Drone Detection With Improved Precision in Traditional Machine Learning and Less Complexity in Single-Shot Detectors

MOHAMAD KASSAB
Sorbonne University, Abu Dhabi, UAE
Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE

RAED ABU ZITAR, Member, IEEE
Sorbonne University, Abu Dhabi, UAE

FREDERIC BARBARESCO
Thales Group, Paris, France

AMAL EL FALLAH SEGHROUCHNI
Mohammed VI Polytechnic University, Rabat, Morocco
Sorbonne University, Paris, France

This work presents a broad study of drone detection based on a variety of machine learning methods, including traditional and deep learning techniques. The datasets used are images obtained from sequences of video frames in both the RGB and infrared (IR) formats, filtered and unfiltered. First, traditional machine learning techniques, such as support vector machine (SVM) and random forest (RF), were investigated to discover their drawbacks and study their feasibility in drone detection. It was evident that those techniques are not suitable for complex datasets (sets with several nondrone objects and clutter in the background). It was observed that the sliding window size results in a bias toward the selection of the bounding box when using the traditional nonmaximum suppression (NMS) method. Therefore, to address this issue, a modified NMS is proposed and tested on the SVM and RF. The SVM and RF with modified NMS managed to achieve a relative improvement of up to 25% based on the evaluation metric. The deep learning techniques, on the other hand, showed better detection performance but less improvement when using the proposed NMS method. Since their biggest drawback is complexity, a modified deep learning paradigm was proposed to mitigate the usual complexity associated with deep learning methods. The proposed paradigm uses single-shot detector (SSD) and AdderNet filters in an attempt to avoid excessive multiplications in the convolutional layers. To demonstrate our method, the most common deep learning techniques were comparatively tested to create a baseline for evaluating the proposed SSD/AdderNet. The training and testing of the deep learning models were repeated six times to investigate the consistency of learning in terms of parameters and performance. The proposed model was able to achieve better results with respect to the IR dataset compared to its counterpart while reducing the number of multiplications at the convolutional layers by 43.42%. Moreover, as a result of lower complexity, the proposed SSD/AdderNet showed fewer training and inference times compared to its counterpart.

I. INTRODUCTION

The advancements in drone capabilities witness technological improvements that allowed them to be used in a variety of applications, including military reconnaissance [1]. In addition to their ability to be invisible to radars, the wide availability of drones poses critical security threats to sensitive areas. Unauthorized small drones can carry explosive and radioactive materials to pursue an attack on individuals and infrastructures [2]. Furthermore, they are capable of invading privacy by the use of a mounted camera [2]. Drone detection techniques include, but are not limited to, radio frequency signal detectors, cameras, radars, and sensor fusion. Regardless of various techniques used to detect drones in literature and practice, drone detection remains a very challenging task today.

With the advancement of machine learning algorithms and the wide availability of computational power, machine learning techniques became a promising tool to tackle the problem of drone detection [3]. State-of-the-art machine learning models utilize deep learning algorithms that rely heavily on convolutional neural networks (CNNs) [4]. The CNN mainly uses multiplications on a heavy scale to extract features, hence making these models computationally expensive.

The main focus of this study is to investigate the use of deep learning techniques as well as traditional machine learning techniques for drone detection. It is crucial to emphasize that if the image processing system fails to keep up with the fast movement of the drone, a blurring effect may appear, consequently compromising the optimal functionality of the proposed detection system. That will require the intervention of blur elimination techniques in a preprocessing stage for the detection [5]. The primary contribution of this study lies in the evaluation of drone detection techniques using extensive and diverse datasets. The applied traditional machine learning techniques include support vector machines (SVMs) and random forest (RF).

Manuscript received 7 September 2022; revised 25 April 2023, 26 July 2023, 30 October 2023, and 1 January 2024; accepted 17 February 2024. Date of publication 23 February 2024; date of current version 9 August 2024.

DOI. No. 10.1109/TAES.2024.3368991

Refereeing of this contribution was handled by G. Chen.

This work was supported in part by Thales Group, Paris, France, in the Context of the Chair of Excellence Project, Sorbonne Center of Artificial Intelligence, Abu Dhabi, UAE.

Authors' addresses: Mohamad Kassab is with Thales/Sorbonne Centre for Artificial Intelligence, Sorbonne University, Abu Dhabi, UAE, and also with the Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE; Raed Abu Zitar is with Thales/Sorbonne Centre for Artificial Intelligence, Sorbonne University, Abu Dhabi, UAE, E-mail: (raed.zitar@sorbonne.ae); Frederic Barbaresco is with Thales Group, 75015 Paris, France; Amal El Fallah Seghrouchni is with Ai Movement—International Artificial Intelligence Center of Morocco, Mohammed VI Polytechnic University, Rabat 11103, Morocco, and also with Sorbonne University, 75005 Paris, France. *(Corresponding author: Raed Abu Zitar.)*

A modified nonmaximum suppression (NMS) algorithm was implemented for the SVM and RF to improve their detection performance and remove the bias toward the size of the sliding window. The study also proposed a hybrid single-shot detector (SSD) model by replacing the normal convolutional filters in the SSD's backbone with AdderNet filters, as proposed in [6]. Those models were compared to the performance of the benchmarks available in [7].

II. LITERATURE REVIEW FOR MACHINE LEARNING METHODS IN DRONE DETECTION

Object detection in the early days of artificial intelligence relied on handcrafted features and traditional object detection algorithms [8]. With the advancements in computational power and the saturation of performance improvement in traditional machine learning techniques, researchers have shifted toward deep learning to boost object detection performance. In the following subsections, the most common deep learning and traditional machine learning techniques will be discussed.

A. Traditional Machine Learning Methods

The use of traditional object detection techniques requires an extra step, which is the creation of handcrafted features. The handcrafted features used in traditional object detection models lack image representation techniques [8]. The SVM combined with the histogram of oriented gradient (HOG) descriptor was used in the early days of object detection [9], [10]. The use of kernels in the SVM allows mapping the data into a higher dimensional space, hence achieving nonlinearity such as deep learning methods [11]. Another traditional machine learning technique that can be combined with the HOG descriptor to perform object detection is RF, as proposed in [12] and [13]. RF is considered an ensemble supervised machine learning algorithm that uses multiple algorithms and takes the majority of votes when dealing with classification tasks.

B. Deep Learning Methods

Deep learning techniques utilize CNNs, which leverage the extraction of features without the use of handcrafted features. Some of the most widely used deep learning models include DETR [14], YOLO [15], SSD [16], and Faster-RCNN [17] . The DETR object detector is considered to be a two-stage detector, and it utilizes a vanilla transformer encoder–decoder structure with positional encoding and a feedforward network to make predictions. Similarly, Faster-RCNN is considered a two-stage detector, and it is composed of two modules, which are fully connected CNNs that output the regions of the proposals [17]. YOLOv3 is a one-stage detector, and it uses a single neural network to predict bounding boxes and probabilities from the full image. Unlike other detection methods, the loss function in YOLOv3 is directly related to the detection performance [15]. The SSD is a one-stage detector, and it uses the feedforward CNN that outputs bounding boxes and utilizes NMS to produce the final predicted bounding boxes [16].

III. METHODS

The hardware used to train and test the models is Intel Xeon Silver 4215, 128-GB RAM, and NVIDIA Quadro RTX 6000 with software platforms Python 3.7 and Ubuntu 20.04.5. In this section, the methods used for traditional techniques and the modified SSD/AdderNet will be presented.

A. Traditional Object Detection Techniques for Drone Detection

To perform drone detection using traditional techniques, two models were used: the SVM object detector and the RF object detector both with the HOG feature extractor. The training and testing of these methods were done on filtered Drone-Vs-Bird and AntiUAV-IR datasets. Both are subsets of the full datasets (see Section V for details). Filtered datasets were mainly used in these methods as the complexity of the full datasets needed huge training time without any relevant gain.

