ORIGINAL ARTICLE

Improving visual differentiation of drones and birds in aerial surveillance using trajectory features

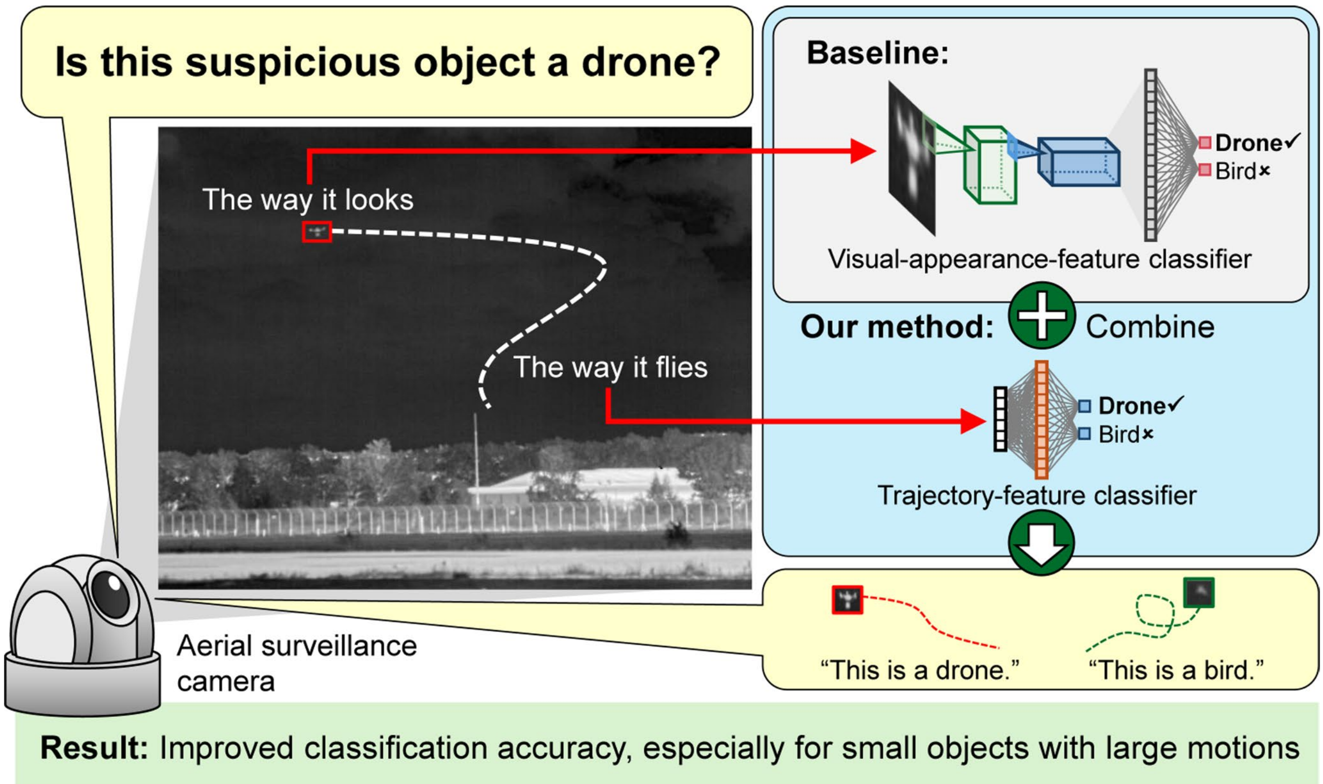
Salil Luesutthiviboon · Guido C. H. E. de Croon · Anique Altena · Mirjam Snellen · Mark Voskuijl

Received: 11 June 2025 / Accepted: 16 March 2026 / Published online: 12 June 2026

