Tracking and Classification of Aerial Objects

Marcia Baptista(✉), Luis Fernandes, and Paulo Chaves

INOV Inesc Inovacao, Lisbon, Portugal
{marcia.baptista,luis.fernandes,paulo.chaves}@inov.pt

Abstract. Unauthorized drone flying can prompt disruptions in critical facilities such as airports or railways. To prevent these situations, we propose a surveillance system that can sense malicious and/or illicit aerial targets. The idea is to track moving aerial objects using a static camera and when a tracked object is considered suspicious, the camera zooms in to take a snapshot of the target. This snapshot is then classified as an aircraft, drone, bird or cloud. In this work, we propose the classical technique of two-frame background subtraction to detect moving objects. We use the discrete Kalman filter to predict the location of each object and the Jonker-Volgenant algorithm to match objects between consecutive image frames. A deep residual network, trained with transfer learning, is used for image classification. The residual net ResNet-50 developed for the ILSVRC competition was retrained for this purpose. The performance of the system was evaluated with positive results in real-world conditions. The system was able to track multiple aerial objects with acceptable accuracy and the classification system also exhibited high performance.

Keywords: Object tracking · Deep learning · Residual networks

1 Introduction

Unmanned aerial vehicles (UAVs) come with numerous advantages. However, along with the positive aspects, drones present some undesirable characteristics, such as the possibility of a sudden crash, cyber attacks, and privacy issues, which could prevent the technology from developing at a faster pace in the short/mid-term. A major issue here is the safety of critical infrastructures such as airports, railways, and other transportation networks. As a response to the unauthorized and/or malicious use of drones, work has been done with the aim of protecting sensitive areas from the presence of drones [1]. One such technique consists in tracking drones within a given perimeter using a video surveillance system. Video surveillance systems work by generating alerts to the facility whenever the trajectory of a drone or other aerial object is considered suspicious. Companies such as the Nippon Electric Company [2] are already investing in the development of these systems. In this work, we propose to advance the state of the art in the field by evaluating, in real-world challenging conditions, a combination of automated vision algorithms and deep learning technologies that help detect the presence of intruding targets in prohibited airspace.

Our work is within the scope of the Advanced Low Flying Aircraft Detection and Tracking (ALFA) project, sponsored by Horizon2020, which builds on results from a

A. L. Martins et al. (Eds.): INTSYS 2019, LNICST 310, pp. 264–276, 2020.
https://doi.org/10.1007/978-3-030-38822-5_18

number of European Union (EU) sub-projects. The main goal of the project is the development of a system for real-time tracking, and classification of suspicious air targets. We are currently developing two modules in parallel: the tracking module and the classification module. The first module is responsible for object tracking. It collects images from the vision system and processes them using classical vision algorithms. The second module is responsible for object classification. Given a zoomed-in picture of an aerial target, it classifies the image as an aircraft, drone, bird or cloud. A deep learning pipeline is used for this purpose. The overall idea is that the tracking module should generate at each moment a list of items that are hypotheses for the presence of suspicious aerial objects. Hypotheses should be proven or disproven using the classification module. This should be done by zooming on the target, taking a snapshot and sending the image to the deep learning solution.

The remaining of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces methods. Section 4 describes the datasets and experimental settings and Sect. 5 presents and discusses the results. Section 6 concludes the paper.

2 Related Work

2.1 Object Tracking

Typically, an object tracker consists of two steps (1) recognizing objects from the background, and (2) following the trajectory of the detected objects [3]. The first step is usually accomplished with motion tracking methods. In the second step, the objects detected are linked into trajectories (or tracks). When an object is detected in the current frame, the model tries to associate the observed item with an existing trajectory. This task of associating objects with trajectories is typically cast as an optimization problem. Classical deterministic approaches to this problem include dynamic programming, bipartite graph matching, min-cost max-flow network flow and conditional random fields [3]. A popular method here is the Hungarian method [4], which is able to solve the bipartite graph matching assignment problem in polynomial time, with complexity $O(n^3)$ where n is the number of trajectories. Despite its popularity, it has been shown that the Jonker–Volgenant solver [5] can obtain similar results to the Hungarian method in less time, considering both average and maximum time [6, 7]. The solver is reported to be ten times faster than a similar coding of the Hungarian code [8]. Probabilistic methods such as Kalman filter [9] and Particle filter [10] can also be utilized in tracking. Here, the state of each object is represented as a distribution with uncertainty. It is also common to find works such as [11, 12], that combine the Hungarian algorithm with Kalman filter in order to obtain a more robust tracking framework.

