JDST., vol. 4, no. 4, p53-p70, 2022
https://doi.org/10.46713/jdst.004.04
www.jdst.eu, e-ISSN 2534-9813
Research Article

Small Drone Detection with Image Tiling in Counter Autonomous System Context

Guillaume Gagné[a], Jean-Philippe Mercier[b], Vincent Paquin[b] and Yves DeVillers[a]

[a] *Defence Research and Development Canada, Quebec, Canada*
https://www.canada.ca/en/defence-research-development.html

[b] *Aerex Avionics, Quebec, Canada*
http://www.aerex.ca/

ABSTRACT

The increasing availability and versatility of drones in the last few years have made them an interesting tool to disrupt privacy, safety and security: they are small, fast and have sufficient payload to carry dangerous items. Because of their size and speed, they can be hard to detect and track. In this paper, we propose an efficient approach based on YOLOv3 [1] to detect drones in high resolution images by integrating an image tiling strategy. With this approach, we finished in second place in the 2020 Drone-vs-Bird Challenge (0.2% behind the winners). Finally, we also describe the ALEXIS system we have developed that can detect and classify using this approach, and track drones in real time from multiple cameras in different bands.

ARTICLE INFO

RECEIVED: 09 Oct 2021
REVISED: 10 Nov 2021
ACCEPTED: 30 Nov 2021
ONLINE: 12 DEC 2021

KEYWORDS

Deep learning, drone detection, image tiling

I. INTRODUCTION

Unmanned Aerial Systems (UAS), commonly called drones, have seen an exponential growth in their proliferation on the commercial market. Grand View

Research mentioned that the ”global commercial drone market (GCDM) size was valued at USD 5.80 billion, with an estimated 274.6 thousand units sold, in 2018” [2]. The same group also reported that the “GCDM size is anticipated to reach USD 129.23 billion by 2025” [3].

Commercial drones are still evolving. For example, drones costing less than $2000 now possess some payload capacity along with sensitive, high-quality and image stabilized cameras. They constitute a potential threat for the safety and security of the general population and government agencies around the world [4], since they require very little knowledge or skills to deploy and can be weaponized. There have been many occurrences around the world where drones have been used for that purpose. For instance, in 2018, a swarm of 13 armed drones built from gathered parts attacked a Russian base in Syria [5].

For such reason, drones are considered an important threat by Canadian Armed Forces (CAF) and they are looking at different solutions to be protected against this new threat. Among other things, they are evaluating the strengths and limitations of their actual equipment and different industrial solutions, which may use technologies such as radar, radio frequencies, electro-optic and infrared (IR) sensors, etc.

In support to the CAF, Defence Research and Development Canada (DRDC) was mandated to evaluate the performance of electro-optical and infrared systems to detect, recognize, identify, and track micro and mini Class 1 (NATO UAS classification [6]) drones and to propose enhancements to these systems. The main objective of the CAF is to detect drones far enough to have time to engage it. However, the farther they are, the smaller they appear in images. It thus makes it more difficult to detect them with standard image processing techniques.

In this work, we propose a solution to overcome the problem of detecting small drones in colour images, as shown in Figure 1. Some related works are presented in section 2. In section 3, we describe the architecture of YOLOv3 (You Only Look Once) [1] along with specific details of our implementation. In section 4, we specify the data and augmentation techniques we have used to train our approach. In section 5, we perform several experiments to show the performance of our approach under different circumstances and report the official results of the 2020 Drone-vs-Bird Challenge in which we participated. In section 6, we describe our fully integrated system for detection and tracking with multiple cameras. Finally, in section 7, we conclude by discussing some limitations of our approach and propose potential directions for future work.

Figure 1: Our approach based on YOLOv3 [1] is able to detect small drones in high resolution images by using an image tiling strategy. Here are some examples of successful detections made by our approach (green: predictions, red: ground-truths). Only a slight part of the red box can be seen in the left image because the green box masks it.

II. Related Work

One of the main challenges of drone detection is the relatively small size of drones in images, which makes them hardly distinguishable from other flying objects such as birds, planes, etc. Different interesting solutions have been proposed in the literature: using temporal information, increasing image resolution, generating artificial datasets, etc. In this section, we review some of them.

