems to be promising and can be implemented in future multi-stage drone detection and identification system.**

***Index Terms*—artificial neural networks, computer vision, feature extraction, machine learning, object detection.**

I. Introduction

As the availability and capabilities of unmanned drones have advanced in recent years, they are utilized for various applications in everyday life. Some notable examples today include agriculture, surveillance, search & rescue, and the film industry [1]. Nevertheless, they are also a tool for illegal operations, hence it is of key importance to effectively detect and identify potentially malicious drones [2].

Unmanned Air Vehicles (UAV-s) pose severe threats to airspace safety that may endanger public safety [3]. There are several examples where drones produced severe disruption to various airports and caused airplanes flights delays. One of the infamous incidents was the Gatwick airport incident [4], which delayed more than 1000 flights and affected 140000 passengers.

Flight authorities worldwide are permanently working on regulating drone operations, for both civilian and commercial purposes. A notable examples are the European Union (EU) regulations 2019/947 and 2019/945 [5] which set out the directions for the safe operation of civil drones in the EU. Regulation is very detailed and considers the weight and the specifications of the drone and intended operations. The regulation proposes the certification and licensing of the drone pilot which aims to ensure long – term safety of drone flight control. Those regulations may discourage careless misuse, but cannot prevent intentional misuse of drones. To ensure the effectiveness of these and similar regulations, it is important to support them by technological means, which encompasses drone detection, identification, and tracking near and inside no flying zones.

In recent years several papers have provided reviews on recent drone detection systems [6-8]. Reviewed papers addressed technologies that encompass acoustic [9-11], visual [12-19], radar [20-23], and radiofrequency [24-25] sensing systems. As pointed out by Taha et. al. [6], the majority of the recent research is only experimental and the outcomes can hardly be compared due to different technology and datasets. The review papers [6-7] indicate that the most of research that addresses the ability of advanced ML models to classify multi-rotor drones from other flying objects (birds, airplanes) seems to presume prior detection.

Acoustic methods are using microphones that can detect drones by analyzing sounds produced by a flying drone itself. Several papers presented a study for drone detection using a sound profile [9-10]. The authors pointed out a relatively low range of this approach, where the maximum acoustic detection range reported in the literature is 290 m [11].

Radar seems to be the most promising, but complex and expensive solution, enabling the detection of a single or several drones in the sky [21]. Employing the radar compared with the acoustic approach assure longer detection ranges. As an example, M. Jahangir et al. used a 10 kW 3-D radar aimed to detect a drone at a maximum distance of 1 km [20]. An overview of the most relevant technologies in radar sensor networks drone detection is provided in [22]. Some researchers aimed to achieve the non-trivial task of classifying drone types by using radar signals. Kaler et al. presented work on detecting rotary drones and distinguishing between single-propeller and multi-propeller drones using a micro-Doppler analysis [23].

The complexity and high cost of the radar have resulted in increasing research efforts that are primarily focused on employing visual data. Computer vision systems are

This work was supported by project "Development and implementation of methods for identification of bio systems and environments" VIF project under University of Split

Digital Object Identifier 10.4316/AECE.2022.02006

commonly used for the detection and identification of objects with respectable success. Recent advances in off-the-shelf, capable, and affordable cameras as the main sensors, supported with equally capable additional hardware may enable real-time detection and identification of flying objects.

Approaches that rely on computer vision can be divided into two major categories. The first is based on shallow Neural Networks (NN) and simple handcrafted features including edges, blobs or color. An example is a work reported by Unlu et al. which developed simple vision-based features [16].

The second, more advanced and accurate approach, is employing Deep Learning (DL) methods. As UAV exhibits distinctive construction and visual characteristics, when relatively high-resolution images of the object are available, DL methods may provide superior results when compared to shallow NN-s [12-13], [17-19].

Grác, Š., et al. developed an image-based detection of multi-rotor UAVs with 97.3% accuracy in ideal conditions [12]. Fast NN-s may provide real-time automated detection using specialized Neural Networks like YOLOv4 [18]. Some deep learning algorithms may be too slow to execute in real-time, hence researchers proposed a multi-stage detector, where the first simple detector is used to detect and localize possible drones on the image, while a more complex model is used for identification. Lead by this idea, Lee et al. proposed a Haar-like feature cascade classifier to detect the drone in the images and a secondary, more advanced CNN network for drone identification [15]. Vison and DL approaches may require high computational power to work in real-time. One example is the work by Kashiyama et al. where the authors presented a monitoring system for flying objects using a 4K camera and a convolutional neural network model which can achieve real-time processing [19]. The approach uses a high-performance cloud processing server that executes the YOLOv3 model, where several cropped images are processed in parallel.