One of the problems noticed in the SVM and RF is their poor performance with highly complex data. Moreover, another problem that was identified during the training and testing phases is their need for a fixed feature size. This causes the algorithms to process only resized images that match the size of the input during the training phase. Since a sliding window is used, the height and width of the sliding window must be fixed. This can be done by either taking a fixed window size that matches the size used in the training phase or resizing the cropped image after taking a different window size. Our experiments included varying the size of the window and resizing the cropped testing images. By doing so, it was noticed that the SVM and RF are always biased toward selecting the wrong bounding box size. This issue arose mainly from the fact that the NMS algorithm selects the highest scoring bounding box; hence, the model will not always be able to cover the full object within the image. Consequently, the algorithm of NMS was modified to eliminate the property of selecting the bounding box with the highest score (even though many detections can be correct). The modified nonmaximum suppression (MNMS) allows the model to combine bounding boxes instead of elimination based on the score. In the proceeding part of this section, the traditional NMS algorithm and the proposed MNMS will be presented.

1) *Traditional NMS Algorithm:* The traditional NMS algorithm can be described as follows.

1) The bounding boxes $B = b_1, b_2, \ldots, b_n$, where n is the number of detections having a confidence score $S = s_1, s_2, \ldots, s_n$ higher than a predefined threshold, are used as an input for the NMS algorithm.
2) The algorithm then arranges the elements of B according to their respective score $s_{i \in n}$.
3) The algorithm then checks the condition $iou(b_i, b_j) \geq threshold$.
4) If true, the algorithm deletes the bounding box with lower confidence from B.

Hence, the traditional NMS algorithm deletes bounding boxes that are correct based on the highest score. Therefore, this causes the algorithm to select bounding boxes that do not cover the whole image.

2) *MNMS Algorithm:* The MNMS algorithm can be described as follows.

1) The bounding boxes $B = b_1, b_2, \ldots, b_n$, where n is the number of detections having a confidence score $S = s_1, s_2, \ldots, s_n$ higher than a predefined threshold, are used as an input for the NMS algorithm.
2) The algorithm arranges the bounding boxes into clusters based on the Euclidean distance d between them. If $d \leq$ predefined measure, then the bounding boxes are put in the same cluster.
3) If the number of bounding boxes in a cluster is exactly 1, $b_{i\in n}$ and $s_{i\in n}$ are discarded from B and S as they represent a weak detection.
4) The algorithm then checks the condition $iou(b_{i\in n}, b_{j\in n}) * (s_{i\in n}, s_{j\in n}) \geq threshold$ for the bounding boxes of each cluster.
5) If true, the algorithm merges the bounding boxes (b_i, bj) into one bounding box and returns the new bounding box.

The MNMS will first check if a single bounding box is within a cluster in order to delete it as it is considered a weak detection. A confident detection should have multiple overlapping bounding boxes; therefore, if a bounding box does not overlap with any other bounding boxes, then it must be a wrong detection. The algorithm will then calculate the intersection over union (IoU) for bounding boxes in the same cluster and multiply that value with the confidence score of each bounding box. The confidence scores range from 0 to 1; hence, the higher the scores are, the higher the intersection over the union value. Finally, if the intersection over the union exceeds a predefined threshold, the algorithm will combine the bounding boxes. Combining bounding boxes will allow the model to overcome the greediness toward a fixed feature size.

B. Modified SSD/AdderNet

To reach state-of-the-art performance, many researchers have increased the complexity of deep learning models dramatically. Hence, many models either require excessive computational power or are not suitable for real-time applications. As suggested in [18], at least 75% of the published papers in the Annual Meeting of the Association for Computational Linguistics, the Conference on Neural Information Processing Systems, and the Conference on Computer Vision and Pattern Recognition evaluate learning models based on detection accuracy only and do not consider computational complexity. Many of the proposed models in the literature have their accuracy proportional to an increase in computational complexity. For the purpose of serving practical applications, a model must take into consideration the tradeoffs between accuracy and computational complexity. In this section, we will describe our proposed SSD/AdderNet model for the purpose of decreasing computational complexity. The model uses the AdderNet filters proposed in [6] to replace the CNN filters in the architecture of the SSD model. AdderNet filters, as proposed by Chen et al. [6], replace multiplication operations in convolutional layers with addition and subtraction operations.

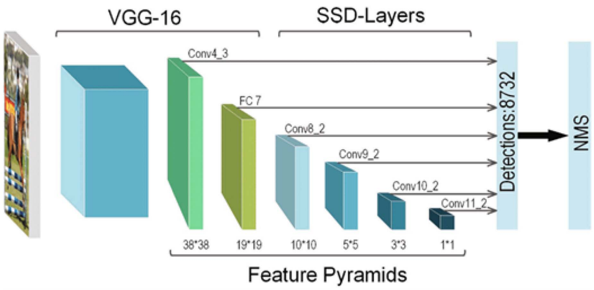

Fig. 1. SSD model [19].

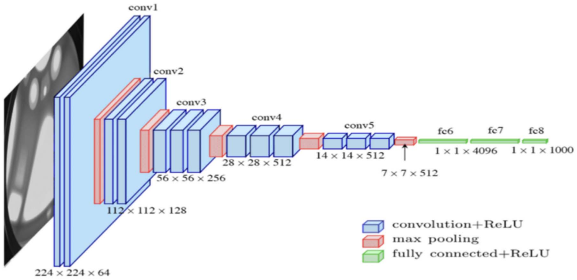

Fig. 2. VGG-16 [20].

C. Architecture and the Training Algorithm of the Proposed Method

The conventional structure of an SSD model, as proposed in [19], is shown in Fig. 1. The aim initially was to replace all the convolutional layers in the SSD model with adders. The resulting SSD/AdderNet showed high instability as a result of that. Therefore, only the convolutional filters of the backbone (VGG-16) were replaced. Since most of the convolutional layers are in the backbone, this resulted in eliminating most of the convolutional multiplications in the model. The VGG-16 proposed in [20] is shown in more detail in Fig. 2. The blue-labeled layers in Fig. 2 (convolutional layers) were replaced with AdderNet filters. Furthermore, the learning rate was set to 2e−4 initially and decayed to 2e−6 to ensure the stability of the hybrid model. The stochastic gradient descent (SGD) algorithm was used as the optimizer with a momentum of 0.9.

To understand the instability faced during the training phase of the hybrid SSD model, a custom dataset including drones and backgrounds (1228 drone images and 2460 background images) was trained for 50 epochs and tested on a classification task using the original model proposed in [6], which is a modified ResNet20 model with AdderNet filters.

From the plots obtained in Figs. 3–5, it is evident that the training accuracy, training loss, and testing accuracy of AdderNet filters are not stable; this can be a consequence of

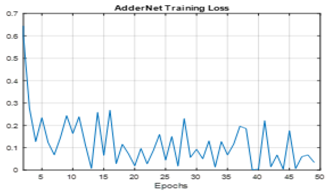

Fig. 3. ResNet20 AdderNet training loss.

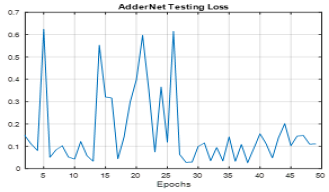

Fig. 4. ResNet20 AdderNet testing loss.

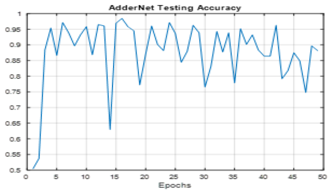

Fig. 5. ResNet20 AdderNet testing accuracy.

replacing normal convolutional filters with AdderNet filters that rely mainly on addition and subtraction. It is believed that this instability is caused by the learning rate parameter as changing from multiplication to addition changes the architecture of the model; hence, a different update parameter is required. The effect of the learning rate parameter value can be clearly seen in Figs. 3–5, as the model is very sensitive to the changes in weights at each epoch. Nevertheless, the final training accuracy reached is 88.2%, which represents a good tradeoff since AdderNet filters do not use any multiplications. Based on these experiments, the learning rate in our proposed SSD/AdderNet model was limited to the values mentioned earlier in order to minimize any resulting instability.

D. Analysis of the SSD/AdderNet Reduction in Complexity

The number of multiplications is used as a computational complexity measure to highlight the advantage of utilizing the hybrid SSD/AdderNet model. The number of multiplications per convolutional layer (NoM) can be mathematically modeled as follows:

$$\mathrm{NoM} = (N - 2 + 2P)(N - 2 + 2P)(m * m * n * d) \quad (1)$$

where N is the height/width of the input, P is the padding, m is the dimension of the kernel, d is the depth (3 in case of RGB), and n is the number of kernels. The number of multiplications that have been eliminated by the proposed model is equal to 367.7785 M, as shown in Table I. The total number of multiplications in the VGG architecture is 0.65 G [6]. Hence, the use of AdderNet filters to replace the convolutional filters reduces the multiplication complexity in the backbone of the SSD by 43.42%.