Since 2012, when AlexNet [7] won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) by a wide margin, most of the computer vision research and state-of-the-art approaches have been based on deep learning. The problem of localizing drones in colour images has not been an exception [8]–[13]. For instance, Nalamati et al. [9] and Saqib et al. [13] evaluated the performance of different object state-of-the-art detectors such as Faster Region Based Convolutional Neural Networks (R-CNN) [14] and Single shot multibox detector (SSD) [15] on images coming from the Drones-vs-Bird Challenge [16]. According to [13], they evaluated different backbone networks for Faster R-CNN, but only evaluated their networks on the training set. Their best reported mean Average Precision (mAP) score was 0.66. They concluded that training for an extra class (birds) instead of training only for drones would likely decrease the number of false positives made by their networks. Likewise, Nalamati et al. [9] compared the performance of Faster R-CNN and SSD, with Faster R-CNN being the clear winner with a mAP of 0.49 on the test set (images not seen in training). As a comparison, our approach achieved a mAP superior to 0.90 when using similar data splits (see section 5.1).

A common limitation of object detectors is that they do not exploit temporal information. In the case of drones and other flying objects, using temporal information would allow networks to distinguish moving/flying objects from still ones and differentiate them by their flying patterns. The winning approach [17] of the 2019 Drone-vs-Bird Challenge [16], for instance, was based on this principle.

They first used multiple frames in input to detect moving objects, then classified them using a Residual Network (ResNet) [18] and filtered the predictions using spatio-temporal information.

To tackle the problem of small objects, the runner-ups of the 2019 Drone-vs-Bird Challenge [16] proposed to use a super resolution network [19] to increase the spatial resolution of images by a factor of 2, which significantly boosted the recall performance of their detector. This technique is an interesting way to improve detection performances, but has the downside of slowing the approach.

Lack of annotated data for drone detection has been a major limitation for a while. The 80 annotated videos (more than 100k frames) in the Drone-vs-Bird Challenge [16] is partially solving that problem. The remaining problem is that it is difficult to acquire image sequences in different settings, since it requires changing physical location and it is now restricted in many places to fly drones. This lack of diversity in datasets can make the models prone to overfit (memorize the settings) and perform poorly when using the model elsewhere of in different weather conditions.

To overcome this problem, a simple solution can be to generate synthetic datasets, which has shown in the literature to be a good way to boost the performance of object detectors [20]–[23]. For instance, according to [20] they propose a simple strategy of cropping objects of interest and pasting them on real images with different blending techniques to reduce the artefacts at the edge of objects. Aker and Kalkan [11] have used a similar technique to generate a detection dataset for drones and birds. Similarly, He et al. [18] have generated a classification dataset to distinguish drones from birds. They first detect the objects either with background subtraction if the camera is fixed or with a Region Proposal Network (RPN) if the camera is moving. RPN is the first part of Faster R-CNN and it predicts regions likely to contain objects (independent of their class).

III. Network Implementation

Our architecture is based on the single-stage object detector YOLOv3 [1]. We have chosen this detector because of its good performance and fast processing compared to two-stage approaches.

In YOLOv3, detections are made at 3 different resolutions (1/32, 1/16 and 1/8 of the original resolution) using 3 predefined anchors for each resolution (10 x 13, 16 x 30, 33 x 23, 30 x 61, 62 x 45, 59 x 119, 116 x 90, 156 x 198, 373 x 326). For each of these anchors, YOLO predicts objectness scores, class scores and size/location offsets.

We employed the public PyTorch implementation of YOLOv3 with Spatial Pyramid Pooling (YOLOv3-SPP) made available by Ultralytics [24]–[25]. As shown in Figure 2, the SPP layer applies different max pooling layers of different size in parallel and concatenates their outputs into feature maps of the same size as the

input by using appropriate padding. In this implementation, SPP is applied to the output of the backbone before the YOLO detection layers.

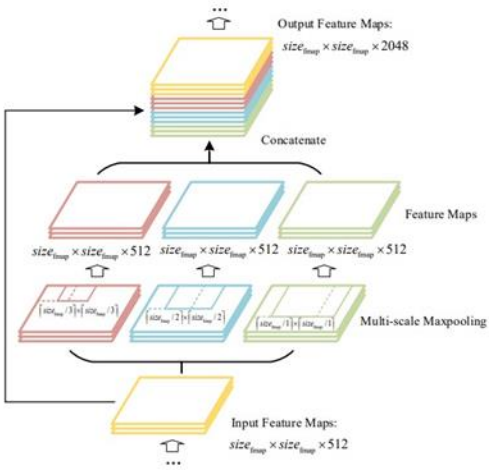


Figure 2: Spatial Pyramid Pooling (SPP) for YOLO [25]. Three max-pooling layers of different size process the input in parallel and keep the original resolution with appropriate padding. The output is a concatenation of the three processed feature maps and the input.