Abstract Detecting malicious drones using aerial surveillance cameras is challenging when the distance is large, because the drone then occupies only a few pixels. Current optical detection methods rely mostly on visual appearance features. Hence, they struggle to differentiate drones from other flying objects, especially birds, when the apparent object size is small. Fortunately, the observed trajectory over time can help improve the differentiation accuracy. Here, we propose to combine classification neural networks of the object's trajectory features and visual appearance features. We train and test the networks using our dataset containing infrared videos of drones and birds, where the variation of drone configurations and flight patterns is relatively larger than other publicly available datasets. We show that, particularly for small objects with high motion, the inclusion of trajectory features for visual classification achieves up to 22% higher frame-wise classification accuracy compared to when only visual appearance features are used. We further demonstrate that integrating both feature types provides improved accuracy over all of the considered trajectories, with 4% more of the trajectories being classified correctly. Consistent results are also shown on an open dataset, confirming the generalizability. Our study demonstrates the crucial role of information beyond the frame-wise visual appearance features in extending the operational range of aerial surveillance cameras.

https://doi.org/10.1007/s00521-026-12080-5

Graphic abstract



Keywords Drone detection · Aerial surveillance camera · Small object classification · Trajectory feature classification

1 Introduction

The ever-increasing availability of consumer drones has led to a growing number of drone owners worldwide [1]. However, drone misuse can pose security threats and cause major disruptions to critical infrastructures [2], such as recent shutdowns of airports due to drone incursions [3, 4]. This triggers an urgent need for aerial surveillance systems that provide early warnings for the presence of drones, so that proper and timely countermeasures can be imposed.

An effective aerial surveillance system for drone detection combines multiple sensors, each providing unique functionalities and advantages [5–7]. In favorable visibility conditions, cameras play a role in verifying detections made by other means of detection [8–10], such as acoustic sensors [11, 12], radio frequency [13, 14], and radars [15, 16], pinpointing the drone location, thanks to the relatively narrower field of view and a finer angular resolution. Cameras can also help identify specific drone types and possible threats [17, 18].

To detect and track drones in aerial surveillance video frames, computer vision algorithms are used [19]. Many state-of-the-art computer vision algorithms contain deep-learning components [20], particularly Convolutional Neural Networks (CNNs) [21], which are trainable prediction models capable of analyzing visual appearance features in video frames to make predictions.

However, reliance of the CNNs on the visual appearance features in the drone detection task could make them fall short in certain conditions, such as low visibility due to weather [22], complex backgrounds [23], and small apparent object sizes [24, 25]. While the low visibility and complex backgrounds could be overcome by specialized cameras (e.g., infrared and event-based cameras [26, 27]) or advanced frame enhancement techniques [28, 29], the small apparent object size remains a critical challenge. If drones are only recognized when their apparent size is large enough, the drone might already be too close to take subsequent actions. On the other hand, at larger distances, drones appear in the video frame as moving dots, occupying a few pixels [30, 31]. This makes it challenging to distinguish the drones from other flying objects, especially birds [32, 33].

Fortunately, the drone's movement could provide valuable cues that differentiate it from other flying objects. This has been shown in previous works. For example, Han et al. [34] studied videos of birds and quadrotor drones flown to mimic the birds. They showed that the drones' apparent trajectory kinematic features, such as the median velocity and acceleration, differ significantly from those of the birds. Therefore, it is possible to create a classifier that differentiates drones from other flying objects based on the trajectory features. Preliminary studies were carried out by Srigarom et al. [35] and Chan et al. [36]. They showed that Support Vector Machine (SVM) classifiers can differentiate the drones from the other flying objects with more than 80% accuracy, based only on the trajectory features.