Drones used for commercial and leisure purposes contain an on-board transmitter and receiver, used to control the UAV by the means of a Radio Frequency (RF) signal. Most of the before mentioned drones also contain cameras that can stream real-time video to the ground operator, thus constantly emitting RF signals. Those RF signals are usually in the ISM (Industrial, Scientific, and Medical) radio bands which are strictly defined and easily monitored. Another advantage of using the RF signal is the possibility to locate the ground operator by the same approach [24-25]. S. Al-Emadi and F. Al-Senaid presented work on drone detection solution using a Deep Learning (DL) technique and RF signals with reported accuracy and an F1 score of over 99.7% [26]. RF-based solutions may fail when the drone is controlled in fully autonomous mode, or its trajectory and mission are preprogrammed. Another step forward in creating a more capable detection system is based on sensor fusion [27-30]. Data from multiple sensors combined can obtain better results than derived from a single sensor. Each additional sensor compensates for individual weaknesses of the previous sensor and, therefore, the sensors fusion approach often enables better results as compared to a single sensor solution. Those approaches may use a combination of a visual camera and thermal camera [26] or a combination of several sensors [28]. Svanström et. al. presented preliminary results of a multi-sensor drone detection system utilizing state-of-the-art machine learning techniques and sensor fusion, which include signals from Infrared, visible camera, and audio signals [30]. The authors reported an F1 score ranging from 0.76 for the infrared sensor up to 0.93 for the audio sensor.

Some innovative approaches using non-standard sensors were also tested. One distinctive example is presented by Wojtanowski et al., where authors developed the sensor based on a laser scanner, which primarily aims to discriminate UAVs from birds [31].

As can already be noticed from the related papers cited in this section, most computer-vision-based approaches use some form of Machine Learning (ML) for object detection and/or decision making. Thus some basic machine learning algorithms used in our paper are briefly described in the following text. ML is artificial intelligence application where a machine learns from past experiences (input data) and makes future predictions [32]. The ML system learns from the dataset provided, usually a set of features extracted from learning examples. ML process consisted of several important phases: data collection, data preparing (or preprocessing), ML algorithm training phase, and testing phase, which finally results in the trained ML model which can perform future decisions, on un-seen data examples.

Datasets appropriate for training and testing ML algorithms, and for building an efficient ML model, should cover a realistic and the most possible widest range of cases. To reduce the model variance (overfitting), the model should learn from large data samples to obtain experience and finally provide generalization capability.

In the following text, some of the commonly used classification ML algorithms are briefly described.

Logistic Regression (LR) is one of the basic ML algorithms used to solve classification problems with a high simplicity of implementation and computational efficiency [33]. It returns the binomial outcome, as it gives the probability (the output variable) of whether an event will occur or not. If the output variable is equal to or exceeds a discrimination threshold, one class is predicted (the event occurred), otherwise, the second class is predicted (the event did not occur).

Naive Bayes (NB) methods are based on applying Bayes' probability theorem with the naive assumption of conditional independence between every pair of input features [33]. The probability table is formed from the feature's values where the algorithm needs to calculate the class probabilities for predicting a new observation (data example). The algorithm works well with small training data sets. More complex models (Support Vector Machine and Artificial Neural Networks) regularly outperform simple NB models if an adequate dataset is available.

The K-Nearest Neighbors (KNN) algorithm is a simple Machine Learning algorithm that can be used to solve both classification and regression problems [32]. As opposed to LR and NB, KNN does not assume any underlying data distribution, but instead, assumes that similar classes exist nearby in features space. The algorithm works by finding the distances between a new input and all the examples in the data. The main advantage of the KNN algorithm is its simplicity, but it has a major shortcoming of becoming significantly slow with larger training datasets.

Support Vector Machines (SVM) are supervised learning models that analyze data used for both classification and regression analysis [32]. The model can perform a non-linear classification using the kernel trick which is mapping inputs into high-dimensional feature spaces. The algorithm then creates a hyperplane in high-dimensional space that has the maximum margin or, in other words, the distance between data points of both classes which separates the data into classes. Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. SVM works well for smaller datasets with a lower rate of present noise and provides performance results similar to more complex Artificial Neural Networks.

A Decision Tree (DT) is a supervised Machine Learning approach aimed to solve classification problems by continuously splitting data based on a certain criterion [32]. The data is split into the nodes where the decision variable is categorical (the outcome is True or False) and the final decisions are in the leaves. The algorithm can be represented in form of a graph with choices and their results in form of a tree, which gives the algorithm's name.

Random Forrest (RF) is the solution when simple Decision Tree encounter overfitting. It consists of a large number of individual decision trees operating as an ensemble [32]. Each tree in the Random Forrest outputs a class prediction, and the class with the most votes becomes the model's prediction. The fundamental concept behind the Random Forrest is that a large number of relatively uncorrelated models working together will outperform any of the individual decision tree models. It can handle large datasets efficiently and can produce a reasonable prediction without hyper-parameter tuning, thus making it one of the fastest and most efficient ML algorithms.

Artificial Neural Networks (ANNs) are a series of algorithms that mimic the way the human brain operates [32]. The Neural Network consists of layers: an input, an output layer, and at least one hidden layer. The process of learning in the Neural Network occurs through the detection of hidden connections between layers of neurons. Output from a neuron in one layer acts as input to the next layer neuron and finally, the output of the last layer is the decision that ANN makes. Deep Learning (DL) is a Machine Learning technique successful in solving challenging problems which are too complex for other ML approaches. DL is commonly implemented using ANN with more than three hidden layers making it effective as Large Neural Networks. DL obtains the best results when compared to other ML models, and it can model complex non-linear relations. The drawback is that the building of the DL model requires considerable computer power for training as it uses the same data for several iterations, both for training and cross-validation.