We selected the version pretrained on the Microsoft Coco dataset [26]. We then trained the network for a certain number of epochs (depending on the size of the dataset for which we trained for). For Drone-vs-Bird, we trained for 5 epochs with Stochastic Gradient Descent (SGD) at an initial learning rate of 0.001 and drops of 50% at epochs 3 and 4. For other datasets, the number of epochs was adjusted to get a similar number of training iterations and the learning rate was multiplied at 60% and 80% of the epochs. The network was trained using a single class (drone) and was optimized both for objectness (using the binary cross-entropy loss) and bounding box regression (using the complete Intersection loss [27] instead of the sum of squared error loss proposed in YOLOv3 [1]).

IV. Data

In this section, we describe the Drone-vs-Bird dataset and other datasets that we have generated or acquired. We also detail our proposed image tiling strategy and enumerate the data augmentation strategies we have employed.

A. Datasets

Since we participated in the 2020 Drones-vs-Bird challenge, the main dataset we have used is the training dataset of the challenge. One of the 77 annotated sequences appeared to have bad annotations, so we removed it from training. Images were extracted from videos at their original frame rate for a total of around

106k frames. In the dataset, only drones were annotated (birds only appear as background).

Figure 3: Example of an infrared image acquired by the DRDC. Drone is shown with a bounding box.

In addition to this dataset, we have followed the procedure of Cut, Paste and Learn [20] to generate 26.5k synthetic images. We have manually segmented objects of around 100 images of drones and birds acquired from Google Images which were then pasted on different outdoor backgrounds from Kaggle[1] using different blending techniques. Images were selected to offer a wide variety of backgrounds and targets (birds and drones). Drones images from public domain or licensable[2] were selected randomly on Google Images, but those with white background were prioritized in our selection to ease segmentation and synthetic image creation. Since the Drone-vs-Bird did not include annotations for birds, we did not generate annotations for them neither in this synthetic dataset.

Finally, we have acquired close to 35k images with 3 different infrared cameras (e.g. in Figure 3) and colour cameras equipped with 2 different lenses to get different zoom levels during the NATO SET-260 [28] trial in June 2019. Among these images, around 8k images have been dropped in the dataset cleaning phase in which we removed those with bad quality or bad annotations.

Note that synthetic data was used for this work and they were not published or made available for public usage. No illustration in this document contains synthetic image information. Dataset from DRDC was also not released for public usage.

B. Image Tiling

Usually, before feeding images to convolutional neural networks, images are downsized to a certain resolution to get a faster result. However, drones can be really small (10-20 pixels) and the input image can be of a really high resolution.

[1] Landscape pictures: https://www.kaggle.com/arnaud58/landscape-pictures.
[2] For example, Amazon provides a license for personal or non-commercial use.

Therefore, downsizing the input image to a low resolution can have a detrimental effect on the performance of the network.

Instead of downsizing, we propose to keep the original resolution and split the image into multiple overlapping tiles, as shown in Figure 4. This strategy has shown great potential to detect small objects [29]. At inference, a batch is generated from the full resolution image. The predictions are then corrected with the location of each tile in the input image and are post-processed with Non-maximum Suppression.

Figure 4: Tiling strategy used at inference. We create a batch of test images by cropping the input image from multiple overlapping tiles organized in a grid-like structure. Predictions are then adjusted with the tile position in the input image and then post-processed with Non-maximum Suppression.

In training, we follow a slightly different protocol, as shown in Fig. 5. Instead of training on all image tiles, which mostly include background, we only sample 1 tile in every image. This tile is sampled at a random location in the image (not in a grid-like structure such as in inference). It is sampled 80% of the time at a location that includes a drone in order to show more positive examples and 20% of the time at a completely random location to see some background examples.

C. Data Augmentation

We used several augmentation methods such as affine transformations, horizontal flips, blurs, noise, contrast, brightness, colour perturbations and many others from the Albumentations package [30] (such as rain, as shown in Fig. 6a). We also used style augmentation [31] (Fig. 6b), which is a style transfer technique that can potentially help generalizing to different domains.

