# Drone, Aircraft and Bird Identification in Video Images Using Object Tracking and Residual Neural Networks International Conference ECAI 2019 https://ieeexplore.ieee.org/document/9...

**Preprint** · April 2020

CITATIONS
0

READS
31

**4 authors**, including:

Armando Fernandes
INOV - Inesc Inovação
**48** PUBLICATIONS **532** CITATIONS
SEE PROFILE

Márcia Baptista
Inesc-ID
**28** PUBLICATIONS **124** CITATIONS
SEE PROFILE

**ECAI 2019 - International Conference – 11th Edition**
Electronics, Computers and Artificial Intelligence
27 June -29 June, 2019, Pitesti, ROMÂNIA



https://ieeexplore.ieee.org/document/9042167

DOI: 10.1109/ECAI46879.2019.9042167

**ECAI 2019 - International Conference – 11th Edition**
Electronics, Computers and Artificial Intelligence
27 June -29 June, 2019, Pitesti, ROMÂNIA

# Drone, Aircraft and Bird Identification in Video Images Using Object Tracking and Residual Neural Networks

Armando Fernandes[a,b], Márcia Baptista[a], Luis Fernandes[a], Paulo Chaves[a]
[a]INOV INESC Inovação
[b]INESC-ID Instituto de Engenharia de Sistemas e Computadores – Investigação e Desenvolvimento
Rua Alves Redol, 9
1000-029 Lisbon, Portugal
armando.fernandes@inov.pt, marcia.baptista@inov.pt, luis.fernandes@inov.pt, paulo.chaves@inov.pt

***Abstract* – As maritime smuggling is being combatted more effectively, the criminal "modus operandi" consists more frequently of using small aircraft and drones for drug transport. To address this issue, we report our efforts to develop a system capable of accurately tracking suspicious flying objects and identifying them on video streams. Our solution consists in coupling classical computer vision with deep learning to perform tracking and object detection. A discrete Kalman filter is used to predict the location of each object being tracked while the Hungarian algorithm is used to match objects between successive frames. Whenever a potential target is considered suspicious the input images are zoomed and fed into a deep learning pipeline that separates images into the classes aircraft, drones, birds or clouds. A literature survey indicates that this problem with important applications is yet to be fully explored.**

***Keywords* – Object Tracking and Detection; Deep learning; Convolutional Neural Networks; Residual Networks**

## I. INTRODUCTION

The rising misuse of LSS (Low, Small and Slow) manned and unmanned aircraft and drones for illegal operations as, drug transport, is an emerging problem. These can be launched from almost any location and move at low altitude and speed to mask their presence. Drones can even autonomously reach any landing site even under adverse environmental conditions [1]. The work reported in this paper was conducted within the framework of the European project "ALFA - Advanced Low Flying Aircraft Detection and Tracking" whose main objective is to develop a system for timely detection, classification and understanding of the intentions of suspect air targets. The project combines radar technology with video detection methods, with the radar providing a direction to look at to an off-the-shelf camera. Since it is possible to use the video system without the radar, the interaction with the radar will not be studied here.

The developed system contains tracking methods that maintain a record of potential air targets. Whenever the tracking methods find suspicious activity, the camera zooms in on the potential target and the resulting image is fed into a deep learning pipeline that separates images into the classes of aircraft, drones, birds or clouds. The explanation for the use of these classes comes from the fact that the robust separation of birds and clouds is essential to avoid false alarms. Birds and clouds cannot be easily eliminated by the tracking methods, as other moving targets are, because their movement can be similar to that of aircraft. The classifier is also able to separate aircrafts from drones (of quadcopter type) since this distinction may result in different countermeasures.

To the best of the authors' knowledge, this is the first study applying an advanced type of convolutional neural network, a residual neural network, to the current problem, as well as a combination of the Hungarian algorithm and Kalman filter. Residual neural networks won the ILSVRC 2015 [2] competition and are appealing because their performance, contrarily to standard convolutional neural networks, does not degrade when depth increases significantly, allowing deeper networks and increased detection efficiency.

The remaining of this paper is organized as follows. Section II reviews related work. Section III introduces and explains the used methods. Section IV describes the datasets and the experimental setting and Section V presents the results. Finally, Section VI concludes the paper.