This paper addresses the automated drone detection based on Machine Learning (ML) whose properties can be compared with the human observer in a similar situation. Due to reduced attention, fatigue, or boredom, the human observer may be unreliable for long–term drone searches. According to the Johnson criterion [34], when visual identification is applied to the drone–size object, the object must be visible with at least 14 pixels on a critical dimension to be effectively identified.

The UAV movement trajectories are most of the time linear, but drones often display sudden changes in direction, speed, acceleration, or height. As opposed to birds or airplanes, the UAV can soar for a prolonged time in a stable position. Analysis of these flight trajectories for short time intervals can produce usable ML models for drone detection and identification. The approach proposed in this paper aims to recognize drones based on their appearance and flight patterns in scenarios when only a low-resolution representation of the object is available due to the camera's wide-angle and long distances from the object (drone).

The contributions of this paper are summarized as follows:

- Collecting the original dataset of high-resolution videos with a fixed camera containing drones, airplanes, and birds, or their combination;
- Recommendation for computationally-inexpensive features based on object trajectory and shape, which are used as inputs to machine learning algorithms;
- The proposition of Machine learning models that best suit input features and offer the best execution times.

The paper is organized as follows: Section II describes our proposed method, Section III presents our experimental setup, used techniques, and describes the used datasets. Section IV presents the experimental work, results, and discussion. Finally, Section V summarises the conclusion and future work.

II. Proposed Methodology

The system for drone detection and identification proposed in this paper is based on computer vision, where ML models are in charge of classifying previously detected objects on the image.

The main idea behind this approach is to use a static single high-resolution camera that can cover a section of the sky and automatically analyze the video feed. As an illustrative example, a 4K camera (3840 x 2160 pixels resolution) that covers a 60° vertical angle can detect a small drone (e.g. 10 cm height) with 2 pixels (minimum for detection according to Johnson's criterion [34]) at a distance of 95 m. Longer distances can be reached by selecting another camera lens with a narrower field of view, or using a larger resolution camera.

The proposed solution is part of our continuous effort to develop a complete system for drone detection, tracking, and identification based on multiple optoelectronic sensors. As compared to the state-of-the-art systems reviewed in related work [6-8], our proposed system aims for real-time drone detection and recognition, which could be run on non-specialized hardware. As can be seen from the introduction, most of the recent solutions employ some form of deep learning approaches [12], [17-18], which require complex hardware and are demanding when higher resolution images are used, which in turn makes them unusable in many real-time scenarios. As an alternative to the recent advances, we intend to further exploit simple "hand-crafted" features, which can be easily obtainable even on higher resolution images (e.g. 4K) or series of images on low-cost hardware. To the best of our knowledge, similar shape descriptors were only used on static images where objects were represented with relatively sufficient resolutions [16].

The simplicity which arises from the utilization of computationally-inexpensive object shape and trajectory descriptors as an ML model feature enables extremely fast algorithm execution. Our approach analyses a broader range of descriptors/features which are described in the following text, while also adding time-based features describing objects' trajectory. The more complex task of final drone identification and possible trajectory estimation was not the primary goal of this work and should be considered a completely and separate task that requires additional research and hardware.

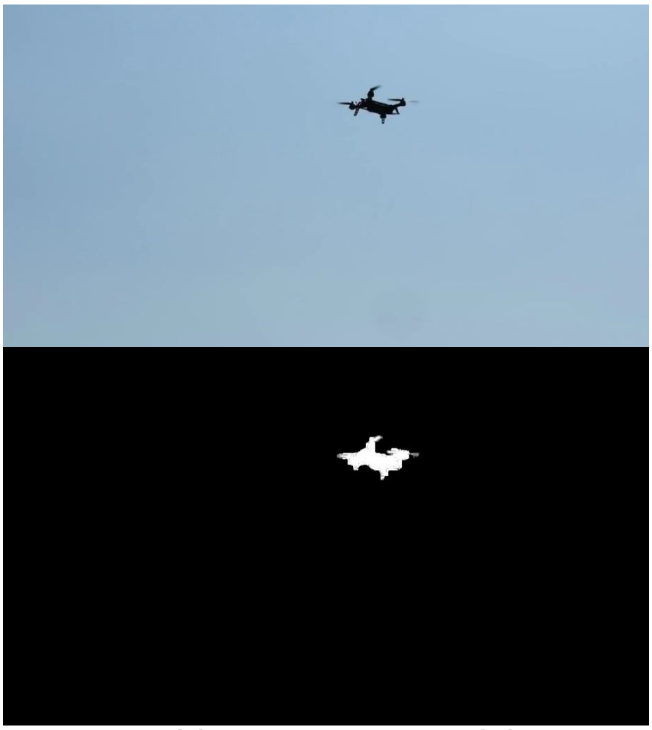

Figure 1. Foreground object extraction using Otsu's method

A common element of a system that detects and identifies objects by a single static camera is a subsystem that subtracts a background allowing distinguishing of the static scene (background) from dynamic objects in the air. The entire scene may have some time-varying components, such as tree branches and leaves that may move during windy conditions, or small variations in the complete scene caused by sudden cloud changes. Otsu's method is one of the most basic methods for extracting foreground objects in an image [35]. The method is a variance-based technique that finds optimal threshold values by analyzing variance between the foreground and background pixels. One example of object extraction is shown in Fig. 1, and for illustrative purposes, a relatively large object is presented in the image. Extracted objects in the proposed approach are considered in binary form (binarized image). The approach presented in our manuscript is solely aiming to propose a fast and real-time enabling classifier for objects extracted from the background, while an investigation or proposing improvements to existing background modelling and foreground extraction algorithms falls beyond the scope presented in this work.