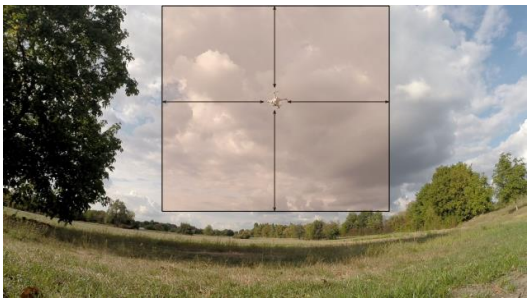
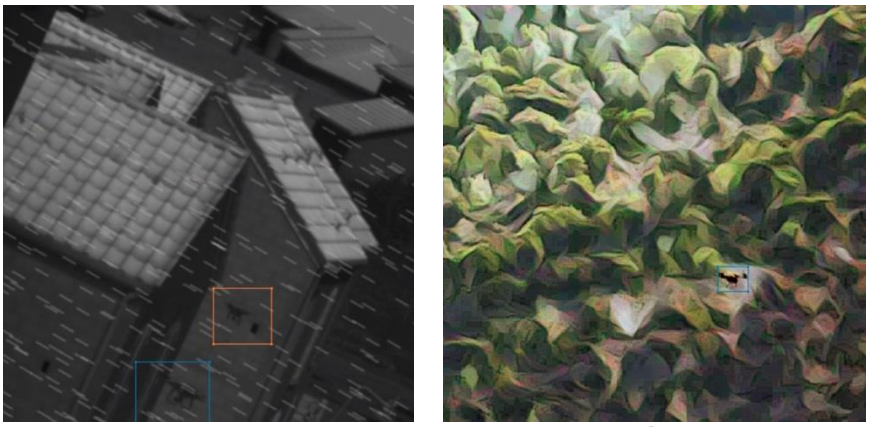


Figure 5: During training, the input image is randomly cropped. 80% of the time, it is cropped at a location that includes a drone. In this image, we show the area covered by all possible crop locations to include this drone.



(a) (b)

Figure 6: Examples of our data augmentation pipeline in (a) and of style augmentation [30] in (b).

V. EXPERIMENTS

In this section, we first assess the global performance of our approach on different data splits of the 2020 Drone-vs-Bird Challenge that we used in preparation of the challenge. We also evaluate the impact of different tile resolutions and data augmentation strategies. Then, we show the generalization performance between different datasets and finally, we report the official results of the 2020 Drone-vs-Bird Challenge.

A. Overall Performance

We evaluated our approach for two different data sampling strategies. The results, which are averaged for 5 different random data splits, are reported in Table 1. The first data sampling strategy that we tested is the standard way of splitting data: all

images were sampled into training (85% of the images), validation (5%) and test (10%) sets. We named this strategy "known sequences", since images were sampled without taking the video sequence of the Drone-vs-Bird dataset into account (meaning that images in the test set could come from the same sequence as images of the training set). For the second strategy, which we named "unknown sequences", we exclusively selected 12 sequences for the test set and the images from the remaining 64 sequences were selected for training and validation. This is a more realistic way of measuring the generalization performance of our approach by decreasing the odds of overfitting the video sequences. The Precision-Recall curves of the 5 different runs on unknown sequences are shown in Fig. 7. For experiments in this section, we used a tile size of 480 pixels and an overlap of 100 pixels between each tile.

Table 1: Results (mAP) of our approach when testing on known and unknown sequences sampled from the 2020 Drone-vs-Bird Challenge. We report the mean and standard deviation for 5 different data splits (1 run per data split).

Sequence	mAP (%)
Known	93.55 ± 0.6
Unknown	66.0 ± 13.5

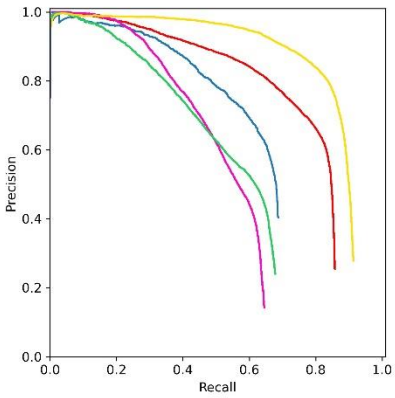


Figure 7: Precision-Recall curves of our approach for the 5 "Unknown sequences" data splits used to generate the results in Table 1.