## II. RELATED WORK

### A. Tracking

Object tracking consists in locating objects throughout their moving stages in a sequence of images [3]. Object tracking is, in general, a challenging problem due to the projection of 3D objects on a 2D image, noisy and/or cluttered background, potential occlusions, lighting issues as well as real-time processing requirements [4]. The tracking of LSS objects poses even more significant problems [5]. With LSS objects, tracking has to deal with missed detections as well as false alarms, and also with the difficulties of appearance similarity among multiple objects.

The majority of existing work on visual tracking employs motion tracking methods [6]. Here, objects are first detected and then linked into trajectories. The task of associating objects with trajectories is typically cast as an optimization problem. Given an observation

(detection) the model attempts to associate the observed object with a trajectory. Some classical and deterministic approaches to this problem include bipartite graph matching, dynamic programming, min-cost max-flow network flow and conditional random field [6]. A method that stands out as a popular deterministic approach is the Hungarian algorithm [7], which is able to solve the bipartite graph matching assignment problem in polynomial time, with time complexity $O(n^3)$ where n is the number of tracks. In contrast with deterministic approaches, probabilistic tracking represents states of objects as a distribution with uncertainty. This more intuitive approach to problem usually relies on filtering techniques such as the Kalman filter [8], or particle filter [9]. It is common to find works combining the Hungarian algorithm with Kalman filter in order to obtain a more comprehensive solution. Examples of works with this combination are [10] and [11]. Frequently, this solution is used with YOLO for people tracking[12]. Curiously, the authors did not find scientific works applying this combination to aircraft and drone tracking.

A major issue in tracking systems is how to measure similarity between objects in frames. Different authors approach this issue differently [6], [7]. For LSS objects the extraction of an appearance model is a challenge as information such as color, shape and motion may be inexistent or limited due to the targets' small and/or varying size.

### *B. Identification*

The first works in object detection involved handcrafted features which required skill and a reasonable amount of time to complete the task. The promising results of deep learning in other fields and the fact that convolutional neural networks can learn their input features encouraged the community to apply them to the current problem. To the best of the authors' knowledge, this is the first study applying an advanced type of convolutional neural network, a residual neural network to the current problem. Some studies in flying object detection have used standard convolutional neural networks [13]–[18] and also VGG and ZFNet [19], that are elaborate convolutional neural networks previous to residual networks. Faster R-CNN [19]–[21] as well as YOLOv2 from Liu *et al.* [22], which are networks able to find a bounding-box for the objects in the images, were also employed. None of these works employs the shortcut connections responsible for residual networks success. These connections are described in detail in the methods section.

From all the studies in flying object detection only a few were found that employ some subset of the classes from the present study, namely birds, drones, aircraft or clouds. These studies will be analysed below in further detail. In fact, no study that employs these four classes simultaneously was found.

In Saqib *et al.* [19], birds and drones were separated using Faster RCNN, ZFNet and VGG. Since the objective was to draw a bounding-box around interesting objects they report a best mean absolute precision (mAP) of 0.66. Even though a 2727 frames dataset was mentioned it is unclear if this was a test set or if all frames contained drones. Another work, Farhadi *et al.* [21], employed a moving object detection system before a Faster-RCNN combined with the VGG model to identify birds and drones. The dataset contained 2130 frames with drones, but the test set dimension is again not clear. They report being able to follow the drone trajectory. In Aker *et al.* [23] a convolutional neural network without an independent object detection method associated was shown to distinguish between drones and birds with precision and recall values surpassing 90%. The work used 89 drone and 126 bird real images to generate a dataset. It is not clear if the test images are real or generated.

The work in Liu *et al.* [22] distinguished between airplanes, helicopters and drones with classification accuracies of 96.03%, 90.47% for the first two classes, but only 52.13% for drones, with YOLOv2. Even though the total number of images available for this work was 30000, the test set had only 300 drone, 100 helicopter and 100 fixed wing aircraft images which is smaller than the approximately 1000 images per class in the test sets of the present work. The images were partly collected from the internet and partly acquired specifically for the study.

Unlu *et al.* [13] used a test set with only 221 bird and 82 drone images and reached correct detection percentages of 93.7% and only 64.6%, respectively, with a convolutional neural network. However, for other algorithms, they reported better results that seem less reliable since the test sets contained less than 50 images in total. The data was obtained from open sources.