TABLE I
Computational Complexity Reduction of Convolutional Layers for the Proposed SSD/AdderNet Model

	Conv 1	**Conv 2**	**Conv 3**
Block 1	Number of filters : 64 Size of filter : 3×3 Padding : same Stride : 1 Input size : 224×224×3 Multiplications = 86.704 M	Number of filters : 64 Size of filter : 3×3 Padding : same Stride : 1 Input size : 224×224×3 Multiplications = 86.704 M	-
Block 2	Number of filters : 128 Size of filter : 3×3 Padding : same Stride : 1 Input size : 112×112×3 Multiplications = 44.35 M	Number of filters : 128 Size of filter : 3×3 Padding : same Stride : 1 Input size : 112×112×3 Multiplications = 44.35 M	-
Block 3	Number of filters: 256 Size of filter : 3×3 Padding : same Stride : 1 Input size : 56×56×3 Multiplications = 21.676 M	Number of filters: 256 Size of filter : 3×3 Padding : same Stride : 1 Input size : 56×56×3 Multiplications = 21.676 M	Number of filters: 256 Size of filter : 3×3 Padding : same Stride : 1 Input size : 56×56×3 Multiplications = 21.676 M
Block 4	Number of filters : 512 Size of filter : 3×3 Padding : same Stride : 1 Input size : 28×28×3 Multiplications = 10.838 M	Number of filters : 512 Size of filter : 3×3 Padding : same Stride : 1 Input size : 28×28×3 Multiplications = 10.838 M	Number of filters : 512 Size of filter : 3×3 Padding : same Stride : 1 Input size : 28×28×3 Multiplications = 10.838 M
Block 5	Number of filters : 512 Size of filter : 3×3 Padding : same Stride : 1 Input size : 14×14×3 Multiplications = 2.7095 M	Number of filters : 512 Size of filter : 3×3 Padding : same Stride : 1 Input size : 14×14×3 Multiplications = 2.7095 M	Number of filters : 512 Size of filter : 3×3 Padding : same Stride : 1 Input size : 14×14×3 Multiplications = 2.7095 M

IV. USED DATASETS

The datasets include a diverse range of drone types, including multirotor drones and fixed-wing drones, which travel at different speeds. The complex datasets used in this article include two RGB datasets and an IR dataset. These datasets are Drone-Vs-Bird, AntiUAV-IR, and AntiUAV-RGB. The datasets are annotated in the COCO format. Complex datasets in our work are datasets that contain images with randomly located objects in the background other than the drone itself. Complexity is mostly about the impurity of the background and the number of nondrone objects within the image. Both the factors increase the true–false detections that hinder the precision values. The datasets include a diverse range of drone types, including multirotor drones and fixed-wing drones, which can travel at different speeds. Moreover, referring to a "complex" dataset implies that the data possess challenges, such as variations in image sizes, scale variations of objects, and semantic complexity. These complexities have a greater effect on traditional machine learning algorithms especially since they often assume fixed input dimensions, making it challenging to handle images of varying resolutions and to capture the relevant features and patterns. As it is well known, traditional machine learning techniques struggle to handle such diverse characteristics in a dataset.

According to the COCO format [21], the sizes (in squared pixels) of the objects in the images of the datasets are divided into the following three groups:

1) small: size $<$ 32 × 32;
2) medium: 32 × 32 $<$ size $<$ 96 × 96;
3) large: size $>$ 96 × 96;

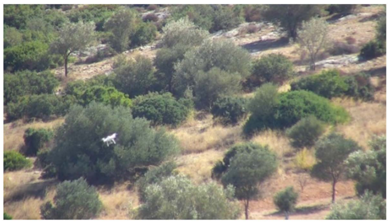

Fig. 6. Drone-Vs-Bird Sample 1.

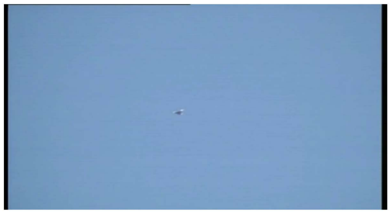

Fig. 7. Drone-Vs-Bird Sample 2.

The evaluation metrics employed to assess the models are aligned with the COCO format, and their descriptions are provided as follows.

1) *Average precision (AP):* It measures the average precision across different levels of precision–recall tradeoffs. The precision–recall curve is generated by varying the confidence threshold for predicted bounding boxes. The area under the precision–recall curve (AP) represents the average precision. AP summarizes the model's ability to accurately detect objects across different recall levels.
2) *$AP_{0.5}$:* It is a specific variant of average precision where the IoU threshold for matching predicted and ground truth bounding boxes is set to 0.5. This means that a predicted bounding box is considered a true positive if it has an IoU of 0.5 or higher with the ground truth box.
3) *$AP_{0.75}$:* Similar to $\text{AP}_{0.5}$, but with a higher IoU threshold of 0.75. It measures the average precision when a predicted bounding box needs to have a higher overlap (IoU) with the ground truth box to be considered a true positive.
4) *Average precision small (AP_S):* It represents the average precision specifically for small drones in the dataset. It focuses on evaluating the model's performance in detecting smaller drones.
5) *Average precision medium (AP_M):* It calculates the average precision for medium-sized drones.
6) *Average precision large (AP_L):* It measures the average precision for large drones.

A. Drone-Vs-Bird Dataset

The Drone-Vs-Bird [22] dataset contains 77 RGB videos. The videos can be converted into images with COCO annotations using the tools provided in [7]. Samples from the RGB dataset are shown in Figs. 6 and 7.

TABLE II
Number of Samples in Full Drone-Vs-Bird

Object size	Training samples	Validation samples
Small	63295	3578
Medium	21124	1104
Large	2413	63
Background	13034	444
Total	99866	5189

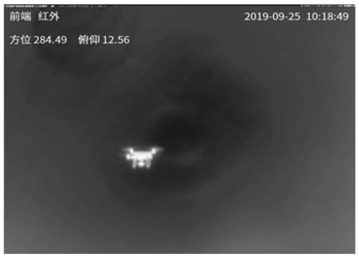

Fig. 8. Anti-UAV-IR Sample 1.

Fig. 9. Anti-UAV-IR Sample 2.

The challenge of the Drone-Vs-Bird dataset can be seen in Fig. 7, where the size of the drone is extremely small. Moreover, the images were generated from videos; therefore, many of the images are repeated, and they contain one class which is "Drone." The number of images obtained after converting the videos is provided in Table II. Furthermore, the Drone-Vs-Bird dataset was filtered by removing small-sized objects to experiment with the effect of object size on the learning process of object detectors. The total (unfiltered) dataset will be referred to as the full dataset, and the filtered dataset will be referred to as the filtered Drone-Vs-Bird dataset.

B. Anti-UAV-IR Dataset

The Anti-UAV-IR dataset can be obtained from [23], and it contains 140 IR videos. The videos can be converted into images with COCO annotations using the tools provided in [7]. Samples from the IR dataset are shown in Figs. 8 and 9.

Compared to the Drone-Vs-Bird dataset, the drone sizes of the Anti-UAV-IR dataset are better. However, there are still extremely small objects, as shown in Fig. 9. The dataset is generated by converting videos into images; therefore, there are many replicates, and the dataset contains only one class which is "Drone." The number of images obtained after converting the videos is provided in Table III. Furthermore, the Anti-UAV-IR dataset was filtered by removing small-sized objects. The full (unfiltered) dataset will be referred to as full Anti-UAV-IR, and the filtered dataset will be referred to as filtered Anti-UAV-IR.

TABLE III
Number of Samples in Full Anti-UAV-IR

Object size	**Training samples**	**Validation samples**
Small	58362	2373
Medium	38660	2670
Large	94	0
Background	1560	197
Total	98676	5240

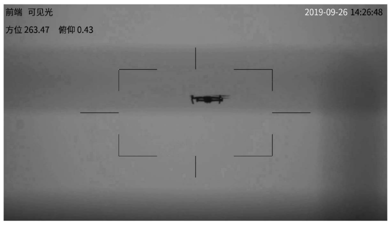

Fig. 10. Anti-UAV-RGB Sample 1.