B. Tile Resolution

We evaluated the impact of tile resolution during training and tests. We report the results of this experiment in Table 2. We performed the experiment using the same data split for all tile sizes. The left column represents the training tile resolution, the top row represents the test resolutions and the values correspond to the mAP

(%). For some reason (not investigated), increasing the tile resolution in our case reduces the detection performance.

Table 2: Impact of tile resolution on the mAP metric (%). The left column represents the training tile resolution and the top row represents the test resolutions.

Tile Size (px)	320	480	640
320	88.8	89.5	86.1
480	91.3	92.9	89.1
640	90.6	92.9	90.1

Table 3: Impact of test tile resolution (with fixed training resolution of 480 px) on detection performances and processing time on images from seen and unseen sequences.

Metrics / Tile Size (px)	320	480	640
Known sequence mAP (%)	91.3	92.9	89.1
Unknown sequence mAP (%)	56.1	57.8	44.3
Processing time (ms)	195	169	125

We also evaluated the impact of the test tile resolution (with a fixed training tile size of 480 pixels) on detection performances and processing times for both seen and unseen sequences. The results of this experiment are reported in Table 3.

From Tables 2 and 3, a training and testing tile size of 480 pixels seems to be a good choice both in terms of detection performances and processing time.

We also compared the performance of image tiling with image resizing in Table 4, for which the reported performances are on the same single known data split. We can see that performances when using higher resolution images are slightly superior to image tiling and the processing time is approximately the same. The result is counter intuitive since the tilling method maintains a higher resolution on the target and aims to get better identification. However, the discrepancy between the full resolution and the tilling is less than a percent and it can be concluded that both method gives similar performance. The biggest advantage of tiling is for image sequences, because image processing can be applied only on some sections which can greatly improve the processing time (up to a 10x gain). Additional details are given in section 6.

C. Data Augmentation

In this experiment, we evaluate the impact of synthetic data and style augmentation. We report the results on an “unknown” sequence in Table 5. We

can see from the results that adding both strategies helped our approach to generalize better to images of unseen sequences.

Table 4: Performance when resizing the biggest side of the image to the mentioned resolution instead of using image tiling

Test Resolution	mAP (%)
480	24.0
640	57.3
1024	73.3
2048	93.2
Full	93.5
480 (tiling)	92.9

Table 5: Impact of synthetic data and style augmentation evaluated on images from unknown sequences of the Drone-vs-Bird (DvB) dataset.

Training	mAP (%)
DvB	52.2
DvB + synthetic	55.6
DvB + synthetic + style augmentation	57.8

D. Generalization to other datasets

We evaluated the generalization performance of our approach by training on different datasets (described in section 4.1). We report in Table 6 the performance matrix when training on a certain dataset (left column) and testing on different datasets (top row). We can observe that generalizing to other datasets is a difficult task. By combining all of them together during training, we can however achieve good generalization.

Table 6: Generalization performance on different test sets (top row) when training with different datasets (left column).

Training images	Drone-vs-Bird	DRDC (colour)	DRDC (IR)
Synthetic	20.7	2.42	4.12
Drone-vs-Bird	92.3	22.1	29.7
DRDC (colour)	30.6	86.9	35.9
DRDC (IR)	24.7	31.7	91.5

All	91.9	76.0	86.1

E. Official 2020 Drone-vs-Bird Challenge Results

We participated in the 2020 Drone-vs-Bird Challenge. Only 3 teams (including us) submitted their results among the participants. Our first submission included the synthetic images we have generated (see section 4A) and the second submission included only the images provided in the challenge. We ranked second in the challenge, 0.2% behind team Gradiant [32] as shown in Table 7. Their best submission included real images from external sources, which may have helped them get a boost in performance. However, as shown in section 5D, better generalization does not necessarily mean that it performs better for a specific dataset (especially one for which many scenes have been seen during training). It is thus difficult to compare both approaches, since different data was used in training.

Table 7: Performance of the different teams that submitted their results.