The project SafeShore [24] issued a "drone-vs-bird detection challenge" whose goal was to detect a drone appearing at some point in a video where birds could also be present. The winner of the competition Schumann *et al.* [18] reported 99.2%, 99.1% and 98.9% correct identification percentages for UAV, birds, and clutter (background), respectively. Even though they have gathered a large dataset with 3386 drone, 3500 bird and 3500 background images, only 10% was used as a test set. The system included an object detection method before the convolutional neural network, as we do, but different from ours. The images were from the internet and acquired specifically for the work. Even though it is rather similar to the present work, Schumann *et al.* did not train a classifier to distinguish drones from larger aircraft as we do. In addition, their convolutional neural network is standard and less advanced than our residual neural network.

## III. METHODS

This section describes how the problem of locating and identifying flying objects on a video stream was addressed. The overall architecture is represented in Figure III-1. As shown in the figure, the proposed architecture consists of a classical computer vision solution used for object tracking followed by a deep learning convolutional model that performs object identification. The next subsections describe each of these building blocks.

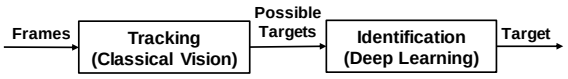


Figure III-1: Architecture proposed for tracking and identification of flying objects.

### A. Tracking

This section describes how we addressed the problem of locating multiple small moving objects in a sequence of images, maintaining their unique identities and capturing their individual trajectories given an input video. As shown in Figure III-1, the system receives at each moment an input image corresponding to a video frame and attempts to find relevant detections. We work with a resolution of 1280×720 pixels. This represents a considerable amount of information to process. It is however, not possible to downsample these images, at the risk of losing important information such as the location of a relevant object. Please note that we aim to detect small targets with a minimum size of three pixels in linear resolution. A trade-off between sensibility and false positive rate is necessary when detecting distant objects in a short period of time. Due to the previous restrictions, it is not possible to do motion detection [25] with the typical preprocessing technique of background subtraction. A more suitable alternative is the consecutive frame subtraction method [26]. In this method, for a given time instant t, the background is assumed to be the frame at time t-1. A difference image $D(x, y, t)$ is calculated as follows:

$$D(x, y, t) = |F(x, y, t) - F(x, y, t-1)| \qquad (1)$$

The accuracy of a frame-difference motion detector is somewhat limited. Due to its design, missed detections and false alarms can occur frequently which can provide misleading information to the tracking algorithm. Accordingly, it is necessary to have a tracking method robust enough to overcome some of the possible failures of the motion detector and the difficulties caused by the similar appearance of multiple small objects. To address the previous issues, an online multi-object tracking method based on the Hungarian algorithm and Kalman filter was used. The goal was to generate a set of reliable object tracks using previous information and current detections.

The Hungarian algorithm assumes the existence of two groups of data. The first is the current data and the second is the previous data. Current data represents each detected moving object in the current video frame where this data has no identity yet, and the previous data represents the data that have a certain identity set by the previous data association process (i.e. the tracks). The cost distance of each pair detection/track is computed in a cost assignment matrix. The algorithm works so that every track is assigned to a detection, and every detection is assigned to a track as to minimize the total cost of assigning all the tracks to detections. The aim is to associate tracks with detections as well as initialize tracks and terminate them.

In this work the focus was on the shape and motion features required to calculate the affinity between two objects. Having to track small targets with limited or no appearance information made it impossible to rely on appearance. Consequently, the following two models were explored [27]:

#### 1) Shape Model

The aspect ratio of the tracked object is expected to change over time and, as a result, the average height/width of the object over the track cannot be used. Instead, the last measurements of the object in the track are used as its shape affinity:

$$aff_{shape}(d_i, t_j) = e^{-w_1\left(\frac{|h_t - h_d|}{h_t + h_d} + \frac{|w_t - w_d|}{w_t + w_d}\right)} \qquad (2)$$

where $d$ and $t$ represent a detection and a track object while $w$ and $h$ represent the width and height of the detection's bounding box and the factor $w_1$ weights the importance of the shape factor in the final cost/affinity calculation;

#### 2) Motion Model

The current position $(x, y)$ is used as a motion model:

$$aff_{motion}(d_i, t_j) = e^{-w_2\left(\left(\frac{|x_t - x_d|}{w_d}\right)^2 + \left(\frac{|y_t - y_d|}{h_d}\right)^2\right)} \qquad (3)$$

where the factor $w_2$ weights the importance of the motion factor in the final cost/affinity calculation.