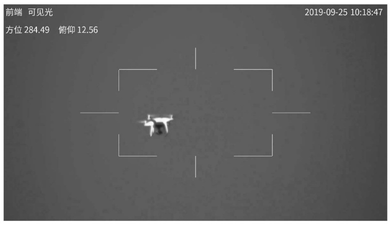

Fig. 11. Anti-UAV-RGB Sample 2.

TABLE IV
Number of Samples in Anti-UAV-RGB

Object size	**Training samples**	**Validation samples**
Small	53213	2578
Medium	63216	2104
Large	9784	1436
Background	14672	2525
Total	140885	8643

C. Anti-UAV-RGB Dataset

The Anti-UAV-RGB dataset can be obtained from [23], and it contains 318 RGB videos. The videos can be converted into images with COCO annotations using the tools provided in [7]. Samples from the IR dataset are shown in Figs. 10 and 11. All the samples in this dataset were used for training and testing deep learning models only.

The number of images obtained after converting the videos is provided in Table IV.

V. EXPERIMENTS AND SIMULATION RESULTS

This section presents the simulation results of the filtered Drone-Vs-Bird dataset and the filtered Anti-UAV-IR dataset using the traditional machine learning methods (SVM with HOG and RF with HOG). Furthermore, we present the experiments of the modified SVM and RF on the same datasets to investigate whether or not NMS is a cause of the poor performance of these methods. The experiments were limited to the filtered datasets in the case of traditional machine learning techniques as they use a sliding window (time consuming), and the number of training and testing samples is too large with unfiltered datasets. Therefore, to speed up the training and testing process, the filtered datasets were used for SVM and RF object detectors. In addition, we have experimented with the MNMS on SSD and SSD/AdderNet using Anti-UAV-IR and Anti-UAV-RGB datasets to investigate whether or not any advantages can be gained from it when used with deep learning methods. Moreover, the simulation results of the deep learning methods (SSD/AdderNet, SSD, Faster-RCNN, YOLOv3, and DETR) based on the unfiltered Drone-Vs-Bird, unfiltered Anti-UAV-IR, and unfiltered Anti-UAV-RGB datasets are also presented. To make our comparisons based on the datasets more reliable and to investigate the consistency of deep learning methods, training and testing for each method were repeated six times. Furthermore, the mean, standard deviation, and confidence interval were calculated for each method. The aim of these repeated experiments was to perform solid and confident comparisons among the different methods.

TABLE V
Filtered Drone-VS-Bird using SVM and RF Object Detectors

Model	**AP**	**AP$_{0.5}$**	**AP$_{0.75}$**	**AP$_{S}$**	**AP$_{M}$**	**AP$_{L}$**
SVM	0.060	**0.287**	0.002	–	0.060	–
SVM$_{\text{NMS}}$	**0.065**	0.280	0.002	–	**0.065**	–
RF	0.015	0.082	0.006	–	0.015	–
RF$_{\text{NMS}}$	0.021	0.089	**0.0087**	–	0.021	–

TABLE VI
Filtered IR Using SVM and RF Object Detectors

Model	**AP**	**AP$_{0.5}$**	**AP$_{0.75}$**	**AP$_{S}$**	**AP$_{M}$**	**AP$_{L}$**
SVM	0.065	0.290	0.003	–	0.065	–
SVM$_{\text{NMS}}$	**0.081**	**0.321**	**0.014**	–	**0.081**	–
RF	0.020	0.085	0.0084	–	0.020	–
RF$_{\text{NMS}}$	0.024	0.091	0.0089	–	0.024	–

TABLE VII
MNMS on Deep Learning Using Anti-UAV-RGB

	AP	AP$_{0.5}$	AP$_{0.75}$	AP$_{S}$	AP$_{M}$	AP$_{L}$
SSD	0.6233	0.9360	0.7380	0.2217	0.6308	0.6705
SSD w/MNMS	**0.630**	**0.938**	**0.742**	**0.241**	**0.631**	**0.672**
SSD/AdderNet	0.5730	0.8773	0.6835	0.2835	0.6322	0.4633
SSD/AdderNet w/MNMS	0.579	0.885	0.687	0.292	0.646	0.470

A. Traditional Machine Learning Methods

The results obtained using SVM, RF, SVM with MNMS (SVM$_{\text{NMS}}$), and RF with MNMS (RF$_{\text{NMS}}$) are shown in Tables V and VI. The increase in precision scores reached up to 25% as can be seen from the tables.

We have applied the MNMS technique also on both the SSD and our proposed SSD/AdderNet techniques (deep learning methods). The results for the Anti-UAV-RGB and IR datasets are shown in Tables VII and VIII, respectively.

As shown in Tables VII and VIII, the results are only slightly better with the MNMS. This is expected as the MNMS aims to solve the bias problem, which clearly exists in traditional machine learning techniques but does not exist in deep learning models. The main metric of comparison is the AP score. The MNMS in comparison to the SVM has,

TABLE VIII
MNMS on Deep Learning Using Anti-UAV-IR

	AP	$AP_{0.5}$	$AP_{0.75}$	AP_S	AP_M	AP_L
SSD	0.4997	0.8377	0.5392	0.3423	0.627	–
SSD w/MNMS	**0.501**	**0.838**	**0.539**	**0.343**	**0.629**	**–**
SSD/AdderNet	0.5402	0.841	0.6252	0.3800	0.6718	–
SSD/AdderNet w/MNMS	**0.5404**	**0.844**	**0.623**	**0.382**	**0.673**	–

TABLE IX
Full Drone-Vs-Bird Versus Filtered Drone-Vs-Bird (SSD/AdderNet)

Dataset	**AP**	$\mathbf{AP_{0.5}}$	$\mathbf{AP_{0.75}}$	$\mathbf{AP_S}$	$\mathbf{AP_M}$	$\mathbf{AP_L}$
Full	0.184	0.4849	0.083	**0.0963**	0.441	0.460
Filtered	**0.489**	**0.976**	**0.383**	–	**0.485**	**0.653**

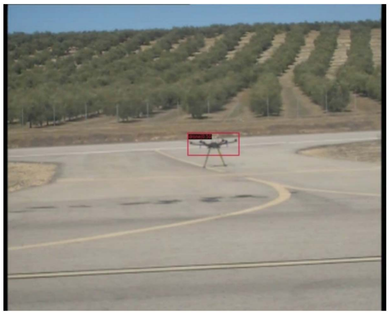

Fig. 12. Sample result: Drone-Vs-Bird.

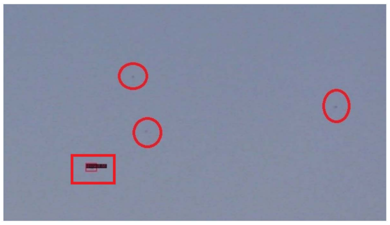

Fig. 13. Sample result: Drone-Vs-Bird with birds-1.

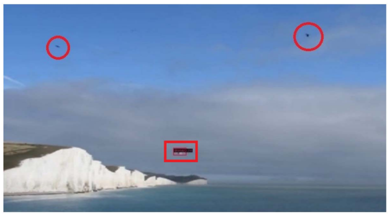

Fig. 14. Sample result: Drone-Vs-Bird with birds-2.

TABLE X
Full Anti-UAV-IR Versus Filtered Anti-UAV-IR (SSD/AdderNet)

Dataset	**AP**	$\mathbf{AP_{0.5}}$	$\mathbf{AP_{0.75}}$	$\mathbf{AP_S}$	$\mathbf{AP_M}$	$\mathbf{AP_L}$
Full	0.540	0.841	0.625	**0.3800**	0.672	–
Filtered	**0.683**	**0.989**	**0.826**	–	**0.685**	**–**

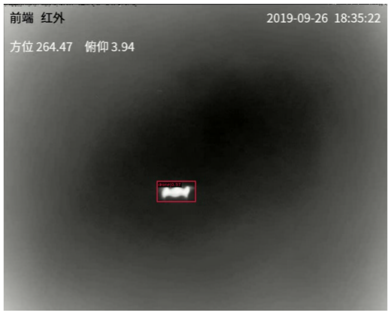

Fig. 15. Sample result: Anti-UAV-IR.