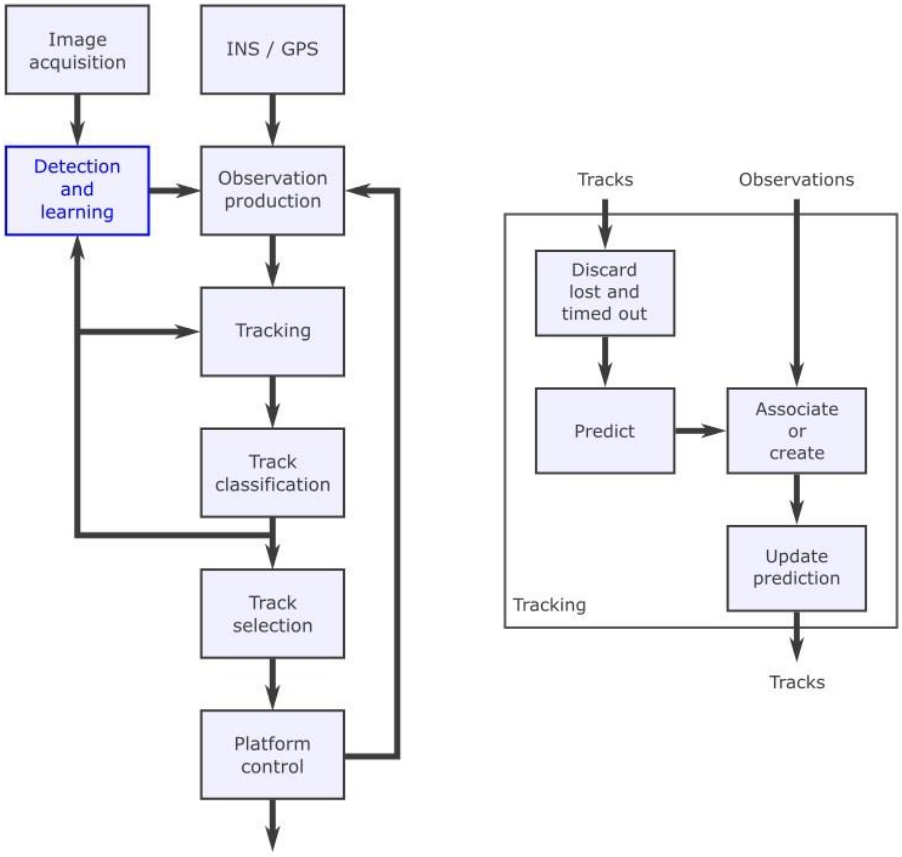
Team	mAP (%)
Gradiant - submission 1	80.0
Gradiant - submission 2	72.9
Gradiant - submission 3	75.3
Eagledrone - submission 1	66.8
ALEXIS (us) - submission 1	79.8
ALEXIS (us) - submission 2	79.4

VI. Global Approach With ALEXIS

So far, we have described our approach for drone detection and reported results on both colour and infrared images. In this section, we detail how we leverage our drone detection approach into a global one that can use multiple sensors at the same time and can track multiple drones in real time.

ALEXIS stands for Automated Light EXperimental Imagery System. It is mainly a data collection system and a real life test bench for counter UAV related algorithm development. Its hardware consist of two RGB visible cameras (narrow and wide field of views) and three infrared imagers (short-wave (SWIR), mid-wave (MWIR) and long-wave (LWIR)). All are mounted on a motorized pan & tilt turret installed on a tripod. The system's control software is DRDC's Versatile Tracking System (VTS). It allows adjusting the camera settings, acquiring and recording data and controlling the pan & tilt turret based on the tracking algorithms results. Each camera can be selected as the primary image source for live detection and tracking. The other cameras are still recorded and can be "hot swapped" as the primary

source if necessary. It is also possible to use the recorded data for later offline processing and analysis. The global tracking framework is at the core of the VTS and is presented in Fig. 8a. Each block is described in the following.



(a) Overall structure. (b) Tracking details.

Figure 8: Overview of the global tracking framework.