In this work, we propose a classifier that combines both the visual appearance features and the trajectory features to improve the differentiation between drones and birds. In particular, we analyze videos of drones and birds from a ground-based aerial surveillance infrared camera system collected by ourselves, where there is a relatively larger variation of drone configurations, flight patterns, and apparent sizes than other datasets. We extract and distinguish two types of visual information: firstly, information on the visual appearance of the object, and secondly, the trajectory information, describing the object position over time. We build a CNN for visual appearance feature classification that represents a classical frame-based object classification method. Next to this, we extract object trajectory feature information. We build a classification network based on the extracted trajectory features. Then we propose a classification network that combines both the trajectory and visual appearance features. Our key contributions are:

- We propose a method for differentiating drones and birds in aerial surveillance videos using trajectory features to complement the visual appearance features. We evaluate the classification accuracy of our proposed method and compare it against a baseline method relying only on the visual appearance features, using both a new dataset and an existing open dataset.
- We identify specific scenarios where the trajectory features enhance classification accuracy of drones and birds, and subsequently help to extend the aerial surveillance camera operational range.
- We present and analyze a new dataset containing bounding-box-labeled video snippets of various drones and birds, captured using an aerial surveillance infrared camera.

2 Related work

2.1 Detection-based vs. detection-free tracking of drones

The task of localizing, tracking, and classifying objects in aerial surveillance videos to identify malicious drones falls under the object detection and tracking task in computer vision. Various approaches to this problem have been presented in recent literature [7]. According to Luo et al. [37] and Yang and Nevatia [38], the approaches can broadly be categorized in two groups: detection-based and detection-free tracking.

Most visual tracking of drones in literature relies on the so-called "detection-based tracking" framework. In this framework, incoming video frames are processed by object detection algorithms, such as You Only Look Once (YOLO) [18, 22, 39, 40] or Single-Shot Detector (SSD) [21, 41]. These algorithms contain deep-learning components, such as the CNNs mentioned earlier, which extract and analyze visual appearance features. Based on prior knowledge obtained during training, the CNN determines whether a drone is present, and if so, its location. This is typically done frame-by-frame. Afterwards, detections across multiple frames are associated with a single identity, i.e., the drone is tracked. To achieve tracking, visual appearance features of the object from the preceding frame are used as a matching template to help locate the drone in subsequent frames. This methodology results in a complete detection-and-tracking architecture, such as SiamRPN [42] and DeepSORT [20].

The detection-based tracking strategy requires accurate and continuous detections across the frames. However, as discussed earlier, the CNNs may fail to meet this requirement in certain situations, such as when the apparent object size is small [24, 25] or when the drone appearance differs from the training data. Consequently, these limitations could restrict the use cases, effectiveness, and operational range of the aerial surveillance camera.

Therefore, we propose a method to be integrated with the "detection-free tracking" framework to avoid relying primarily on the object's appearance. In this framework, a suspicious flying object is detected and tracked in the video frame, without assigning a specific class to it. The classification is done in a subsequent step. Seidaliyeva et al. [43], for instance, employed this detection-free tracking architecture for drone detection. In their work, moving objects are detected by background subtraction. Then these objects are classified based on their visual appearance using a MobileNetV2 [44] CNN.

Other algorithms can also be used to capture the suspicious flying object in the first step of the detection-free tracking framework. Examples are optical flow [45, 46], change detection, frame differencing [30, 47], and bio-inspired frame enhancement [29]. Our present work focuses on the subsequent classification step.

2.2 Trajectory-based drone vs. bird classification

When the apparent object size is small and the visual information is limited, differentiating drones from other flying objects, especially birds [32, 33], becomes challenging. However, the motion pattern, i.e., apparent trajectory, of the flying object can reveal crucial information about its true class [48].

Han et al. [34] analyzed video snippets of birds and quadrotor drones manually flown to mimic the birds. They extracted and compared various trajectory kinematic features of the drones and the birds. They found that the birds' median velocity, acceleration, and heading change rate are significantly higher than those of the drones. Additionally, the birds exhibit larger trajectory fluctuation amplitudes than the drones. These findings suggest that trajectory features can be used as a basis for object classification.