Twelve features taken into the consideration are as follows:

- Horizontal change of the object center of mass location between two frames;
- Vertical change of the object center of mass location between two frames;
- The average speed of an object for the last 10 frames;
- The average acceleration of an object for the last 10 frames;
- Direction of the movement;
- Relative change in detected blob area between two frames;
- The ratio of object size and corresponding frame (extent);
- The ratio of size and average speed (in terms of pixels);
- The ratio of size and average acceleration;
- The ratio of a small and large part of the ellipse;
- The eccentricity of an ellipse that bounds the object;
- Orientation of ellipse that bounds the object.

The proposed system starts by detecting all foreground objects in every image from the video feed. The feature vector is then created for all objects available on the scene. Recommended features are in detail described in the following paragraphs. If a new object is detected, a corresponding new feature table is created, and when the object is lost in the cameras' ROI, the feature table stops updating.

All of the collected features are computationally inexpensive to calculate and are commonly included in CV libraries such as OpenCV or Matlab Image Acquisition toolbox.

Some of the selected features require analysis of several consecutive image frames (i.e. time-dependent), hence extracted objects have to be kept in memory for a few frames. Another advantage of implementing time-based features is the total independence of object resolution on an image, where the object can be presented with a relatively small number of pixels. The system is aimed to work on video streams (series of images) thus each detected object is tracked through all available frames using Optical Flow [36], effectively creating a feature table for each tracked object, where rows represent frames and columns represent features.

III. Experimental Setup

All videos were obtained by the Nikon B700 main camera which can record in up to 4K resolution. The camera was placed on a specially prepared gimbal system that enables controllable movement of the main camera system as shown in Fig. 2. Gimbal system features an additional Raspberry PI 4 computer board with a fixed compatible wide-angle camera. A motorized gimbal would in the future upgrade enable tracking of a detected object while changing ROI (Region of Interest) for the main camera, and the Raspberry PI computer would hold algorithms for automated tracking of a detected object.

Currently, motorized gimbal offers only (manually) controllable main camera orientation in global reference frame and preview of video captured with Raspberry PI camera. According to the main camera datasheet, focal lengths ranging from 4.3 mm to 258 mm and for sensor size of 1/2.3” (6.17 mm x 4.55 mm) offers variable visible angle in the range from 71° for the lowest focal length setting and 1.34° for the highest focal length setting. In practice, the highest focal length setting can “enlarge” a far object but the image is unstable due to motorized platform environmental impact (breeze). Setting the main camera focal length to a minimal of 4.3 mm would cover a larger proportion of the sky and consequently allow detection of the only object close to the detection system.

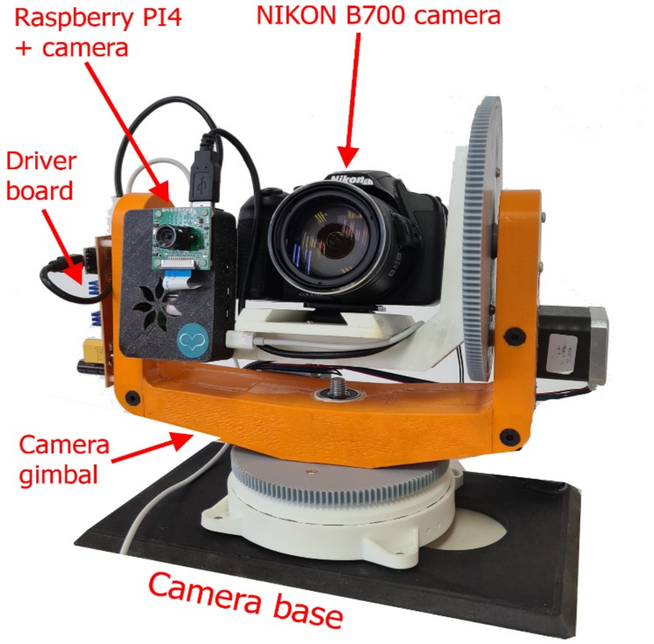

Figure 2. Experimental setup with gimbal, main camera, and additional Raspberry PI computer board

Camera focal length (zoom level) was set to cover around 10° of the horizontal visible range, which enables our preliminary trials to record object movements without having to readjust gimbal orientation. Videos were recorded at several locations and times of day (to ensure varying weather conditions) and contains drones, birds, and airplanes. Videos were idealized, with minimum background objects present in the scene. Drones themselves were remotely controlled by an operator or set to automatic route-following mode. To ensure better generalization of drone detection, several drone types with different capabilities (ability to maintain position and height) were tested: Bugs B6, Eachine EX5, Eachine E010, Eachine H8, and DJI Mavic mini drone. The original plan was to include several helicopter videos and add them as a new class, but not a single helicopter in flight was detected during data collection.

Sample image frames from recorded videos are presented in Fig. 3. Videos prepared in our dataset have a relatively simple scene, as the detection of objects in a complex environment was not the primary focus of our work.

Up to our knowledge similar dataset is not readily available, which complicates direct comparison with other approaches.