The system's inputs consist of an image stream and data streams from devices like turret encoders, an IMU (Inertial Measurement Unit) and a GPS (Global Positioning System). The information provided by the encoders and the IMU is useful to model the physical system's dynamics, while the GPS position is used to geo-localize the tracked targets. The targets distance can be measured by a range finder or estimated by triangulation knowing the objects size. The image stream is fed into the detection and learning module. This is where the presented neural network is put to contribution. It can be used in two ways according to the available computational resources: alone as the main runtime detector or to initialize a lighter weight pattern matcher (for example: MOSSE [33]). The learning process is performed off-line when the main runtime detector is the neural network, while it

is said on-line when a pattern matcher is used since it is updated at each iteration with the results of the previous one. Since, both detection methods can produce multiple results, the remaining modules are designed to handle it; hence the framework implements a multiple target tracker. The observation producer knows the camera characteristics and uses the information from the physical world to convert the image results into a higher level physical representation: the observations. The tracking module associates each observation with an existing track or a new track (more details are given later). A track is more or less a series of observations. The same neural network can be used to periodically classify the tracks to validate that we are still following an appropriate target when the network is not the runtime detector. Other techniques can also be used to classify tracks based on the time behaviour of their properties. For example, the variation of the bounding box area over the time could help to distinguish better between drones and birds. Indeed, one would expect a pulsation pattern for a flying bird's bounding box while a more constant (or slow changing) area is expected for a drone. Finally, a single track is selected amongst the probable drones according to a preference criterion (e.g. the nearest to the Field of View's (FoV) center, the more likely to leave the FoV, the most probable drone, etc...) to produce a command that will drive the turret. The targets GPS position can also be provided to other tracking systems and counter-measures.

Fig. 8b gives more details on the tracking block. It is based on two main concepts: measurements (the observations) and predictions. The tracker uses a model to produce a prediction for each active track. The model considers the camera's field of view, the pan & tilt speed and position and target features (e.g. bounding box, intensity and shape statistics). The observations know their production system, hence the predictor can access the physical world information via the observations. The predictions are used to associate each observation to an active track. Since the associator handles multiple tracks and observations, it is possible to perform an optimised assignment at this stage. When there is no possible association for an observation, a new track is created and added to the active tracks pool. When an active track cannot be updated with an observation, it is updated with its prediction. Generally, this happens in the presence of occlusion or when a target leaves the field of view. A track can be updated with its prediction only for a certain specified amount of time before being declared as lost and being removed from the active pool. Depending on the selected prediction technique, the predictions can be updated based on the association results.

The architecture combining a neural network and classical engineered algorithms allows for more flexibility than using a giant neural network taking an image stream as input and producing Pan and Tilt commands as outputs. Indeed, it allows to easily integrate the physical model of the observation system into the tracking process and thus make it possible to handle occlusions effectively. It also lightens a lot the network training requirements. It also makes it possible to add an

extra step of classification on the resulting tracks, either with a different neural network or with an engineered algorithm.

VII. Conclusion

In this paper, we proposed to use an image tiling strategy in combination with a detection approach based on YOLOv3 to detect small drones both in colour and infrared images. Our second place in the 2020 Drone-vs-Bird Challenge [16] shows that our results are quite competitive. However, when compared with our experiments on unknown sequences reported in Table 1, we can deduce that the sequences from the challenge were relatively easy. Many were in fact acquired in the same settings/backgrounds as the images in the training set, making it easier than expected for our approach. Also, our network may be biased towards finding any flying objects and the performance could potentially decrease drastically if many distractors (birds, planes, etc.) are present. Training for these additional classes would likely help to reduce this bias.

Interestingly, the synthetic images we have generated did not improve much the performance of our approach, especially for the challenge sequences. It can likely be explained by the domain gap, mainly caused by edge artefacts created by pasting cropped objects on real backgrounds. By generating images completely in simulation instead of relying on imperfect segmentation masks, Hinterstoisser et al. [21] nearly doubled the performance of their detection approach when training with synthetic objects rendered on synthetic backgrounds instead of using real backgrounds. It could therefore be an interesting direction for future work, as it would also allow us to test a rendering technique called Domain Randomization [22]–[23], which has shown great generalization performances for object detection.

Finally, we have developed a fully integrated system called ALEXIS which allows us to detect and track drones in real time and test algorithms in real context. At this point, the system has been evaluated qualitatively, but not quantitatively. Eventually, it would be interesting to compare the performance of the full system with the detection network.

References

[1] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[2] Grand View Research. Commercial drone market worth $129.2 billion by 2025 — cagr: 56.5%, June 2019.

[3] Grand View Research. Commercial drone market size, industry analysis report, 2019-2025, June 2019.

[4] G4S North America. White paper: Drones: Threat from above. Technical report, G4S North America, 2017.

[5] David Reid. A swarm of armed drones attacked a russian military base in syria. CNBC, Jan 2018.