Based on this concept, Srigarom et al. [35] and Chan et al. [36] extracted trajectory features from approximately 50 video snippets of drones, birds, and kites. They applied principal component analysis to the extracted trajectory features to reduce the feature dimensionality. Subsequently, they applied Support Vector Machine (SVM) to differentiate the object class. They found that the trajectory-based SVM classification gives more than 80% accuracy in differentiating the drones from the other flying objects. The authors recommended combining the trajectory features with the visual appearance features to obtain a more accurate classification of flying objects.

Our work is motivated by the aforementioned findings and their recommendations. We expect that the integration of the trajectory features with the visual appearance features will overcome the limitations of aerial surveillance cameras in cases where the object size is small.

2.3 Datasets for drone vs. bird classification

A dataset to study drone vs. bird differentiation in aerial surveillance using trajectory information must meet several requirements. First, it should provide video sequences rather than isolated frames. Next, it is also important that the dataset contains a balanced number of drone and bird frames. Finally, an extensive variation of drone configurations, e.g., fixed- and rotating-wing drones flown both manually and autonomously to represent real-world operations, is preferred.

A comprehensive survey of counter-drone datasets and approaches is given in the recent work of Dong et al. [49]. Examples include the dataset of Hui [50], the DUT Anti-UAV dataset [51], the Anti-UAV dataset [52], and the Anti-UAV410 dataset [53]. These datasets focus exclusively on the drone class, mostly containing rotating-wing drones only. Some of the older datasets contain a relatively small number of 20 to 30 sequences, which might be insufficient for trajectory-based analysis.

A smaller group of datasets addresses the drone vs. bird discrimination. The Drone-vs-Bird dataset [32] provides videos of drones filmed by visible-light cameras, and birds that occasionally appear on the scene. This dataset highlights the challenge of robust drone detection while minimizing false alarms triggered by birds. However, this dataset remains drone dominant and still lacks bounding-box labeling of birds. Han et al. [34] presented a dataset focusing specifically on differentiating drone and bird trajectories in the work mentioned earlier. The labeled trajectories extracted from drone and bird video snippets are available. However, the drone variety is limited to small quadrotor drones flown to mimic the birds. Svanström et al. [54] presented a dataset that contains bounding-box labeled video snippets of drones and birds. This dataset offers a balanced number of both classes. We employ the dataset of Svanström et al. in our study. We will further demonstrate that this dataset only includes manually flown quadrotors filmed at close range, where a visual-appearance classifier already performs well.

In this work, we expand the data variety compared to the previous works. We present a dedicated dataset, containing drones in both fixed-wing and rotating-wing configurations, flown both manually and autonomously, with a varying filming distance. This provides us with extensive variation in the amount of visual appearance features. Besides, unlike many previous works where visible-light cameras were used, we use data from an infrared camera. This is because infrared cameras are more robust to varying daylight conditions, and are therefore more preferred for aerial surveillance. We also employ an equally large set of bird video snippets, filmed by the same camera, in our study. This is done to ensure a balance between the drone and the bird classes.

3 Methods

3.1 Overview of our method

We handle videos from a ground-based aerial surveillance camera system shown in Fig. 1a. We assume that a suspicious flying object has been detected and tracked in the video frame using one of the methods mentioned in Sect. 2.1. As a result, a bounding box around the suspicious object is provided, but its actual class is yet to be concluded. For a video frame with a suspicious flying object as shown in Fig. 1b, we extract and analyze two types of information: firstly, visual information, describing the appearance of the object, and secondly, the trajectory information, describing the object position over time up to the current frame. Next, we create a visual-appearance-feature classification CNN, an architecture as shown in Fig. 1c. This network is used as a baseline, representing a classical object classification method relying on its appearance. Next to this, we extract object trajectory feature information, namely (1) the median velocity, (2) the median acceleration, (3) the median heading change rate, and (4) tortuosity. All of the features are apparent values, i.e., as seen by the camera. Sections 3.2 to 3.4 describe the trajectory feature extraction methodology in detail. We build a classification network shown in Fig. 1d based on the extracted trajectory features. The mathematical symbols in this figure are described in the subsequent subsections. Then we combine the visual-appearance-feature classification network and the trajectory-feature classification network into one network as shown in Fig. 1e. In this network, we classify the suspicious flying object based on the concatenated visual appearance and trajectory features. Detailed description of our network architecture is provided in Sect. 3.5. We expect the combined network classification accuracy to outperform the baseline visual-appearance-feature classifier due to the inclusion of the trajectory features.