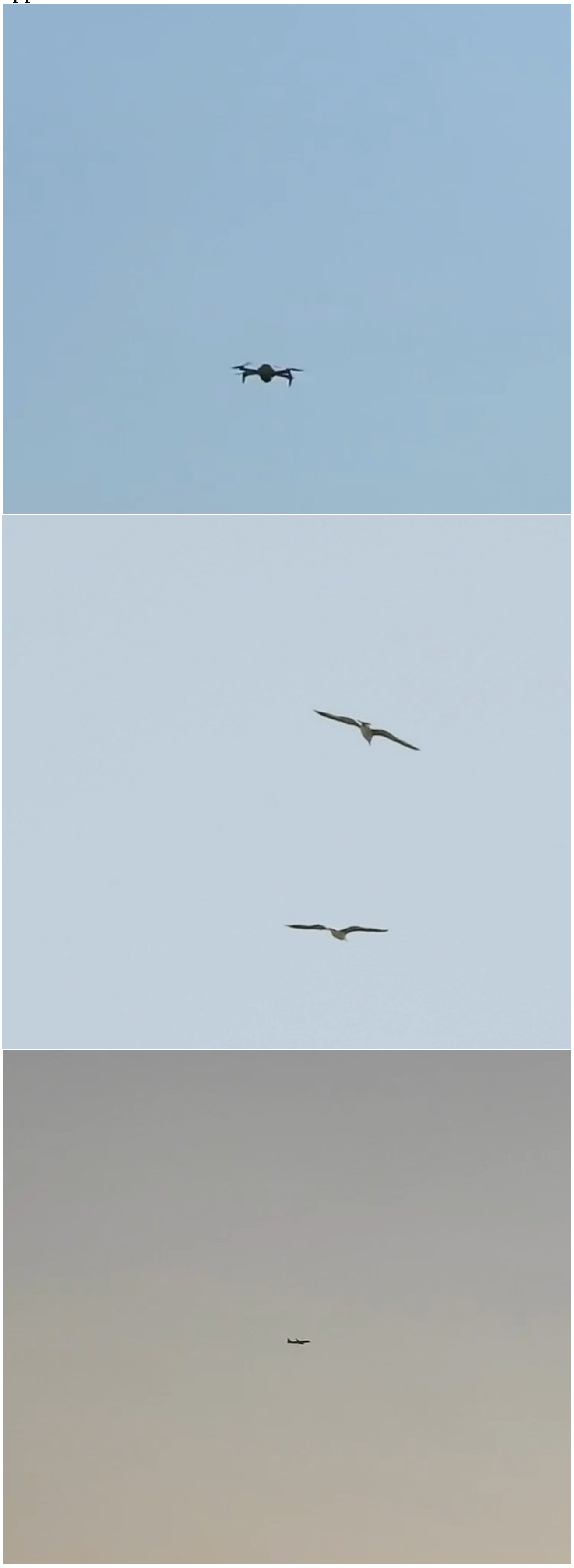

Figure 3. The cropped parts of sample images containing a drone (top), a bird (middle), and an airplane (bottom)

A key property of our dataset is that a fixed camera in combination with a large ROI enables the tracking of movement of all (relatively small) detected objects in the absolute reference frame (defined by a fixed camera) in a large area. This in turn enables the extraction of time-varying features (trajectory-based) which can't be effectively obtained by a non-fixed camera when extrinsic camera information is not available. To speed up the detection algorithm (for the real-time implementation), each video is resampled at 10 fps, since the preliminary test showed that 10 fps is sufficient in extracting some usable features in the time domain.

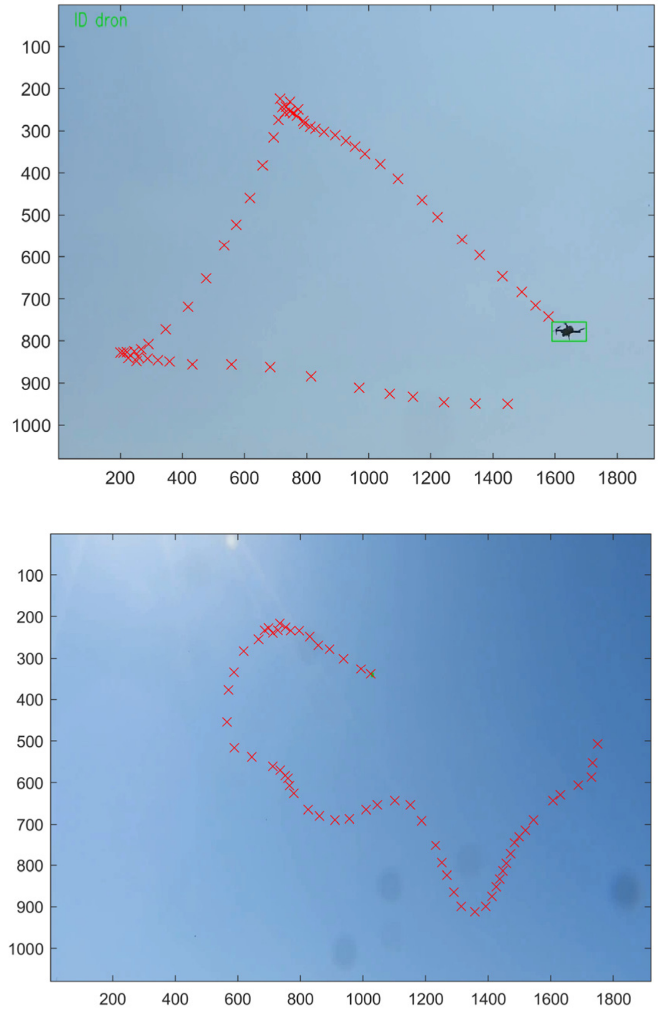

Figure 4. Trajectory placed over single video frame for drone (top) and bird (bottom)