[6] Róbert Szabolcsi. Beyond training minimums – a new concept of the uav operator training program, International conference knowledge-based organization. 22. 10.1515/kbo-2016-0096, 2016.
[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
[8] Stamatios Samaras, Eleni Diamantidou, Dimitrios Ataloglou, Nikos Sakellariou, Anastasios Vafeiadis, Vasilis Magoulianitis, Antonios Lalas, Anastasios Dimou, Dimitrios Zarpalas, Konstantinos Votis, et al. Deep learning on multi sensor data for counter uav applications—a systematic review. Sensors, 19(22):4837, 2019.
[9] Mrunalini Nalamati, Ankit Kapoor, Muhammed Saqib, Nabin Sharma, and Michael Blumenstein. Drone detection in long-range surveillance videos. In 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–6. IEEE, 2019.
[10] Arne Schumann, Lars Sommer, Johannes Klatte, Tobias Schuchert, and Jurgen Beyerer. Deep cross-domain flying object classification¨ for robust uav detection. In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–6. IEEE, 2017.
[11] Cemal Aker and Sinan Kalkan. Using deep networks for drone detection. In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–6. IEEE, 2017.
[12] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Detecting flying objects using a single moving camera. IEEE transactions on pattern analysis and machine intelligence, 39(5):879–892, 2016.
[13] Muhammad Saqib, Sultan Daud Khan, Nabin Sharma, and Michael Blumenstein. A study on detecting drones using deep convolutional neural networks. In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–5. IEEE, 2017.
[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.
[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.
[16] Angelo Coluccia, Alessio Fascista, Arne Schumann, Lars Sommer, Marian Ghenescu, Tomas Piatrik, Geert De Cubber, Mrunalini Nalamati, Ankit Kapoor, Muhammad Saqib, et al. Drone-vs-bird detection challenge at ieee avss2019. In 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–7. IEEE, 2019.
[17] Celine Craye and Salem Ardjoune. Spatio-temporal semantic segmentation for drone detection. In 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–5. IEEE, 2019.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
[19] Vasileios Magoulianitis, Dimitrios Ataloglou, Anastasios Dimou, Dimitrios Zarpalas, and Petros Daras. Does deep super-resolution enhance uav detection? In 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 1–6. IEEE, 2019.
[20] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 1301–1310, 2017.
[21] Stefan Hinterstoisser, Olivier Pauly, Hauke Heibel, Marek Martina, and Martin Bokeloh. An annotation saved is an annotation earned: Using fully synthetic training for object detection. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 0–0, 2019.
[22] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. In 2019 International Conference on Robotics and Automation (ICRA), pages 7249–7255. IEEE, 2019.
[23] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 969–977, 2018.
[24] Glenn Jocher, Yonghye Kwon, guigarfr, Josh Veitch-Michaelis, perry0418, Ttayu, Marc, Gabriel Bianconi, Fatih Baltacı, Daniel Suess, idow09, WannaSeaU, Wang Xinyu, Timothy M. Shead, Thomas Havlik, Piotr Skalski, NirZarrabi, LukeAI, LinCoce, Jeremy Hu, IlyaOvodov, GoogleWiki, Francisco Reveriano, Falak, and Dustin Kendall. Ultralytics yolov3, May 2020.
[25] Zhanchao Huang, Jianlin Wang, Xuesong Fu, Tao Yu, Yongqi Guo, and Rutong Wang. Dc-spp-yolo: dense connection and spatial pyramid pooling based yolo for object detection. Information Sciences, 2020.
[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence Zitnick.´ Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014.
[27] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. arXiv preprint arXiv:1911.08287, 2019.
[28] https://www.sto.nato.int/search/Pages/activities_results.aspx?k=set-260&s=Search%20Activities, NATO SET-260 web page activity, consulted 04/09/2019.
[29] F Ozge Unel, BO Ozkalayci, and C Cigla. The power of tiling for small object detection, 2019.

[30] Alexander Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I Iglovikov, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. arXiv preprint arXiv:1809.06839, 2018.

[31] Philip T Jackson, Amir Atapour-Abarghouei, Stephen Bonner, Toby P Breckon, and Boguslaw Obara. Style augmentation: Data augmentation via style randomization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 83–92, 2019.

[32] Gradiant. https://www.gradiant.org/en/.

[33] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 2544–2550, 2010.