3.2 Trajectory reconstruction

We process the gray-scale infrared video snippets to reconstruct the trajectories. In every frame of the snippet, the object is pre-labeled by a ground-truth bounding box. Let an arbitrary video snippet contain a trajectory denoted by j. For every arbitrary snippet frame i of trajectory j, a bounding box is defined by $[p_b, q_b, P, Q]_{i,j}$, where p_b,

 https://doi.org/10.1007/s00521-026-12080-5

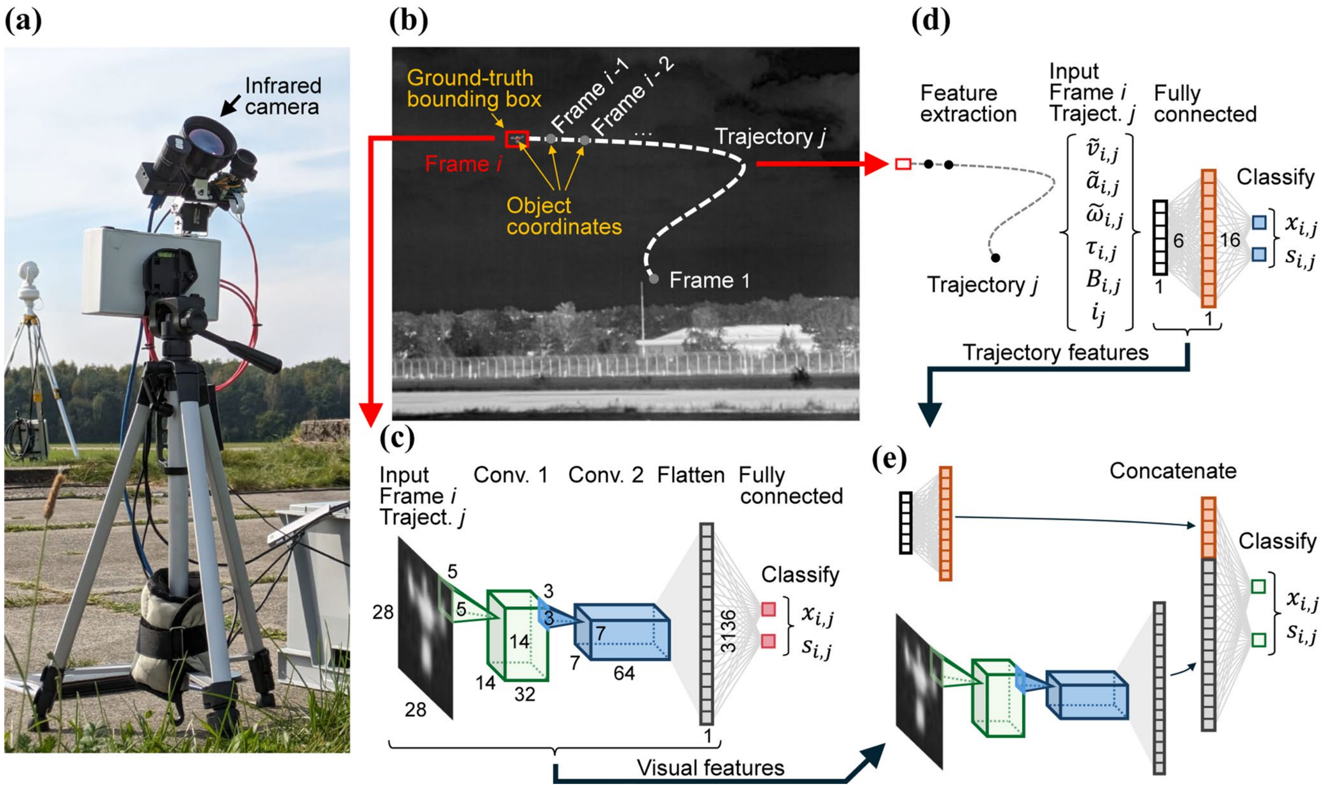


Fig. 1 Classification of a suspicious flying object in an aerial surveillance video. (**a**) Our aerial surveillance camera system. (**b**) Visual and trajectory information of the suspicious flying object in an aerial surveillance video frame. (**c**) A CNN for suspicious flying object classification based on the visual appearance features. (**d**) A classification network based on extracted trajectory features of the suspicious flying object. (**e**) A combined network for classifying the suspicious flying object based on both the visual and trajectory features

q_b, P, and Q are the bounding-box-corner horizontal and vertical pixel coordinates, pixel width, and pixel height, respectively.

We extract a coordinate pair in the bounding box to represent the object location. To extract this coordinate pair, we employed the so-called "squared gray-scale centroid" method, which has been shown to be one of the most simple and accurate methods to extract sub-pixel locations of a target in an image [55]. Let $G_{i,j}(p,q)$ denote the gray-scale pixel value of a video frame i belonging to a trajectory j, at a pixel coordinate (p, q), where the higher $G_{i,j}(p,q)$ means the brighter pixel, the object pixel coordinates ($p_{i,j}$, $q_{i,j}$) are calculated by