TABLE XI
Anti-UAV-RGB (SSD/AdderNet)(Full Dataset)

	AP	$\mathbf{AP_{0.5}}$	$\mathbf{AP_{0.75}}$	$\mathbf{AP_S}$	$\mathbf{AP_M}$	$\mathbf{AP_L}$
Results	0.578	0.884	0.687	0.299	0.635	0.465

in general, achieved higher scores. However, the MNMS merges bounding boxes instead of choosing the bounding box with the highest score based on a predefined Euclidean distance. It is highly possible that the algorithm had chosen two adjacent bounding boxes, which caused it to cover a larger area than what is required. That resulted in a minor degradation of performance as in the single case of SVM and SVM_{MNMS} for $AP_{0.5}$ of Table V. Adjusting the predefined Euclidean distance to a lower value can eliminate this problem

B. Deep Learning Methods With the Drone-Vs-Bird Dataset

The results using SSD/AdderNet on the full (unfiltered) and filtered Drone-Vs-Bird datasets are provided in Table IX. A sample detection using SSD/AdderNet is shown in Fig. 12. The results of all the deep learning methods on the full (unfiltered) Drone-Vs-Bird dataset are provided in Table XIII. In Figs. 13 and 14, we can clearly see that the model is detecting the presence of the drones without detecting the birds as false positives. That also demonstrates the ability of the model to detect small drone images without mixing them with the birds. Circles are drawn around the birds, while additional frames were drawn around the drones for clarification.

C. Deep Learning Methods With the Anti-UAV-IR Dataset

The results using SSD/AdderNet on the full (unfiltered) and filtered Anti-UAV-IR datasets are provided in Table X. A sample detection using SSD/AdderNet is shown in Fig. 15. The results of all the deep learning methods on the full (unfiltered) Anti-UAV-IR dataset are provided in Table XIV.

D. Deep Learning Methods With the Anti-UAV-RGB Dataset

The results using SSD/AdderNet on the Anti-UAV-RGB dataset are provided in Table XI. A sample detection using SSD/AdderNet is shown in Fig. 16. The results of all the deep learning methods on the Anti-UAV-RGB dataset are provided in Table XV.

Table XII shows the offline training times (in hours) and the frame rate in terms of second/image (seconds required to process one image). The system variables (computational resources and setting) were kept fixed in all the simulations. By comparing the proposed method (SSD/AdderNet) to its

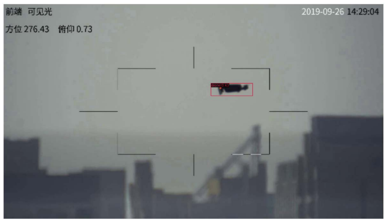

Fig. 16. Sample result: Anti-UAV-RGB.

TABLE XII
Training/Inference Time for Deep Learning Methods

Dataset	Model	Training Time (h)	Inference (Second/Image)
Drone-Vs-Bird	Faster-RCNN	58	0.0493
	SSD	**24**	**0.0297**
	YOLOv3	19.5	0.0240
	DETR	53	0.0521
	SSD/AdderNet	**20**	**0.0285**
Anti-UAV-IR	Faster-RCNN	63	0.0459
	SSD	**36**	**0.0291**
	YOLOv3	33	0.0253
	DETR	69	0.0483
	SSD/AdderNet	**33.5**	**0.0275**
Anti-UAV-RGB	Faster-RCNN	92	0.0481
	SSD	**34**	**0.0293**
	YOLOv3	27	.0242
	DETR	105	0.0524
	SSD/AdderNet	**32**	**0.0281**

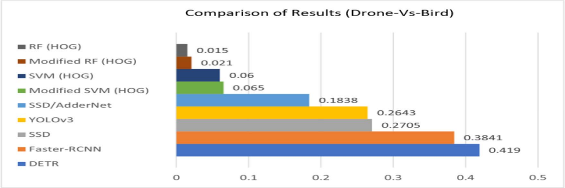

Fig. 17. Comparison of results (Drone-Vs-Bird).

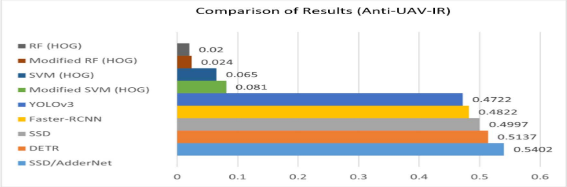

Fig. 18. Comparison of results (Anti-UAV-IR).

counterpart (SSD), we can see that the proposed method requires less training time in addition to less inference time to process a single image. That would prove the effectiveness of the complexity reduction that was obtained. If we are talking about critical conditions where decisions have to be made by the detection models in a few seconds, then the SSD/AdderNet would be preferable over the SSD. In those critical situations, every meter and every second counts.

VI. DISCUSSION AND ANALYSIS

Figs. 17– 19 provide visual comparisons of results using the main metric (mean average precision (mAP), IoU = 0.50:0.95, area = all) for the Drone-Vs-Bird dataset, Anti-UAV-IR dataset, and Anti-UAV-RGB dataset, respectively (all of them are the largest datasets available as open source with more than 100 000 samples in each one of them). As can be seen from Figs. 18 and 19, deep learning methods have significantly better performance than traditional machine learning methods even though traditional machine learning models were tested on filtered datasets. In Fig. 17, traditional machine learning methods (SVM and RF) were not tested on the (Anti-UAV-RGB) since it was proved by the previous two datasets that they have lower performance compared to deep learning methods.

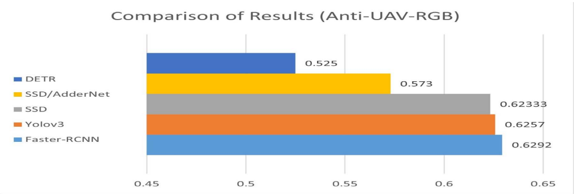

Fig. 19. Comparison of results (Anti-UAV-RGB).

The performance of traditional machine learning techniques was not satisfactory, and it was noticed that these methods are biased toward the size of the sliding window. Therefore, an MNMS was proposed to tackle this issue. The performance of traditional machine learning with the MNMS has slightly improved (a maximum of 25% improvement was reached in some cases). This proves that combining bounding boxes with the highest confidence score after eliminating the possible wrong detections based on Euclidean distance is better than selecting the bounding box with the highest score. Both the SVM and RF object detectors perform decently with less complex datasets, where the objects within the images have approximately similar sizes. The poor performance of SVM and RF object detectors can be a result of them being nonparametric models (complexity increases as the number of training samples increases). Moreover, the generalization of traditional machine learning models to include different object sizes can be computationally expensive. Compared to deep learning techniques, traditional machine learning trains a classifier using handcrafted features; then, a sliding window is used at the testing phase to detect objects. On the other hand, deep learning models, such as SSD, have a localization loss at each iteration that penalizes wrong bounding boxes; consequently, the model generalizes better.

As can be seen from Tables XIII– XV, the performance of all the deep learning models when tested with Anti-UAV-IR and Anti-UAV-RGB images is much better than performance with Drone-Vs-Bird images. This difference in performance between the Anti-UAV-IR and Anti-UAV-RGB datasets can be justified by the fact that these datasets are less complex than the Drone-Vs-Bird dataset as they contain clearer images with larger drone sizes. Moreover, the Drone-Vs-Bird dataset contains eight drone types, whereas the Anti-UAV-IR and Anti-UAV-RGB datasets contain four drone types. In addition, the Drone-Vs-Bird dataset contains smaller size objects compared with the other datasets. All of that contributes to the inferiority of performance with the Drone-Vs-Bird dataset.