Assuming the independence between the two previous models, the affinity between detections/tracks can be calculated as:

$$a_{ij} = aff_{shape}(d_i, t_j) * aff_{motion}(d_i, t_j) \qquad (4)$$

In tracking applications, it is important to predict object motion to ensure that the matching of tracks and detections is achieved with the least possible error. A filter algorithm is used to help establish the tracking model, using the existing object information to predict future locations. The camera used has a frame rate of approximately 6 frames per second, so there is relatively little change between two adjacent frames. Accordingly, the location of the moving target is considered to have little change and a filter solution is used to estimate the object's location in a small range.

There are many filter algorithms applicable in tracking applications, including particle filter, low pass filter, and many others. Each filter has its own advantages and disadvantages. Per example, in the context of sensing multiple objects, the low pass filter has the lowest robustness in accuracy and precision but requires few computational resources, being a considerably efficient solution. Our option was the low pass filter, a fast and simple solution. The most common implementation of the low pass filter is the discrete Kalman filter. At each moment the Kalman filter estimates the object position and performs parameter correction.

This section described how the classical image processing methods identify a set of potential targets that can be imaged with zoom and afterwards processed with the deep learning model that will be described in the following section.

### B. Identification/Deep Learning

The present section describes the deep learning model. The goal here is to classify images as aircraft, drones, birds or clouds. The flow proceeds as follows. First, the model receives, as input, images of moving objects that were previously considered to be potential targets of interest by the tracking system. The criteria for triggering the deep learning classification is the successful tracking of the same moving object for more than $\tau_p$ frames. It is assumed that the images are subject to considerable zoom before being fed to the deep learning model. The output here is a classification in the

form of an array of probabilities for each considered class (aircraft, drones, birds, clouds). Below is described the types of deep learning models and methods used to accomplish this classification.

*1) Convolutional Neural Networks (CNN)*

A CNN is composed of convolutional, pooling and fully connected layers. The objective of these layers in the CNN is to create high order features that improve the classification efficiency. The convolutional layers consist of a set of feature maps with an associated receptive field of the size of only a small region of the input spectra. Each feature map output corresponds to the convolution (dot product) between the receptive field weights and all image points. This means that contiguous points in the feature map were determined in overlapping and contiguous regions of the input image. This way, several features are determined over the whole image. The big advantage of using convolutional layers is that they have much less weights to be learned than a fully connected neural network. The pooling layer is applied to the feature maps and performs a down-sampling. In the present work, the feature maps were divided into non-overlapping regions and their maximum taken. After the convolution and pooling layers, a fully connected neural network with one hidden layer processes the features coming from the previous layers.

*2) Residual Networks (ResNet)*

ResNet, an enhanced type of CNN, was created to solve the problem that deeper neural networks were providing worse results than shallower networks. They also have less parameter to train, than other previous, very deep architectures such as Inception, being therefore less computationally demanding. The way to be able to build deeper networks was the introduction of shortcut connections that allow for residual learning. The shortcut connection is a direct connection from the input to the output of the various (residual) modules that compose a ResNet. Residual learning means that instead of having to learn a mapping from input to output, called H, one can use each module to approximate a residual function F = H-x where x is the module input. The x is propagated by the shortcut connections. Even though the modules are capable of learning H or F it is easier to learn the F functions. The shortcut also allows for a better propagation of the training gradients, which is fundamental to create deeper networks. The residual modules that compose the ResNet are formed by a 1x1 convolution, followed by a 3x3 convolution and another 1x1 convolution. This module is said to have a bottleneck design. The 1x1 convolutions are used for reducing and increasing the dimensions of the information flowing though the network. The dimensions are reduced before the expensive 3x3 convolutions and are increased after. The 1x1 convolution corresponds to multiplying a single input pixel from various feature maps by the values of a filter and passing the result through a nonlinearity. This can be seen as applying a perceptron to the input.

*3) Transfer learning*

The present work used transfer learning meaning that a ResNet-50 developed for ILSVRC competition was retrained for our problem. The ILSVRC consists of classifying images into 1000 different classes that are not those of the present problem. The reason to use the ResNet-50 trained for ILSVRC comes from the fact that it has learned to extract a large number of high-level features that are useful for the current task. Consequently, the training requires smaller/simpler weight adjustments in order to create the network for the desired task. With this procedure it was necessary to remove the top layer of the ResNet, which had a 1000 classes output, and add a new top layer with four outputs, one for each of our classes. Transfer learning has the advantage of allowing to successfully training deep networks, such as ResNet, using a smaller number of training patterns when compared to having to train them from random weights or other types of weight initialization.