$$p_{i,j} = p_b^{i,j} + \frac{\sum_{q=1}^{Q_{i,j}} \sum_{p=1}^{P_{i,j}} p G_{i,j}^2(p_b^{i,j} + p - 1, q_b^{i,j} + q - 1)}{\sum_{q=1}^{Q_{i,j}} \sum_{p=1}^{P_{i,j}} G_{i,j}^2(p_b^{i,j} + p - 1, q_b^{i,j} + q - 1)} \tag{1}$$

and

$$q_{i,j} = q_b^{i,j} + \frac{\sum_{p=1}^{P_{i,j}} \sum_{q=1}^{Q_{i,j}} q G_{i,j}^2(p_b^{i,j} + p - 1, q_b^{i,j} + q - 1)}{\sum_{p=1}^{P_{i,j}} \sum_{q=1}^{Q_{i,j}} G_{i,j}^2(p_b^{i,j} + p - 1, q_b^{i,j} + q - 1)}. \tag{2}$$

Next, for every video sequence, we construct the trajectory by connecting all of the extracted representative coordinate points. We applied a Gaussian weighted moving average smoothing, with a window length corresponding to one second, to the data to dampen artifacts caused by small vibrations of the camera. We qualitatively checked that the characteristic trajectory shape is conserved after the filter had been applied. We use the hat symbol to denote the smoothened coordinate point as follows: $(\hat{p}_{i,j}, \hat{q}_{i,j})$.

3.3 Trajectory feature calculation

Given the trajectory point across all of the snippet frames, we extract trajectory features. We consider the following features:

1. Velocity $\tilde{v}$ (pixels/s)
2. Acceleration $\tilde{a}$ (pixels/s^2)
3. Heading change rate $\tilde{\omega}$ (degrees/s)
4. Tortuosity τ (-)

The velocity, acceleration, and heading change rate are the features used by Han et al. [34]. The tortuosity is introduced in this work. Details on the calculation of these features are provided in the following.