As a matter of fact, there is no straightforward way to predict which object detector will perform better. The

TABLE XIII
Verification of Results (Drone-Vs-Bird)

Model		**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Run 6**	**Mean**	**Std**	**CI**
SSD	**AP**	0.270	0.274	0.276	0.269	0.269	0.265	0.2705	0.0039370039	0.2705 ± 0.00292
	AP$_{0.5}$	0.620	0.626	0.631	0.620	0.615	0.634	0.6243	0.0072846871	0.6243 ± 0.0054
	AP$_{0.75}$	0.172	0.175	0.183	0.176	0.175	0.153	0.1723	0.010152175	0.1723 ± 0.00752
	AP$_{S}$	0.164	0.168	0.169	0.162	0.160	0.167	0.1650	0.0035777088	0.165 ± 0.00265
	AP$_{M}$	0.521	0.527	0.526	0.521	0.517	0.497	0.5182	0.010998485	0.5182 ± 0.00815
	AP$_{L}$	0.646	0.655	0.643	0.643	0.656	0.635	0.6463	0.0079916623	0.6463 ± 0.00592
Faster-RCNN	**AP**	0.476	0.309	0.356	0.392	0.451	0.321	0.3841	0.0079916623	0.3841 ± 0.00592
	AP$_{0.5}$	0.734	0.622	0.721	0.621	0.689	0.724	0.6852	0.051572926	0.6852 ± 0.0382
	AP$_{0.75}$	0.507	0.282	0.521	0.412	0.421	0.501	**0.4407**	0.090418287	0.4407 ± 0.067
	AP$_{S}$	0.360	0.181	0.355	0.310	0.305	0.341	**0.3087**	0.066545223	0.3087 ± 0.0493
	AP$_{M}$	0.792	0.620	0.712	0.601	0.621	0.728	**0.6790**	0.076404188	0.679 ± 0.0566
	AP$_{L}$	0.765	0.699	0.695	0.731	0.701	0.732	**0.7205**	0.027259861	0.7205 ± 0.0202
SSD-AdderNet	**AP**	0.181	0.179	0.180	0.182	0.181	0.200	0.1838	0.0079854034	0.18383333 ± 0.00639
	AP$_{0.5}$	0.489	0.455	0.487	0.488	0.480	0.533	0.4887	0.025208464	0.48866667 ± 0.0202
	AP$_{0.75}$	0.078	0.080	0.079	0.077	0.076	0.105	0.0825	0.011113055	0.0825 ± 0.00889
	AP$_{S}$	0.097	0.092	0.091	0.096	0.098	0.104	0.0963	0.0046761808	0.096333333 ± 0.00374
	AP$_{M}$	0.436	0.432	0.435	0.436	0.431	0.473	0.4405	0.016059265	0.4405 ± 0.0128
	AP$_{L}$	0.454	0.454	0.453	0.450	0.449	0.495	0.4592	0.017679555	0.45916667 ± 0.0141
YOLOv3	**AP**	0.391	0.238	0.238	0.237	0.241	0.241	0.2643	0.0038534667	0.2643 ± 0.00308
	AP$_{0.5}$	0.782	0.621	0.630	0.617	0.614	0.613	0.6462	0.0044661667	0.6462 ± 0.00357
	AP$_{0.75}$	0.338	0.136	0.121	0.122	0.124	0.133	0.1623	0.0074434667	0.1623 ± 0.00592
	AP$_{S}$	0.318	0.163	0.165	0.164	0.166	0.165	0.1902	0.0039229667	0.1902 ± 0.00314
	AP$_{M}$	0.664	0.503	0.488	0.500	0.495	0.504	0.5257	0.0046274667	0.5257 ± 0.0037
	AP$_{L}$	0.696	0.639	0.628	0.632	0.644	0.644	0.6472	0.00061376667	0.6472 ± 0.000491
DETR	**AP**	0.501	0.356	0.412	0.422	0.399	0.424	**0.4190**	0.047277902	0.419 ± 0.0378
	AP$_{0.5}$	0.846	0.806	0.823	0.812	0.832	0.813	**0.8220**	0.014926487	0.822 ± 0.0119
	AP$_{0.75}$	0.506	0.239	0.234	0.261	0.291	0.256	0.2978	0.10394502	0.29783333 ± 0.0832
	AP$_{S}$	0.398	0.250	0.267	0.266	0.257	0.274	0.2853	0.055827114	0.28533333 ± 0.0447
	AP$_{M}$	0.760	0.621	0.652	0.627	0.651	0.683	0.5257	0.66566667	0.66566667 ± 0.0409
	AP$_{L}$	0.733	0.694	0.725	0.696	0.701	0.718	0.7112	0.016388004	0.71116667 ± 0.0131

TABLE XIV
Verification of Results (Anti-UAV-IR)

Model		**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Run 6**	**Mean**	**Std**	**CI**
SSD	**AP**	0.503	0.497	0.501	0.498	0.497	0.502	0.4997	0.0026583203	0.49966667 ± 0.00213
	AP$_{0.5}$	0.840	0.835	0.837	0.839	0.840	0.835	0.8377	0.0023380904	0.83766667 ± 0.00187
	AP$_{0.75}$	0.545	0.534	0.541	0.536	0.542	0.537	0.5392	0.0041673333	0.53916667 ± 0.00333
	AP$_{S}$	0.350	0.333	0.338	0.345	0.349	0.339	0.3423	0.0067428975	0.34233333 ± 0.0054
	AP$_{M}$	0.626	0.628	0.629	0.625	0.628	0.626	0.627	0.0015491933	0.627 ± 0.00124
	AP$_{L}$	–	–	–	–	–	–	–	–	–
Faster-RCNN	**AP**	0.507	0.477	0.473	0.477	0.481	0.478	0.4822	0.00015456667	0.48216667 ± 0.000124
	AP$_{0.5}$	0.780	0.816	0.808	0.812	0.811	0.814	0.8068	0.00018016667	0.80683333 ± 0.000144
	AP$_{0.75}$	0.584	0.505	0.500	0.507	0.513	0.507	0.5193	0.0010210667	0.51933333 ± 0.000817
	AP$_{S}$	0.309	0.313	0.304	0.311	0.314	0.311	0.3103	1.2666667E-5	0.31033333 ± 0.0000101
	AP$_{M}$	0.674	0.612	0.614	0.614	0.616	0.614	0.624	0.0006016	0.624 ± 0.000481
	AP$_{L}$	–	–	–	–	–	–	–	–	–
SSD-AdderNet	**AP**	0.541	0.541	0.542	0.539	0.540	0.538	**0.5402**	0.0014719601	0.54016667 ± 0.00118
	AP$_{0.5}$	0.839	0.843	0.842	0.841	0.839	0.842	**0.841**	0.0016733201	0.841 ± 0.00134
	AP$_{0.75}$	0.626	0.625	0.625	0.626	0.624	0.625	**0.6252**	0.00075277265	0.62516667 ± 0.000602
	AP$_{S}$	0.377	0.383	0.379	0.378	0.381	0.382	**0.3800**	0.0023664319	0.38 ± 0.00189
	AP$_{M}$	0.673	0.673	0.672	0.671	0.672	0.670	0.6718	0.0011690452	0.67183333 ± 0.000935
	AP$_{L}$	–	–	–	–	–	–	–	–	–
YOLOv3	**AP**	0.463	0.478	0.474	0.476	0.465	0.477	0.4722	0.0064935866	0.47216667 ± 0.0052
	AP$_{0.5}$	0.773	0.814	0.820	0.810	0.820	0.811	0.8080	0.017674841	0.808 ± 0.0141
	AP$_{0.75}$	0.503	0.507	0.500	0.502	0.501	0.504	0.5028	0.0024832774	0.50283333 ± 0.00199
	AP$_{S}$	0.274	0.332	0.335	0.331	0.332	0.330	0.3223	0.023737453	0.32233333 ± 0.019
	AP$_{M}$	0.640	0.610	0.603	0.609	0.611	0.612	0.6142	0.013044795	0.61416667 ± 0.0104
	AP$_{L}$	–	–	–	–	–	–	–	–	–
DETR	**AP**	0.516	0.515	0.510	0.512	0.513	0.516	0.5137	0.0024221203	0.51366667 ± 0.00194
	AP$_{0.5}$	0.770	0.770	0.765	0.768	0.766	0.769	0.7680	0.0020976177	0.768 ± 0.00168
	AP$_{0.75}$	0.607	0.606	0.603	0.605	0.602	0.603	0.6043	0.0019663842	0.60433333 ± 0.00157
	AP$_{S}$	0.295	0.293	0.292	0.294	0.291	0.290	0.2925	0.0018708287	0.2925 ± 0.0015
	AP$_{M}$	0.694	0.694	0.690	0.692	0.692	0.693	**0.6925**	0.0015165751	0.6925 ± 0.00121
	AP$_{L}$	–	–	–	–	–	–	–	–	–