2.2 Object Classification

Recently, there has been an increased interest in the classification of aerial targets using deep convolutional neural networks (CNNs). Some studies have used standard CNNs to address this problem [13–15]. The advantage of these networks, compared to other more complex CNNs, is the optimized use of computational resources. In Aker et al. [14], a CNN was shown to distinguish between drones and birds with precision and recall values above 90%. Unlu et al. [13] reached detection percentages of 93.7% and 64.6%, for birds and drones respectively, with a CNN. The project SafeShore [16] proposed a "drone-vs-bird detection challenge" where the goal was to detect drones in a video where birds could also be present. The winner of the competition, Schumann et al. [15], reported 99.2%, 99.1% and 98.9% correct identification percentages for UAV, birds, and clutter (background) using a CNN.

Other works have applied more advanced CNNs [17, 18, 19]. Advanced CNNs have typically the disadvantage of being harder to tune and require more training data. To accelerate the training and improve performance, some works [17] use pre-trained models and transfer learning to build the image classification models. In Saqib et al. [17], birds and drones were classified using ZFNet, VGG16, and VGG_M_1024 (all with Faster-RCNN). Transfer learning was used to help the system converge faster and to deal with the sparse dataset used. The authors reported the best mean absolute precision (mAP) of 0.66 with VGG16. The work in Liu et al. [18] used YOLOv2 to distinguish between airplanes, helicopters, and drones with classification accuracies of 96.03%, 90.47%, and 52.13%. The authors did not report using transfer learning but mentioned the use of a comprehensive dataset of about 30,000 images. The work of Park et al. [19] compared six convolutional models in their ability to distinguish between 11 drone models, namely YOLOv2, SSD with MobileNet, SSD with Inception V2, R-FCN with Resnet 101, Faster-RCNN with Resnet 101, and Faster-RCNN with Inception Resnet. The authors reported an F-measure of 74.3% for the best model (Faster-RCNN with Inception Resnet). The authors did not mention the use of transfer learning but refer a dataset of 9,525 labeled drone images.

Despite the positive results, the previous works are, however, not totally adequate for airport or railway surveillance, as none of them is all-encompassing of the classes of birds, drones, aircraft, and clouds. Perhaps the most suitable work would be that of Schumann et al. [15], which distinguished between drones, birds, and clutter. However, Schumann et al. did not train the classifier to distinguish drones from aircraft as we do. Nevertheless, and especially in airport surveillance, it can be important to differentiate between small aircraft and drones as countermeasures can be quite different. In addition, the convolutional neural network used is standard and less advanced than our proposal. In addition, the authors report on a large dataset, with 3386 drones, 3500 bird, and 3500 background images, but only 10% of these data were used as the test set.

2.3 Contributions

In our previous work [20], in the same line of research, we used a tracker based on the Hungarian algorithm. In this work, we instead use the faster method of Jonker–Volgenant. Previously, the residual network ResNet-50 was trained and tested using images from the Internet. In this work, the network is the same but we train/evaluate it with photos acquired in real-world conditions.

3 Solution

This section describes how the overall problem of tracking and classifying aerial objects from a video stream was addressed. The proposed architecture is represented in Fig. 1. As shown, the solution consists in using classical computer vision for object tracking and deep learning for object classification. The idea is that when the tracking detects a suspicious aerial object a snapshot is taken by the camera and a deep convolutional model performs object classification using the zoomed-in image. The following sections describe the tracking and classification modules in more detail.