## IV. Datasets and Experimental Setting

### *A. Data Collection*

Two datasets were employed. The first, to evaluate the tracking, consisted of a video of a drone flying over an open field in Leiria, Portugal. While the drone was flying over, an off-the-shelf camera automatically tracked potential moving objects. This dataset was not used for testing the deep learning algorithms because there were no images of birds and aircrafts. The second dataset, for deep learning, consisted of images of aircraft, drones, birds and clouds, collected from the internet. By doing this it was assumed that, in the future, with the constant technological evolution, the video and tracking systems are capable of providing images of comparable quality to those gathered in the internet. Some image examples are shown in Figure IV-1. The aircraft class contained both airplane and helicopter 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, even though in many cases this was not possible. In this situation, the presence of other elements in the image was minimized. The total number of images for the classes aircraft, drones, birds and clouds were 2452, 2491, 2545 and 2758, respectively. The dimensions of images found in internet were variable, but they were mostly large images with more than 100x100 pixels. The images were resized to the proper input dimensions of the used neural networks. The number of images was augmented 29 times by generating new images from the original ones, which was done applying rotation, shift, shear, zoom and flip.

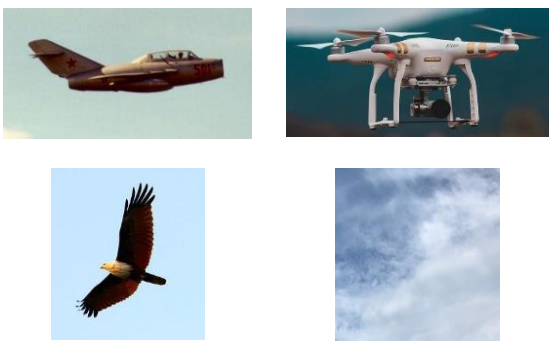

Figure IV-1: Images of the four classes.

### B. *Evaluation Method*

To have a rough sketch of the tracking system capabilities, 1 minute and 32 seconds of video with two flying drones, acquired with the system processing six frames per second (fps), were analyzed in terms of detection accuracy, average length of correctly tracked trajectory, average length of not tracked trajectory and false positives. To train and evaluate the deep learning classifiers, the data were split into three sets for training, validation and testing. The training set was used to adjust the CNN or ResNet weights, while the validation set was used to choose the hyper parameters providing the best results. Once the best hyper parameters were chosen, the final network generalization ability was assessed on the test set. The split used for each class was 1000 images for training, 500 for validation and the remaining for test which resulted in a total of 4000, 2000 and 4246 images for training, validation and test, respectively. Only the training images were augmented.

### C. *Configuration of Deep Learning Models*

The work comprised the creation of CNN and ResNet-50 with four output neurons, one for each class. The software was implemented in KERAS (https://keras.io/) to run in graphical processing units. The training error was categorical cross entropy and the training algorithm the stochastic gradient descent. The used activation functions were rectified linear units except for the output neurons that had a softmax. The CNN was composed of four modules, with a convolution layer and a max pooling each, followed by a fully connected network and an output layer. The input images had 50x50 pixels and were grayscale or color. The ResNet-50 had an input convolutional layer and max pooling, followed by 48 residual modules. On top there was a fully connected network. The ResNet input images had 50x50 and 101x101 pixels and three colors. ResNet were trained with transfer learning while CNN were not. Aircraft, drone, birds and clouds have the same apparent size for the CNN and ResNet which is typical in ILSVRC; however, their shapes and remaining features are different which allows proper classification.

## V. RESULTS

The drones were tracked with an accuracy of ~81% on the 553 frames of the video. The identified tracks had an average length of ~35 ± 20 continuous frames. The trajectory parts that were not correctly identified had an average length of ~42 ± 11 frames. Moreover, it was possible to track small targets of ~3 pixels, see Figure V-1a, as well as larger objects, see Figure V-1b. The motion and shape models allowed to distinguish the drone, and its distinct movements from most of the background artifacts in the sky. However, there were 7 false targets that were persistently detected and tracked for more than 30 frames. These were caused by cloud movement and image noise. The previous values regarding the tracking system are only indicative since final values would require more extensive testing.