The feature table was manually labeled, by visual inspection of the video feed and adding corresponding labels for each table row as follows: class 0 drone, class 1 bird, and class 2 airplane. Obtained data samples contain around 6000 rows of features, with 14% airplane classes, 34% drone classes, and 52% of bird classes. A series of 30 feature rows (that corresponds to the time frame of 3 s) was reshaped into a single vector that acts as an input of the ML model.

Fig. 4 contains images of the drone and bird (single frame from the video) together with their corresponding trajectories. Each mark on the image represents the object's centroid location. It is evident that drones exhibit more linear behavior while retaining on place for several frames (dense group of marks), while in contrast bird exhibits erratic curved behavior with constant change in position. Vectors are 25% overlapping using a sliding windows technique, thus effectively allowing the algorithm to make use of time-varying features. Feature tables with less than 30 elements were excluded from analysis as they do not provide enough information for correct classification.

Output (target) of a training model does not change its class during the flight. Feature tables are formatted in a form of a CVS (comma separated values) for easy import into various ML frameworks. Matlab R2021a [37] was selected for image pre-processing, training, and testing of the ML models. As Matlab already includes all toolboxes required for effortless image processing, extraction of objects, and finally data representation no other software packages were required at the moment. The proposed future upgrade would include rewriting the code to be compatible with Python / OpenCV [38], which would in turn enable the use of a motorized gimbal platform. Both training and testing were executed on an Intel i5-9400 workstation with 32 GB of RAM and dedicated NVidia GeForce GTX 1080Ti capable of executing ML algorithms.

IV. Results and Discussion

Selected ML models were trained using training data (90% of total samples), while the performance of each model was tested on test data (10% of total samples).

Figure 5. Result image containing detected and classified drone and two birds

Results containing training times and execution times for all used models are presented in Table I. As the number of input features was relatively small (12 features x 30 frames), the training of all models was impressively fast (Table I). Also, the execution time of models was fast, with Random Forest being the slowest (4.2 ms). Times required for detection of objects, and extracting features were not considered in this calculation, as it was not the main goal of this research.

TABLE I. Model Training and Execution Times

Model	**Training Time [ms]**	**Execution Time [ms]**
Logistic regression	952	0.7
Naïve Bayes	1230	0.9
KNN	493	4.1
SVM	7534	1.2
Decision tree	912	0.4
Random Forest	2512	4.2
ANN	14533	1.9

Parameters for the SVM algorithm are selected by cross-validation (RBF - Radial Basis Function kernel with C=10 and gamma 0.001). Also, several ANN architectures were tested, while the best results were obtained for 2 hidden layers with 5 neurons. In the Random Forest model 10 estimators were chosen as optimal.

Simple graphical output was provided during video analysis which labels the location and target class for each detected object. In an example, shown in Fig. 5, three objects were detected and correctly labeled. Here should be noted that for illustrative purposes objects in Fig. 5, were relatively large, while the algorithm performs well on objects which contain only several pixels. The system is capable of classifying all objects on an image if an object is large enough to be properly detected and extracted from the background. Each object is detected and tracked independently.

Results of accuracy for all tested models are presented in Table II. Of all ML models that were tested with our dataset, only SVM, ANN, and Random Forest showed promising results, with the classification accuracy of 95%, 96%, and 98%, respectively. Other considered models (apart from the Decision tree) performed with an accuracy below 90% and are not suitable for future study.

TABLE II. MODEL ACCURACY FOR OBJECT CLASSIFICATION

Model	Accuracy
Logistic regression	0.82
Naïve Bayes	0.77
KNN	0.79
SVM	0.95
Decision tree	0.94
Random Forest	**0.98**
ANN	0.96

Surprisingly, Random Forest showed slightly better results for a given input dataset, as compared to ANN, while the relatively simple ANN model was more than twice faster than the Random Forest model, with still considerable accuracy of 96%.

TABLE III. COMPARISON OF REVIEWED METHODS

Method	Sensors involved	Range	Performance
Acoustic [11]	Microphone array	Short	Distance 150 m-290 m
Radar [20]	Radar sensor network	Long	Accuracy 0.82-0.99
CV- Neural Networks [16]	Wide-angle camera	/	Accuracy 0.86
CV- Deep Learning [12]	Video camera	/	Detection Rate 0.97
CV- YOLOv4 [18]	Video Camera	/	F1 Score 0.79
CV- Haar + deep learning [15]	Onboard video camera	/	Accuracy 0.86
RF - Deep learning [26]	RF receiver	Long	F1 Score 0.99
Multi-sensor – ML [30]	IR camera / video camera Microphone	Short-Medium	F1 Score up to 0.93
Our proposed method	**4K Video Camera**	**Medium / Long**	**Accuracy 0.98**

To compare our approach with others reported in the literature for the similar performed task, we have provided Table III, which contains a list of sensors included in the research, reported detection distance (if available), and performance score. As reviewed papers are not uniform in reporting performance, some papers report an F1 score while others report classification Accuracy or Detection Rate. For the acoustic method [12] only the maximum distance achieved was reported. As pointed out by review papers [7-8] outcomes for different approaches cannot be directly compared due to different techniques used and a specific dataset that was obtained for each approach separately. Nevertheless, our proposed method performs with scores comparable to or better than competitive approaches, while offering adjustable range and fast model execution times.