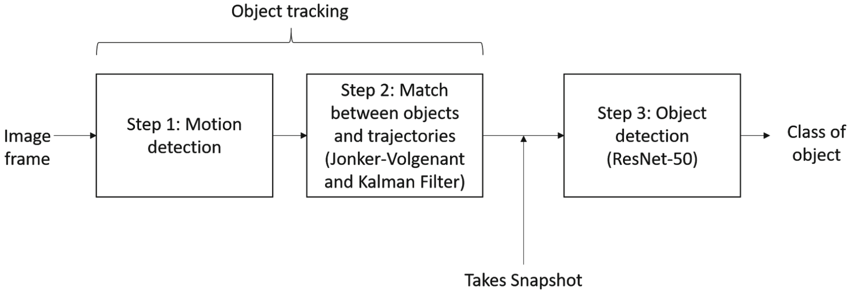

Object tracking
Image frame
Step 1: Motion detection
Step 2: Match between objects and trajectories (Jonker-Volgenant and Kalman Filter)
Step 3: Object detection (ResNet-50)
Class of object
Takes Snapshot

Fig. 1. The architecture proposed for tracking and classification of air targets.

3.1 Object Tracking

This section describes the methods used for locating and tracking one or more aerial objects in an input video. As shown in Fig. 1, the system receives at each moment a video frame and attempts to find relevant object observations (or detections). This is done by using a frame-difference motion detector that performs binary thresholding using a minimum and maximum threshold values ($\text{Thesh}_{\text{Min}}$ and $\text{Thresh}_{\text{Max}}$) [21]. The items detected are then dilated in order to optimize the probability of detecting well-defined targets. The dilating operation expands the found shapes, making them bigger according to a kernel (set to a 3x3 matrix) and a number of iterations (D_{I}). The outcome of this stage is a set of detections.

A multi-object tracking method based on the Jonker–Volgenant algorithm and Kalman filter is used to generate a set of reliable trajectories (or tracks) using previous information and detections in the current frame. The Kalman filter is used to help establish the tracking model, using the existing object information to predict future locations. At each moment, the filter estimates the object position and performs parameter correction. The Jonker–Volgenant algorithm is based on defining a cost matrix between tracks and detections and solving the nodes correspondence through a linear assignment method. The core of the Jonker–Volgenant algorithm is the shortest augmenting path traversal, as in the Hungarian solver, but it uses heuristics to reduce the execution time. The goal of the solver is to associate tracks with detections and also to start and remove tracks. A track is removed after a number of continuous frames are skipped (F_R) and an association is valid only if the Euclidean distance between the track and the observation is less than a certain threshold (D_V).

3.2 Object Classification

This section describes the methods used for classifying an aerial object after it is considered suspicious by the system. The deep learning model used here is a residual network. This is a state-of-the-art kind of convolutional neural network (CNN) that has achieved high classification performance on several datasets, such as ImageNet. The residual connections of residual networks make it possible to train deeper networks while reducing the probability of having overfitting problems. In addition, since these networks work by stacking modules of the same topology there is a reduced number of hyper-parameters. This simplicity also reduces the risk of overfitting. Transfer learning is used to train the residual network. Transfer learning has the advantage of reducing the training time of the neural network while resulting in a lower generalization error.

The residual network ResNet-50 was the network chosen to be retrained for our dataset. ResNet-50 is a network trained on a large set of images with 1000 categories. This training allows the network to detect generic features from images. Our re-training consisted of doing only small/simple weight adjustments in order to create the network for the intended classification. Prior to the re-training, it was necessary to remove the top layer of the ResNet-50, that considered the output of 1000 classes, and add a new layer with four outputs, one for each of the considered classes: aircraft, drone, bird, and clouds. The aircraft class contained both airplane and helicopter as well as military and civilian airplanes. The drone class included quadcopters, hexacopters, and octocopters. The output of the network was an array of classification probabilities.