The results obtained with the deep networks are shown in Table 1. A good indication of the generalization ability of the models is the small difference in results between the validation and the test results. In Table 1 the worse results were obtained with the relatively shallow convolutional neural network number 1, with grayscale input images. In this situation, the classes aircraft and birds exhibit classification efficiencies smaller than 90% in test. In CNN number 2, keeping the image size and changing to three colors has the largest impact in the birds and clouds classes in test but is still insufficient to make birds class pass the 90% threshold. When looking at the ResNet-50, number 3, with images of input size 50x50 pixels and with three colors, the lowest classification efficiency in test is 94.7% for birds, which means a significant improvement with respect to the CNN. In fact, the smallest improvement in test when changing from CNN to ResNet-50 with 50x50 color images was 3.3 percentage points (p.p.) in the clouds class and reached 5.3 p.p. in birds. Finally, when increasing the size of the input images from 50x50 pixels, in ResNet number 3, to 101x101 pixels, in ResNet number 4, while keeping three colors, test results larger than 98% were obtained. It is interesting to observe that clouds are the easiest class to separate, with 100% classification efficiency, probably due to the absence of well-defined shapes in images. The second best class is birds with 98.6% efficiency, followed closely by drones and aircrafts with 98.2% and 98.1%, respectively. In the 4246 test images only 51 were misclassified making it hard to find misclassification causes, however, there are some candidates such as the objects being blurred, off-center or occupying a small percentage of the image area.

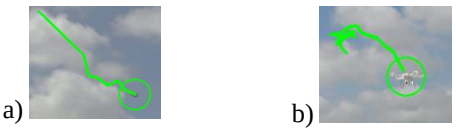


Figure V-1: Images of tracking a quadcopter in video.

## VI. CONCLUSIONS

The present work advances the state-of-the-art by, to the authors' knowledge, being the first to present a system capable of tracking and classifying aircraft, birds and drones. We were also the first to use residual neural networks for the classification system. The correct classification percentages larger than 98%, in test, for all classes, surpass those of the work of Liu *et al.* and Unlu *et al.* and are comparable to those of Schumann *et al.*, even though these results were

TABLE I. PERCENTAGE OF CORRECTLY CLASSIFIED IMAGES (RECALL), IN VALIDATION AND TEST SETS, FOR EACH CLASS. THE BEST RESULT IS SHOWN IN GREY.

| Number | Network type | Image (pixel x pixel x color) | Validation | | | | Test | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Aircraft | Birds | Clouds | Drones | Aircraft | Birds | Clouds | Drones |
| 1 | CNN | 50x50x1 | 88.4 | 87.6 | 92.2 | 88.2 | 89.8 | 85.0 | 90.7 | 92.6 |
| 2 | CNN | 50x50x3 | 91.4 | 91.2 | 96.2 | 91.8 | 90.5 | 89.4 | 96.1 | 92.8 |
| 3 | ResNet-50 | 50x50x3 | 96.6 | 96.4 | 99.8 | 96.4 | 95.3 | 94.7 | 99.4 | 96.5 |
| 4 | ResNet-50 | 101x101x3 | 98.0 | 99.3 | 100 | 97.9 | 98.1 | 98.6 | 100 | 98.2 |

obtained with different datasets. Moreover, we have a more encompassing model that distinguishes birds from aircraft and drones. Liu *et al.* does not separate birds, while Unlu *et al.* and Schumann *et al.* do not identify aircrafts. In addition, the three works employ significantly smaller test sets than the present work. The use of residual networks allowed to improve the detection efficiency in all classes, by at least 3.3 percentage points, with respect to a standard convolution neural network with four convolution layers. The preliminary results from the simple evaluation of the tracking methods open up good perspectives for the future use of these methods.

As future work we intend to perform field tests where all the components of the project will be tested simultaneously (camera, tracking system and deep learning). It will be relevant to compare the quality of the images provided by the system camera and those used to create the ResNet because large differences might have an impact on the system identification performance. Finally, we would like to point out that the developed detector can also be used in airport surveillance where drones and birds pose a serious safety risk.

## 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] M. Phillips and J. Kuhns, "Illicit Drug Trafficking," *Transnatl. Crime Glob. Secur.*, pp. 3–22, 2018.