performance may vary depending on the settings and the data variations. For the purpose of real-life applications, the best selling point is to balance accuracy and detection speed. Based on Drone-Vs-Bird results in Table XIII, DETR performed the best among other object detectors. Therefore, DETR is a much better two-stage detector than Faster-RCNN for the Drone-Vs-Bird dataset. In terms of speed, DETR has a much simpler architecture than Faster-RCNN as it eliminates the use of a region proposal network, it does not use NMS, and it has a transformer architecture. In terms of the tightness of bounding boxes, Faster-RCNN scored higher than DETR. Both DETR and Faster-RCNN have a minimum image height of 800 pixels [7], whereas SSD and YOLOv3 have feature maps that degrade the resolution, hence making it harder to detect smaller objects. Given that the Drone-Vs-Bird dataset contains a large number of small objects, SSD and YOLOv3 are expected to perform worse than the two-stage detectors. Both SSD and YOLOv3 seem to be performing very well with large objects. This can be justified by the fact that degrading the resolution of large objects does not cause spatial information to degrade significantly. With AdderNet integrated

TABLE XV
Verification of Results (Anti-UAV-RGB)

Model		**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Run 6**	**Mean**	**Std**	**CI**
SSD	**AP**	0.622	0.616	0.644	0.615	0.632	0.611	0.62333	0.012484657	0.62333333 ± 0.00999
	AP$_{0.5}$	0.938	0.944	0.964	0.922	0.912	0.936	0.9360	0.018022209	0.936 ± 0.0144
	AP$_{0.75}$	0.745	0.722	0.753	0.713	0.744	0.751	0.7380	0.016492423	0.738 ± 0.0132
	AP$_{S}$	0.222	0.222	0.243	0.212	0.211	0.220	0.2217	0.011535453	0.22166667 ± 0.00923
	AP$_{M}$	0.630	0.631	0.633	0.629	0.631	0.631	0.6308	0.0013291601	0.63083333 ± 0.00106
	AP$_{L}$	0.672	0.671	0.677	0.662	0.670	0.671	0.6705	0.0048476799	0.6705 ± 0.00388
Faster-RCNN	**AP**	0.630	0.631	0.626	0.631	0.627	0.630	**0.6292**	0.0021369761	0.6705 ± 0.00171
	AP$_{0.5}$	0.965	0.961	0.952	0.969	0.971	0.958	**0.9627**	0.0071180522	0.96266667 ± 0.0057
	AP$_{0\cdot75}$	0.754	0.751	0.748	0.744	0.756	0.753	**0.751**	0.0043817805	0.751 ± 0.0057
	AP$_{S}$	0.229	0.228	0.219	0.212	0.227	0.228	0.2238	0.0068532231	0.22383333 ± 0.00548
	AP$_{M}$	0.638	0.653	0.621	0.640	0.629	0.611	0.632	0.015758596	0.632 ± 0.0126
	AP$_{L}$	0.657	0.655	0.649	0.641	0.658	0.657	0.6528	0.0066458007	0.65283333 ± 0.00532
SSD-AdderNet	**AP**	0.578	0.577	0.568	0.571	0.574	0.570	0.5730	0.004	0.573 ± 0.0032
	AP$_{0.5}$	0.880	0.884	0.862	0.877	0.872	0.889	0.8773	0.0095008772	0.87733333 ± 0.0076
	AP$_{0.75}$	0.681	0.687	0.677	0.689	0.681	0.686	0.6835	0.0045497253	0.6835 ± 0.00364
	AP$_{S}$	0.299	0.288	0.291	0.296	0.297	0.230	0.2835	0.026523574	0.2835 ± 0.0212
	AP$_{M}$	0.635	0.642	0.621	0.633	0.631	0.631	**0.6322**	0.0068239773	0.63216667 ± 0.00546
	AP$_{L}$	0.465	0.466	0.461	0.460	0.468	0.460	0.4633	0.00344	0.46 ± 0.003
YOLOv3	**AP**	0.623	0.626	0.613	0.643	0.622	0.627	0.6257	0.0098319208	0.62566667 ± 0.00787
	AP$_{0.5}$	0.961	0.962	0.971	0.955	0.959	0.963	0.9618	0.0053072278	0.96183333 ± 0.00425
	AP$_{0.75}$	0.735	0.732	0.733	0.728	0.731	0.729	0.7313	0.0025819889	0.73133333 ± 0.00207
	AP$_{S}$	0.317	0.317	0.325	0.311	0.311	0.319	**0.3223**	0.31666667	0.31666667 ± 0.00422
	AP$_{M}$	0.626	0.622	0.623	0.621	0.629	0.624	0.6242	0.002927	0.6242 ± 0.00234
	AP$_{L}$	0.676	0.672	0.671	0.677	0.679	0.670	**0.6742**	0.0036560452	0.67416667 ± 0.00293
DETR	**AP**	0.529	0.522	0.521	0.524	0.529	0.525	0.5250	0.0034058773	0.525 ± 0.00273
	AP$_{0.5}$	0.923	0.921	0.921	0.922	0.925	0.919	0.9218	0.0020412415	0.92183333 ± 0.00163
	AP$_{0.75}$	0.580	0.581	0.589	0.577	0.572	0.580	0.5798	0.0055647701	0.57983333 ± 0.00445
	AP$_{S}$	0.004	0.014	0.024	0.05	0.009	0.004	0.0175	0.017592612	0.0175 ± 0.0141
	AP$_{M}$	0.515	0.512	0.511	0.505	0.509	0.522	0.5123	0.0057850382	0.51233333 ± 0.00463
	AP$_{L}$	0.632	0.623	0.631	0.628	0.629	0.631	0.629	0.003286	0.63 ± 0.003

into SSD for the Drone-Vs-Bird dataset, the performance of the model became lower, especially with smaller objects as expected. However, for medium and large objects, the performance was fairly close to other methods. One compelling reason for choosing the SSD to integrate with AdderNet filters is its single-stage detection architecture, featuring convolutional layers that can be efficiently replaced with the AdderNet filters. This substitution process results in a notably reduced model complexity compared to other methods. As a consequence, SSD/AdderNet proves to be a more fitting option than the original SSD for real-time detections when considering computational efficiency and speed. The SSD was selected over the YOLOv3 (another single-stage detector) because the SSD showed better precision results with respect to the RGB datasets.

By comparing the IR results in Table XIV, most deep learning models showed close and fairly good precisions. No degradation in performance caused by the reduction of resolution was noticed in the SSD and the YOLOv3. Hence, it can be concluded that high-resolution images are not an important factor for drone detection. Moreover, results suggest that one-stage detectors are a better option when dealing with IR images as they require less training and inference times making them more suitable for real-life applications. Interestingly, SSD/AdderNet outperformed all other deep learning techniques when tested with the large Anti-UAV-IR dataset. It is also important to note that features produced by AdderNet filters are clusters of classes [6], whereas features from normal convolutional layers are divided by angles [6]. Given the high contrast in IR images (the backgrounds are almost uniform or extremely dark in all the images, in addition to the fact that drones are extremely bright), and that only the VGG-16 (backbone) of the SSD was replaced with the AdderNet filters, it can be concluded that features produced by AdderNet filters are better than the features produced by normal convolutional filters when dealing with IR images.

For the case of the Anti-UAV-RGB dataset, the proposed SSD/AdderNet scored an mAP of 0.5730, where the highest score of 0.6292 was achieved by Faster-RCNN. Upon comparing the proposed SSD/AdderNet with its counterpart, SSD, in Tables XIII–XV, it becomes evident that the proposed SSD/AdderNet outperforms the SSD with respect to the IR dataset. However, it falls slightly behind SSD when considering the RGB datasets. Notably, the SSD/AdderNet demonstrates a remarkable advantage in terms of complexity, boasting a 43.42% reduction in the number of multiplications required for its operations. The main selling point here lies in the reduced complexity of the convolutional layers, a crucial factor for real-time applications that uses detectors as fundamental components. According to Table XII, the inference times for the SSD/AdderNet were less than its counterpart (the SSD) for the three datasets. Training times of the SSD/AdderNet for the three datasets were also lesser.