4 Methodology

In this section, we present the methods and materials used to perform the evaluation of the system. We evaluate the system at two different stages: at the first stage we evaluate the tracking system and at the second stage we evaluate the deep learning pipeline for classification. This evaluation is done independently. In the following text, we describe the datasets used, the configuration of the tracking and deep learning solutions as well as the evaluation methods.

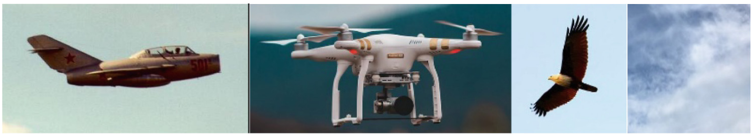
4.1 Datasets

From 24th to 28th of June 2019 a field experiment of the ALFA project took place in Cacela Velha, Portugal. The dates of the experiment were selected to guarantee good weather conditions. While different aerial targets were flying overhead (helicopter, light airplane, and drones), an off-the-shelf camera followed and recorded the moving objects. Some of these videos were collected and used in this study. Overall, they form dataset DS-1 and are further described in Table 1. In the scope of this paper, the goal of dataset DS-1 was to be used to evaluate the tracking capabilities of the system.

To evaluate the classification system we had three datasets: DS-2, DS-3, and DS-4. A total of 7763 images was collected from the Internet in order to be used to train the ResNet-50. This dataset, DS-2, consisted of images of aircraft, drones, birds, and clouds, where 2452, 2491, 2545 and 2758 were the number of images for each class, respectively. Some image examples are shown in Fig. 2. The aircraft class contained helicopters as well as military and civilian airplanes. The drones class included quadcopters, hexacopters, and octocopters. The images were cropped with the aim of having flying objects or birds against the sky. The images were resized to the proper input dimensions of the used neural networks. The number of images was augmented several times by generating new images from the original ones, which was done applying rotation, shift, shear, zoom, and flip.

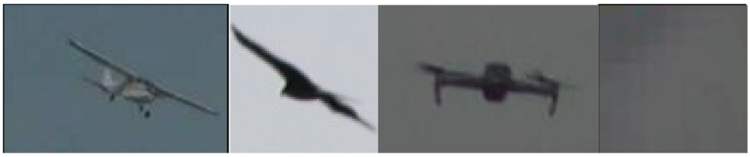
Table 1. Description of videos in dataset DS-1 (videos to evaluate tracking).

Video	Description	Frames	Frames with target(s)
1	Video of a light airplane (Cessna 172)	4187	3069 (73.30%)
2	Video of a helicopter (Eurocopter AS355F1)	3225	3045 (94.42%)
3	Video of a quadcopter	5630	5530 (98.22%)



a) aircraft b) UAV (drone) c) bird d) clouds

Fig. 2. Examples of images collected from the internet (DS-2) to train the classification system.



a) aircraft b) bird c) UAV (drone) d) clouds

Fig. 3. Examples of images from dataset DS-4 used to evaluate the classification system

The images collected from the Internet had high quality (see Fig. 2). However, our previous tests indicated that zoomed-in images of aerial objects from the field do not have this same quality (please compare the images in Fig. 2 with Fig. 3). Accordingly, and in order to to make our ResNet-50 more capable of handling images collected from the field, a new dataset, DS-3, also based on the same set of internet images, was created. Here, besides rotation, shear, zoom, and flipping, the object shift was increased, the colors were randomly changed and Gaussian blur was added.

We did several field experiments to collect zoomed-in pictures of aerial objects. The purpose here was to test ResNet-50 in real-world conditions. The experiments took place in several different places of Portugal (Leiria, Nazaré) and Holland (Monster). This dataset, DS-4, was composed of 582 aircraft, 39 birds, 128 clouds and 1091 drone images. To obtain the testing images, the camera zoomed in on different objects and took a snapshot for posterior processing. The small number of bird images was increased by using 876 images from a work [22] where images were gathered in a wind farm and had relatively low quality due to capturing distant birds. Only images of birds with more than 40x40 pixels were used. This resulted in a total of 915 images with birds. Some examples of images from this dataset are shown in Fig. 3.