V. CONCLUSION

In this research, several ML algorithms were employed for classifying previously detected and extracted small-resolution objects in the air. Databases containing drone, airplane, and bird videos were compiled in this research, using a fixed 4K video camera. Twelve different features were extracted by analyzing objects' shapes and trajectories, and eight different ML models were trained and evaluated. The best performance of the trained model for selected features showed to be a Random Forest with 98% of accuracy, with ANN as a second runner with 96% of accuracy. Both algorithms are executed extremely fast (4.5 and 1.8 ms, respectively) which indicates that both algorithms can be used in real-time scenarios. The bottleneck of the system was the detection and tracking subsystem, in which performance trials were not of key interest for this paper.

Obtained results are promising and display that objects of insufficient resolution can be classified using handcrafted features that include flight trajectory and basic shape descriptors. In future work, a more diverse and non-ideal video dataset will be used (with even more diversity of drones) to further improve the classification results. Other commonly found flying objects (helicopters, insects, etc.) should also be included in the dataset. In addition to dataset expansion, more features may be considered and analyzed for their usability for ML algorithms' performance.

REFERENCES

[1] R. Merkert and J. A. Bushell, "Managing the drone revolution: A systematic literature review into the current use of airborne drones and future strategic directions for their effective control," Journal of Air Transport Management, vol. 89, 101929, 2020. doi:10.1016/j.jairtraman.2020.101929

[2] V. Chamola, P. Kotesh, A. Agarwal, Naren, N. Gupta, and M. Guizani, "A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques," Ad Hoc Networks, vol. 111, Feb. 1 2021. doi:10.1016/j.adhoc.2020.102324

[3] G. Lykou, Georgia, D. Moustakas and D. Gritzalis, "Defending airports from UAS: A Survey on cyber-attacks and counter-drone sensing technologies," Sensors, no. 12: 3537. 2020. doi:10.3390/s20123537

[4] J. O'Malley, "The no drone zone," Engineering & Technology, vol. 14, no. 2, pp. 34-38, Mar 2019. doi:10.1049/et.2019.0201

[5] M. Huttunen, "Civil unmanned aircraft systems and security: The European approach," J. Transp. Secur, vol 12, pp 83–101, 2019. doi:10.1007/s12198-019-00203-0

[6] B. Taha and A. Shoufan, "Machine learning-based drone detection and classification: State-of-the-Art in research," IEEE Access, vol. 7, pp. 138669-138682, 2019. doi:10.1109/ACCESS.2019.2942944

[7] M. Azari, H. Sallouha, A. Chiumento, S. Rajendran, E. Vinogradov, and S. Pollin, "Key technologies and system trade-offs for detection and localization of amateur drones," IEEE Communications

Magazine, vol. 56, no. 1, pp. 51-57, Jan. 2018. doi:10.1109/MCOM.2017.1700442

[8] I. Guvenc, F. Koohifar, S. Singh, M. Sichitiu, and D. Matolak, "Detection, tracking, and interdiction for amateur drones," IEEE Communications Magazine, vol. 56, no. 4, pp. 75-81, Apr. 2018. doi:10.1109/MCOM.2018.1700455

[9] A. Bernardini, F. Mangiatordi, E. Pallotti, and L. Capodiferro, "Drone detection by acoustic signature identification," Electronic Imaging, vol. 2017, pp. 60-64, 2017. doi:10.2352/ISSN.2470-1173.2017.10.IMAWM-168

[10] J. Kim, C. Park, J. Ahn, Y. Ko, J. Park, and J. Gallagher, "Real-time UAV sound detection and analysis system," 2017 IEEE Sensors Applications Symposium (Sas), 2017. doi:10.1109/SAS.2017.7894058

[11] J. Busset et al., "Detection and tracking of drones using advanced acoustic cameras," Unmanned/unattended Sensors and Sensor Networks XI; and Advanced Free-Space Optical Communication Techniques and Applications, vol. 9647, 2015, Art no. 96470F. doi:10.1117/12.2194309

[12] S. Grac, P. Beno, F. Duchon, M. Dekan, and M. Tolgyessy, "Automated detection of multi-rotor uavs using a machine-learning approach," Applied System Innovation, vol. 3, no. 3, Sep. 2020, Art no. 29. doi:10.3390/asi3030029

[13] U. Seidaliyeva, D. Akhmetov, L. Ilipbayeva, and E. Matson, "Real-time and accurate drone detection in a video with a static background," Sensors, Article vol. 20, no. 14, Jul. 2020, Art no. 3856. doi:10.3390/s20143856

[14] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," IEEE Transactions on Pattern Analysis and Machine Intelligence, Article vol. 39, no. 5, pp. 879-892, May 2017. doi:10.1109/TPAMI.2016.2564408