It is worth mentioning that the size of objects present in the datasets significantly affects the performance of an object detector, as shown in Tables IX and X. The SSD/AdderNet performance increased by 37.5% on the main mAP metric (when using larger object windows). Finally, the standard deviations for the repeated experiments, as shown in Tables XIII–XV, are small, and the resulting confidence intervals are tight, which implies consistency when using deep learning methods with the aforementioned datasets. As a summary, we can conclude that the proposed model (SSD/AdderNet) was able to achieve the best precision results with respect to the IR dataset and falls a little bit behind with respect to the RGB datasets while exhibiting significant reductions in terms of complexity along with training and inference times. Inference time is highly crucial for real-time applications if the detection was performed on long sequences of images.

VII. CONCLUSION AND FUTURE WORK

The aim of this work was to investigate the tradeoffs between computational complexity and precision when detecting drones in real-time applications. We sought to explore and compare various methods to enhance the detection process focusing on traditional machine learning methods, such as SVM and RF, and deep learning methods, such as two- and one-stage detectors. We proposed an MNMS method to enhance the detection precision for both traditional machine learning methods and deep learning methods. As we saw, the improvements were more substantial with the traditional machine learning methods. In addition, we proposed a deep learning approach (SSD/AdderNet) that places emphasis on reducing computational complexity while still achieving precision values comparable to its counterparts (the deep learning methods in general and the SSD in particular). Comparing the proposed model with other one-stage detectors (such as SSD and YOLOv3), the performance was a little bit behind based on RGB datasets. However, it outperformed both the one- and two-stage detectors (such as Faster-RCNN and DETR) based on the Anti-UAV-IR dataset. Six different runs showed statistically consistent performance of the simulations we had. Moreover, simulations showed that the proposed model performed better for mid-range applications (where drone sizes were medium/large). Finally, the SSD/AdderNet proved to have faster inference and less training time as opposed to its counterpart (the SSD).

Future work will include synthesizing new datasets for “drones” and “birds” to perform multiclass detection and classification, in addition to integrating the AdderNet layer with two-stage detectors seeking high precision values and less complexity at the same time.

REFERENCES

[1] A. Mairaj, A. I. Baba, and A. Y. Javaid, “Application specific drone simulators: Recent advances and challenges,” *Simul. Model. Pract. Theory*, vol. 94, pp. 100–117, 2019.

[2] I. Guvenc, F. Koohifar, S. Singh, M. L. Sichitiu, and D. Matolak, “Detection, tracking, and interdiction for amateur drones,” *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 75–81, Apr. 2018.

[3] P.-Y. Lagrave and F. Barbaresco, “Introduction to robust machine learning with geometric methods for defense applications,” Jul. 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03309807

[4] P.-Y. Lagrave and F. Barbaresco, “Hyperbolic equivariant convolutional neural networks for fish-eye image processing,” Feb. 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03553274

[5] T. Nagano et al., “Image processing device that removes motion blur from an image and method of removing motion blur from an image,” U.S. Patent 7 750 943, Jul. 6, 2010.

[6] H. Chen et al., “AdderNet: Do we really need multiplications in deep learning?,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1468–1477.

[7] B. K. Isaac-Medina, M. Poyser, D. Organisciak, C. G. Willcocks, T. P. Breckon, and H. P. Shum, “Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 1223–1232.

[8] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” in *Proc. IEEE*, vol. 111, no. 3, Mar. 2023, pp. 257–276, doi: 10.1109/JPROC.2023.3238524.

[9] Y. Pang, Y. Yuan, X. Li, and J. Pan, “Efficient hog human detection,” *Signal Process.*, vol. 91, no. 4, pp. 773–781, 2011.

[10] F. Han, Y. Shan, R. Cekander, H. S. Sawhney, and R. Kumar, “A two-stage approach to people and vehicle detection with HOG-based SVM,” in *Proc. Perform. Metrics Intell. Syst. Workshop*, 2006, pp. 133–140.

[11] C. Savas and F. Dovis, “The impact of different kernel functions on the performance of scintillation detection based on support vector machines,” *Sensors*, vol. 19, no. 23, 2019, Art. no. 5219.

[12] S. Sedai, P. K. Roy, and R. Garnavi, “Right ventricle landmark detection using multiscale HOG and random forest classifier,” in *Proc. IEEE 12th Int. Symp. Biomed. Imag.*, 2015, pp. 814–818.

[13] A. I. Salhi, M. Kardouchi, and N. Belacel, “Fast and efficient face recognition system using random forest and histograms of oriented gradients,” in *Proc. Int. Conf. Biometrics Special Int. Group*, 2012, pp. 1–11.

[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[16] W. Liu et al., “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[17] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, vol. 28, pp. 91–99.

[18] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, “Green AI,” *Commun. ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[19] L. Wei, W. Cui, Z. Hu, H. Sun, and S. Hou, “A single-shot multi-level feature reused neural network for object detection,” *Vis. Comput.*, vol. 37, no. 1, pp. 133–142, 2021.

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

[21] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[22] A. Coluccia et al., “Drone-vs-bird detection challenge at IEEE AVSS2021,” in *Proc. IEEE 17th Int. Conf. Adv. Video Signal Based Surveill.*, 2021, pp. 1–8.

[23] N. Jiang et al., “Anti-UAV: A large multi-modal benchmark for UAV tracking,” 2021, *arXiv:2101.08466*.

Mohamad Kassab received the B.S. degree in electrical and electronics engineering from the American University of Sharjah, Sharjah, UAE, in 2021, and the master’s degree in machine learning in 2023 from the Mohamed Bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE, where he is currently working toward the Ph.D. degree.

He has worked in collaboration with Sorbonne University, Abu Dhabi, to propose new algorithms that reduce the complexity of object detection. His research work focuses on reducing the complexity of deep neural networks, especially in the field of computer vision.

Raed Abu Zitar (Member, IEEE) received the Ph.D. degree in computer engineering from Wayne State University, Detroit, MI, USA, in 1993.

He is currently a Senior Research Scientist with the Thales Endowed Chair of Excellence, Sorbonne Centre for Artificial Intelligence, Sorbonne University, Abu Dhabi, UAE. He has contributed to more than 120 publications and is involved in many funded projects along with graduate students’ supervision. His research interests include machine learning, neural networks, cybersecurity, and stochastic processing.

Frederic Barbaresco received the M.S. degree from SUPELEC, France, in 1991.

He is a Senior Thales Expert in Artificial Intelligence with the Technical Department, Thales Land and Air Systems, Thales Group, Paris, where he is also a Smart Sensors Segment Leader with the Thales Corporate Technical Department (Key Technology Domain “Processing, Control and Cognition”). He is a Thales Representative with the AI Expert Group, AeroSpace and Defense Industries Association of Europe, Brussels, Belgium. He is the author of more than 200 scientific publications and more than 20 patents.

Mr. Barbaresco was the recipient of the 2014 Aymée Poirson Prize of the French Academy of Science for the application of science to industry and the Ampère Medal. He is an Emeritus Member of the society of electricity and electronics engineering (SEE) and the President of the SEE information systems and informatics club (ISIC) club “Information and Communication Systems Engineering.” He is a French MC representative of European COST CaLISTA. He is a General Chair of several elite and highly specialized conferences and a Guest Editors of Special Issues “Lie Group Machine Learning and Lie Group Structure Preserving Integrators.”

Amal El Fallah Seghrouchni received the Ph.D. degree from Sorbonne University, France, in 1991.

She is the Head of Ai Movement—International Artificial Intelligence Center of Morocco, Mohammed VI Polytechnic University, Rabat, Morocco, and a Full Professor with the Computer Lab of Paris 6, CNRS, Sorbonne University, Paris, France. She is the author of more than 100 publications and supervised more than 33 Ph.D. students. Her research interests include multiagent systems, artificial intelligence systems, and ambient intelligence.

Ms. Seghrouchni is a Member of the World Commission on the Ethics of Scientific Knowledge and Technology, United Nations Educational, Scientific and Cultural Organization. She was the General Chair of 2020 International Conference on Autonomous Agents and Multiagent Systems, Auckland, New Zealand.