4.2 Configuration

The object tracking system was configured for each video in the dataset. Generally, in the implementation of the Kalman filter, we have set the process noise (Q) high (set to 10), compared to the measurement noise (R) which was set to 0.001. This allowed us to adapt more effectively to the sudden changes in the speed of the aerial objects. We also had to set the maximum Euclidean distance traveled (D_V = 75, 200 and 300) to a large value because the objects sometimes moved fast and traveled great distances from one frame to the other. By setting the number of dilating iterations (D_I) to 15, 30, and 40 we were able to track both close and distant aerial objects. To capture significative changes but disregard minor alterations we set the parameter $\text{Thresh}_{\text{MIN}}$ of binary thresholding to 30. The parameter $\text{Thresh}_{\text{MAX}}$ was set to 255. The maximum number of frames that a track could be idle (F_S) was set to 20 frames.

Our work comprised the creation of a ResNet-50 with four output neurons, one for each considered class. The activation function was Rectified linear unit (ReLU). The ResNet-50 had an input convolutional layer and max pooling, followed by 48 residual modules. In the end, there was a fully connected network. ResNet was trained with transfer learning. The software was implemented in KERAS (https://keras.io/) to run in graphical processing units (GPU). The training algorithm used was Stochastic gradient descent (SGD), the training error was measured by Categorical cross-entropy.

4.3 Evaluation Method

To perform an evaluation of the tracking system, we developed an automatic tool to help annotate video. All videos in dataset DS-1 were annotated with this tool by manually placing bounding boxes around aerial objects and interpolating their trajectories between keyframes. All objects were annotated, except in case of total occlusion. Each object of interest entering the scene got a unique ID, i.e. if a target left the screen to reappear later again, a new identifier would be assigned. Please note that bounding boxes were fairly aligned but not always perfectly aligned due to incorrect interpolation or mistakes made by the annotator.

To evaluate the tracking system we used the metrics referred in [23]. We chose this work because the authors define tracking performance in terms of tracks. The basis of the evaluation is the Intersection over Union (IoU) metric:

$$IoU_k = \frac{Area_{overlap}}{Area_{Union}} \tag{1}$$

Intersection over Union (IoU) for a given object in frame k is a ratio where the numerator is the area of overlap between the predicted bounding box and the ground-truth bounding box and the denominator is the area of union. We define the following binary variable based on a threshold Th_{IoU} which in our examples is set to 20%, as in [23]:

$$O_k = \begin{cases} 1, & IoU_k \geq Th_{IoU} \\ 0, & cc \end{cases} \tag{2}$$

The concept of IoU allows classifying tracks as true positive (TP), false positive (FP) or false negative (FN). Concretely, a ground truth track GT with N number of frames has been correctly detected if there exists at least one track T where:

$$\frac{\sum_{k=1}^{N} IoU_k(GT,T)O_k(GT,T)}{N} \geq Th_{spatial} \tag{3}$$

$$\frac{\sum_{k=1}^{N} O_k(GT,T)}{N} \geq Th_{temporal} \tag{4}$$

The previous conditions mean that coverage (in number of frames) should be larger than a predefined overlap threshold which we set to 15% ($Th_{temporal}$), as in [23]. We also impose that the system track has sufficient spatial overlap ($Th_{spatial}$) with the ground truth track, that is set to 20%, as in [23]. A ground truth track is considered to have not been detected correctly whenever conditions (3) or (4) do not hold for all system tracks. We also measure the number of ground truth track fragmentations (TF) as the number of system tracks that fulfill conditions (3) and (4) for a given ground truth track. For each of these system tracks, we calculate closeness of track (CT) for a given track and a ground truth track GT as the ratio of the sum of IoU over the number of frames where there is a temporal overlap. In a similar way, we compute the track matching error (TME) as the average distance error between a system track and a ground truth track GT. Distance is measured as the Euclidean distance between the centroids of the two tracks. Finally, we use the metric of track completeness (TC) and average track completeness as the:

$$TC = \frac{\sum_{k=1}^{N} O_k(GT,T)}{N} \tag{5}$$

$$TCM = \frac{\sum_{GT=1}^{N_{GT}} \max(TC(GT,T))}{N_{GT}} \tag{6}$$

In order to evaluate the classification capabilities of ResNet-50, three datasets were used for training (DS-2 and DS-3) and three datasets (DS-2, DS-3, and DS-4) were used for testing. Concretely, the dataset DS-2 was split into three sets for training, validation, and testing. The training set was used to adjust the network weights, while the validation set was used to select the best hyperparameters. The network performance was evaluated on the testing set in order to serve as a baseline. The split used for each class was 1000 images for training, 500 for validation and the remaining for the testing which resulted in a total of 4000, 2000 and 4246 images for training, validation, and testing, respectively. A similar procedure was used for dataset DS-3. Even though the testing sets of DS-2 and DS-3 were used for evaluating the models created, we also run tests using the data of DS-4, which consists of 582, 915, 128, and 1091 images of aircraft, birds, clouds, and drones, respectively. Here, we were interested in investigating how the models worked under real-world conditions.

To evaluate the ResNet-50 models we use the metric of recall, i.e. the number of items correctly identified as positive out of the total actual positives for a given class—TP/(TP+FN). We also compute the metric of precision, i.e. the number of items correctly identified as positive out of all instances where the algorithm declared the class—TP/(TP+FP). We analyze recall/precision for each class of interest. The macro-average F-Score is considered, i.e. the harmonic mean of the average recall and average precision.

5 Results

This section presents the results of evaluating the tracking system and evaluating the system classification methods. The goal here was to show that we can attain reasonable performance in a real-world scenario.

Table 2. Evaluation results for DS-1.

Video	Video 1	Video 2	Video 3
Correctly detected tracks (TP) (%)	83.33%	83.33%	100%
Incorrectly detected tracks (FN) (%)	16.67%	16.67%	0%
Average track fragmentations (*TF*)	1.17 ± 0.4	1.20 ± 0.4	1
Average of track closeness (*CT*) (%)	54.02 ± 18.8	54.03 ± 20.2	59.82 ± 16.1
Average track matching error (*TME*)	6.66 ± 3.7	18.44 ± 14.3	27.70 ± 9.6
Average track completeness (*TC*) (%)	66.63 ± 19.3	61.44 ± 30.0	55.23 ± 27.8

5.1 Tracking

We investigated the performance of the tracking system in each of the three videos of dataset DS-1. As shown in Table 2, the percentage of correctly detected tracks was consistently high. Almost all tracks were covered both spatially and temporally. In regards to tracking completeness, our results were also positive, showing that the ground truth tracks had considerable temporal overlap with their longest corresponding system tracks. Concretely, ground truth tracks were able to be covered by their longest system tracks by a considerable percentage – up to 55%. In regards to tracking closeness, the results were positive.

The track closeness was high, above 54%, meaning that the spatial coverage was on average reasonably good. It would be difficult to reach larger values in this respect. This can be explained by the fact that the bounding boxes annotated are usually larger than the bounding boxes detected, which results in low Intersection over Union (IoU) values. Moreover, the results in terms of track closeness suggest that detection by frame subtraction is a method not always robust to slow object movements or very rapid object movements. In the case of slow movement, only a small part of the object may be detected. In the case of rapid movement, the object detected may encompass the location of the object in the previous frame and in the current frame. More sophisticated object detection techniques could be used to improve the metrics of track closeness.

The metric of track matching error showed that the predicted trajectories and the real trajectories were consistently close. Please consider that the average Euclidean distances correspond to images with 1280 × 720 pixels. Accordingly, the TMEs obtained are considerably low, a promising result.