[15] D. Lee, W. La, and H. Kim, "Drone detection and identification system using artificial intelligence," International Conference on Information and Communication Technology Convergence (ICTC), 2018. pp. 1131-1133. doi:10.1109/ICTC.2018.8539442

[16] E. Unlu, E. Zenou, and N. Rivière, "Using shape descriptors for UAV detection," Electronic Imaging, vol. 2018, pp. 128-1-128-5, 2018

[17] D. Lee, "CNN-based single object detection and tracking in videos and its application to drone detection," Multimedia Tools and Applications, vol. 80, no. 26-27, pp. 34237-34248, Nov. 2021. doi:10.1007/s11042-020-09924-0

[18] S. Singha and B. Aydin, "Automated drone detection using YOLOv4," Drones, vol. 5, no. 3, Sep 2021, Art no. 95, doi:10.3390/drones5030095

[19] T. Kashiyama, H. Sobue, and Y. Sekimoto, "Sky monitoring system for flying object detection using 4K resolution camera," Sensors, vol. 20, no. 24, Dec. 2020, Art no. 7071. doi:10.3390/s20247071

[20] M. Jahangir and C. Baker, "Robust detection of Micro-UAS drones with L-band 3-D holographic radar," Sensor Signal Processing for Defence (SSPD), 2016, pp. 1-5. doi:10.1109/SSPD.2016.7590610

[21] W. Zhang and G. Li, "Detection of multiple micro-drones via cadence velocity diagram analysis," Electronics Letters, vol. 54, no. 7, pp. 441-442, Apr. 5 2018. doi:10.1049/el.2017.4317

[22] A. Coluccia, G. Parisi, and A. Fascista, "Detection and classification of multirotor drones in radar sensor networks: A review," Sensors, vol. 20, no. 15, Aug. 2020, Art no. 4172. doi:10.3390/s20154172

[23] P. Klaer et al., "An investigation of rotary drone HERM line spectrum under manoeuvering conditions," Sensors, vol. 20, no. 20, Oct. 2020, Art no. 5940. doi:10.3390/s20205940

[24] S. Basak, S. Rajendran, S. Pollin, and B. Scheers, "Combined RF-based drone detection and classification," IEEE Transactions on Cognitive Communications and Networking, vol. 8, no. 1, pp. 111-120, Mar 2022. doi:10.1109/TCCN.2021.3099114

[25] M. Ezuma, F. Erden, C. Anjinappa, O. Ozdemir, and I. Guvenc, "Micro-UAV detection and classification from RF fingerprints using machine learning techniques," 2019 IEEE Aerospace Conference, 2019, pp. 1-13. doi:10.1109/AERO.2019.8741970

[26] S. Al-Emadi and F. Al-Senaid, "Drone detection approach based on radio-frequency using convolutional neural network," 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), 2020. pp. 29-34, doi:10.1109/ICIoT48696.2020.9089489

[27] F. Svanström, "Drone detection and classification using machine learning and sensor fusion," Master's Programme in Embedded and Intelligent Systems, School of Information Technology, Halmstad University, Halmstad, Sweeden, urn:nbn:se:hh:diva-42141, 2020

[28] S. Jovanoska, M. Brötje and W. Koch, "Multisensor data fusion for UAV detection and tracking," 19th International Radar Symposium (IRS), 2018, pp. 1-10, 2018. doi:10.23919/IRS.2018.8447971

[29] S. Samaras et al., "Deep learning on multi sensor data for counter UAV applications - A systematic review," Sensors, vol. 19, no. 22, Nov. 2019, Art no. 4837. doi:10.3390/s19224837

[30] F. Svanstrom, C. Englund, and F. Alonso-Fernandez, "Real-time drone detection and tracking with visible, thermal and acoustic sensors," 25th International Conference on Pattern Recognition (ICPR), pp. 7265-7272, 2021. doi:10.1109/ICPR48806.2021.9413241

[31] J. Wojtanowski, M. Zygmunt, T. Drozd, M. Jakubaszek, M. Zyczkowski, and M. Muzal, "Distinguishing drones from birds in a UAV searching laser scanner based on echo depolarization measurement," Sensors, vol. 21, no. 16, Aug. 2021, Art no. 5597. doi:10.3390/s21165597

[32] A. Burkov, The hundred-page machine learning book. Andriy Burkov, 2019

[33] J. Brownlee, Master machine learning algorithms: Discover how they work and implement them from scratch. Machine Learning Mastery, 2016

[34] T. A. Sjaardema, C. S. Smith, and G. C. Birch, "History and evolution of the Johnson criteria," Sandia National Lab., Albuquerque, United States, 2015. doi:10.2172/1222446

[35] T. Goh, S. Basah, H. Yazid, M. Safar, and F. Saad, "Performance analysis of image thresholding: Otsu technique," Measurement, vol. 114, pp. 298-307, Jan 2018. doi:10.1016/j.measurement.2017.09.052

[36] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2432-2439, 2010. doi:10.1109/CVPR.2010.5539939

[37] G. Ciaburro, MATLAB for machine learning. Packt Publishing, 2017.

[38] G. Bradski, "The OpenCV library," Dr. Dobb's Journal of Software Tools, vol. 120, pp. 122-125, 2000