[2] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[3] R. Rout, "A survey on object detection and tracking algorithms," National Institute of Technology Rourkela, Rourkela, India, 2013.

[4] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, Nov. 2011.

[5] C. Huang, B. Wu, and R. Nevatia, "Robust Object Tracking by Hierarchical Association of Detection Responses," in *Computer Vision – ECCV 2008*, vol. 5303, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 788–801.

[6] W. Luo *et al.*, "Multiple Object Tracking: A Literature Review," *ArXiv14097618 Cs*, Sep. 2014.

[7] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logist. Q.*, vol. 2, no. 1–2, pp. 83–97, Mar. 1955.

[8] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[9] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-Based Particle Filter for Tracking Multiple Interacting Targets," in *Computer Vision - ECCV 2004*, vol. 3024, T. Pajdla and J. Matas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 279–290.

[10] B. Sahbani and W. Adiprawita, "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)*, Bandung, Indonesia, 2016, pp. 109–115.

[11] E. Hamuda, B. Mc Ginley, M. Glavin, and E. Jones, "Improved image processing-based crop detection using Kalman filtering and the Hungarian algorithm," *Comput. Electron. Agric.*, vol. 148, pp. 37–44, May 2018.

[12] Y. Zhao, Q. Chen, W. Cao, J. Yang, J. Xiong, and G. Gui, "Deep Learning for Risk Detection and Trajectory Tracking at Construction Sites," *IEEE Access*, vol. 7, pp. 30905–30912, 2019.

[13] E. Unlu, E. Zenou, and N. Riviere, "Using Shape Descriptors for UAV Detection," *Electron. Imaging*, vol. 2018, no. 9, pp. 128-1-128–5, Jan. 2018.

[14] D. H. Ye, J. Li, Q. Chen, J. Wachs, and C. Bouman, "Deep Learning for Moving Object Detection and Tracking from a Single Camera in Unmanned Aerial Vehicles (UAVs)," *Electron. Imaging*, vol. 2018, no. 10, pp. 466-1-466–6, Jan. 2018.

[15] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting Flying Objects Using a Single Moving Camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 879–892, May 2017.

[16] J. James, J. J. Ford, and T. L. Molloy, "Learning to Detect Aircraft for Long-Range Vision-Based Sense-and-Avoid Systems," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4383–4390, Oct. 2018.

[17] S. Hwang, J. Lee, H. Shin, S. Cho, and D. H. Shim, "Aircraft Detection using Deep Convolutional Neural Network in Small Unmanned Aircraft Systems," in *2018 AIAA Information Systems-AIAA Infotech @ Aerospace*, Kissimmee, Florida, 2018.

[18] A. Schumann, L. Sommer, J. Klatte, T. Schuchert, and J. 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)*, Lecce, Italy, 2017, pp. 1–6.

[19] M. Saqib, S. Daud Khan, N. Sharma, and M. 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)*, Lecce, Italy, 2017, pp. 1–5.

[20] Y. Chen, P. Aggarwal, J. Choi, and C.-C. J. Kuo, "A Deep Learning Approach to Drone Monitoring," *ArXiv171200863 Cs*, Dec. 2017.

[21] M. Farhadi and R. Amandi, "Drone detection using combined motion and shape features," in *IEEE International Workshop on Small-Drone Surveillance, Detection and Counteraction Techniques*, Lecce, Italy, 2017.

[22] H. Liu, F. Qu, Y. Liu, W. Zhao, and Y. Chen, "A drone detection with aircraft classification based on a camera array," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 322, p. 052005, Mar. 2018.

[23] C. Aker and S. Kalkan, "Using Deep Networks for Drone Detection," *ArXiv170605726 Cs*, Jun. 2017.

[24] A. Coluccia *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, 2017, pp. 1–6.

[25] M. Lavanya, "Real Time Motion Detection Using Background Subtraction Method and Frame Difference," *Int. J. Sci. Res. IJSR*, vol. 3, no. 6, pp. 1857–1861, 2014.

[26] N. Singla, "Motion Detection Based on Frame Difference Method," *Int. J. Inf. Comput. Technol.*, vol. 4, no. 15, pp. 1559–1565, 2014.

[27] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, "POI: Multiple Object Tracking with High Performance Detection and Appearance Feature," *ArXiv161006136 Cs*, Oct. 2016.