5.2 Classification

The results of evaluating the classification system are presented in Table 3. We were first interested in investigating the performance of the ResNet-50 trained with the original images from the Internet (dataset DS-2). When testing this ResNet-50 using the dataset of the images that the authors collected from the field (dataset DS-4), the F-1 score decreased significantly, from 98.7% to 73.5% compared to the same ResNet-50 tested on the DS-2 testing set. This was due to a decrease in both precision and recall. Concretely, the drone class decreased from a recall of 98.7%, with internet obtained images, down to 70.0%, with images from the field. The performance reduction was even more significative for the aircraft class, from 98.1% down to 45.5%. The bird class had the lowest F1-score of all the classes due to its very low precision (38.0%). This overall performance decrease was probably due to the lower quality of the images in the new test set (from dataset DS-4), namely the sky color not being from a vibrant blue and due to some blur that originated less defined shapes.

Table 3. Recall (R) (%), Precision (P) (%) and F-1 Score for different ResNet-50.

Net	Input Size	Data		Classes								F-1 Score
		Train Dataset	Testing Dataset	Aircraft		Birds		Clouds		Drones		
				R	P	R	P	R	P	R	P	
1	101 × 101	DS-2	DS-2	98.1	97.4	98.6	98.8	100	100	98.2	98.7	98.7
2	71 × 71	DS-3	DS-3	94.9	95.6	96.7	99.0	99.7	100	97.8	94.3	97.2
3	101 × 101	DS-2	DS-4	45.5	70.9	95.7	71.0	100	38.0	70.0	99.0	73.5
4	71 × 71	DS-3	DS-4	83.0	84.4	88.3	97.3	100	97.0	99.5	91.8	92.7

In order to try to make the ResNet-50 more capable of handling lower quality images, a new training set (from dataset DS-3), which consisted of internet images that were subject to blur, color change and object shift in addition to the operations of rotation, shear, zoom, and flipping, was used to make a new model. This new ResNet-50, had a high F1-score when tested on the same dataset. The F-1 score of 97.2% is comparable to the score of 98.8% obtained by the ResNet-50 trained and tested with only high-quality images from the internet (dataset DS-2). The small difference that we found is probably a consequence of the more demanding training set and also of decreasing the image sizes from 101 × 101 pixels to 71 × 71 pixels in order to limit the amount of memory necessary to create and use the datasets.

When the new ResNet-50, trained with data from dataset DS-3, was applied to the test set of images collected from the field, dataset DS-4, the F1-Score increased to 92.7%. This result suggests that this model is suitable to be used in real-world conditions. The model's recall of 83% for aircraft and 99.5% for drones is a significant improvement with respect to the previous figures of 45.5% and 70.0%, indicating that, as expected, training with different colors and blur brings robustness to classification.

6 Conclusions

The widespread use of amateur drones and other aircraft poses various safety, security and privacy threats. To address these challenges, drone surveillance is an important but not totally explored topic. In this paper, we were interested in evaluating in real-world conditions a tracking and classification system that targets drones, birds and other aircraft. This kind of methods can be integrated into surveillance systems used in airports or can be used to secure other intelligent transportation systems, such as railways or the metro network.

This paper comes from a line of work [20] in which we used a tracker based on the Hungarian algorithm and trained/evaluated a ResNet-50 for classification with images from the Internet. In this work, we use the Jonker–Volgenant for tracking and train the same network with photos acquired in real-world conditions. Our results are positive showing that we can attain reasonable performance in tracking and classifying multiple aerial targets.

As future work, we intend to improve the detection methods of close objects as we found out that by using movement subtraction to detect aerial objects we were not always able to fully detect the true boundaries of the object in the frame. Rapid and slow movements made the detection bounding box encompass both the previous and the next location of the object in the frame.

Acknowledgment. The "ALFA - Advanced Low Flying Aircrafts Detection and Tracking" project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 700002.

References

1. Altawy, R., Youssef, A.M.: Security, privacy, and safety aspects of civilian drones: a survey. ACM Trans. Cyber-Phys. Syst. **1**(2), 1–25 (2016)

2. “NEC’s surveillance system will detect, track drones,” PCWorld, 08-Oct-2015. https://www.pcworld.com/article/2990525/necs-surveillance-system-will-detect-track-drones.html. Accessed 03 Aug 2019
3. Luo, W., et al.: Multiple Object Tracking: A Literature Review, ArXiv14097618 Cs, Sep. 2014
4. Kuhn, H.W.: The Hungarian method for the assignment problem. Nav. Res. Logist. Q. **2**(1–2), 83–97 (1955)
5. Jonker, R., Volgenant, A.: A shortest augmenting path algorithm for dense and sparse linear assignment problems. Computing **38**(4), 325–340 (1987)
6. Serratosa, F.: Speeding up fast bipartite graph matching through a new cost matrix. Int. J. Pattern Recogn. Artif. Intell. **29**(02), 1550010 (2015)
7. Levedahl, M.: Performance comparison of 2D assignment algorithms for assigning truth objects to measured tracks. In: Signal and Data Processing of Small Targets 2000, vol. 4048, pp. 380–389 (2000)
8. Cao, Y.: LAPJV—Jonker-Volgenant algorithm for linear assignment problem, v3. 0. Mathworks File Exch. vol. 26836 (2013). http://www.mathworks.com/matlabcentral/fileexchange
9. Reid, D.: An algorithm for tracking multiple targets. IEEE Trans. Autom. Control **24**(6), 843–854 (1979)
10. Khan, Z., Balch, T., Dellaert, F.: An MCMC-based particle filter for tracking multiple interacting targets. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. LNCS, vol. 3024, pp. 279–290. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24673-2_23
11. Luetteke, F., Zhang, X., Franke, J.: Implementation of the Hungarian Method for object tracking on a camera monitored transportation system. In: ROBOTIK 2012, 7th German Conference on Robotics, pp. 1–6 (2012)
12. Sahbani, B., Adiprawita, W.: Kalman filter and iterative-Hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system. In: 2016 6th International Conference on System Engineering and Technology (ICSET), pp. 109–115 (2016)
13. Unlu, E., Zenou, E., Riviere, N.: Using shape descriptors for UAV detection. Electron. Imaging. **2018**(9), 128-1–128-5 (2018)
14. Aker, C., Kalkan, S.: Using Deep Networks for Drone Detection, *ArXiv170605726 Cs*, Jun. 2017
15. Schumann, A., Sommer, L., Klatte, J., Schuchert, T., Beyerer, J.: Deep cross-domain flying object classification for robust UAV detection. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, pp. 1–6 (2017)
16. Coluccia, A. et al.: Drone-vs-bird detection challenge at IEEE AVSS2017. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, pp. 1–6 (2017)
17. Saqib, M., Daud Khan, S., Sharma, N., Blumenstein, M.: A study on detecting drones using deep convolutional neural networks. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, pp. 1–5 (2017)
18. Liu, H., Qu, F., Liu, Y., Zhao, W., Chen, Y.: A drone detection with aircraft classification based on a camera array. In: IOP Conference Series: Materials Science and Engineering, vol. 322, p. 052005, March 2018
19. Park, J., Kim, D.H., Shin, Y.S., Lee, S.: A comparison of convolutional object detectors for real-time drone tracking using a PTZ camera. In: presented at the 2017 17th International Conference on Control, Automation and Systems (ICCAS), pp. 696–699 (2017)

20. Fernandes, A., Baptista, M., Fernandes, L., Chaves, P.: Drone, aircraft and bird identification in video images using object tracking and residual neural networks. In: presented at the Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania (2019)
21. Singla, N.: Motion detection based on frame difference method. Int. J. Inf. Comput. Technol. **4**(15), 1559–1565 (2014)
22. Image Dataset for Bird Detection. http://bird.nae-lab.org/dataset/. Accessed 04 Aug 2019
23. Yin, F., Makris, D., Velastin, S.A.: Performance evaluation of object tracking algorithms. In: presented at the IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Rio De Janeiro, Brazil, p. 25 (2007)