3.3.1 Velocity, acceleration, and heading change rate

Let $\Delta t_{i,j} = t_{i,j} - t_{i-1,j}$ denote the time step between frames i and $i-1$ in trajectory j. We define a velocity vector for frame i in trajectory j comprising of a horizontal and vertical component as $\vec{v}_{i,j} = [(\hat{p}_{i,j} - \hat{p}_{i-1,j})/\Delta t_{i,j}, (\hat{q}_{i,j} - \hat{q}_{i-1,j})/\Delta t_{i,j}]^T$, where T denotes the transpose. Based on this velocity vector, we calculate the trajectory features; namely, the velocity $v_{i,j}$, the acceleration $a_{i,j}$, and the heading change rate $\omega_{i,j}$, as follows:

$$v_{i,j} = \|\vec{v}_{i,j}\|, \tag{3}$$

$$a_{i,j} = \frac{|v_{i,j} - v_{i-1,j}|}{\Delta t_{i,j}}, \tag{4}$$

$$\omega_{i,j} = \left| \cos^{-1}\left(\frac{\vec{v}_{i,j} \cdot \vec{v}_{i-1,j}}{\|\vec{v}_{i,j}\| \; \|\vec{v}_{i-1,j}\|} \right) \right| \Big/ \Delta t_{i,j}. \tag{5}$$

The aforementioned trajectory feature calculations are in accordance with Han et al. and Vlachos et al. [34, 56]. We however do not consider the trajectory fluctuations as in their work, or the periodicity as in Srigrarom et al. [35]. The motivation for excluding this is twofold. Firstly, adding this feature would require a higher frame rate and frequency-domain processing of the data. This contrasts with our intention to keep our method quick and simple. Secondly, the object should be close enough to the camera to effectively observe such fluctuations. In the cases where the object is close to the camera, we already expect the classification based on the visual appearance features to accurately deduce the object class.

Finally, we determine the median value of the extracted features. For frame i in trajectory j, we use all of the values from frame 1 to compute the median values. We use the tilde symbol to denote the median values as follows: $\tilde{v}_{i,j}$, $\tilde{a}_{i,j}$, and $\tilde{\omega}_{i,j}$.

3.3.2 Tortuosity

In addition to the aforementioned features, we propose to include a new feature, the so-called tortuosity τ. This is based on the observation made by Srigrarom et al. [35] that the drones tend to have a more straight flight path than the birds. The tortuosity is a general measure of the twist or crookedness of a path that has been employed in a wide range of contexts, such as fluid mechanics [57] and animal movement studies [58]. In this work, for an arbitrary frame i in trajectory j, denoted by i_j, the tortuosity $\tau_{i,j}$ is defined as a ratio between the distance traveled and the straight-line distance, and has a minimum value of 1,

$$\tau_{i,j} = \begin{cases} 1, & \text{for } i_j = 1 \\ \dfrac{\text{Distance traveled frame 1 to } i_j}{\text{Straight-line distance frame 1 to } i_j}, & \text{for } i_j > 1 \end{cases}. \tag{6}$$

Visualizations to highlight the difference between the tortuosity and the heading change rate are provided in Fig. 2. Objects with high tortuosity are illustrated in the right part of Fig. 2. Such objects travel from the start to the end point over a longer distance than a straight-line path from the start to the end point would have taken. This is often characterized by loops or crossing of its preceding path. On the contrary, objects with a high heading change rate do not necessarily have a high tortuosity. This is illustrated in the lower left part of Fig. 2. Unlike a trajectory with high tortuosity, a trajectory with a high heading change rate may appear to be a straight path overall, but occasional sudden changes in the heading angle lead to a high heading change rate.

It is worth emphasizing that, although omitted in the rest of this article for conciseness, the aforementioned features are apparent magnitudes. This means that they are the values as observed by the camera, which is placed at any arbitrary orientation with respect to the earth. Here, we assume that the camera is the only information source and the object's trajectory with respect to the earth is unknown. Therefore, the absolute values and directions of the features with respect to the earth reference frame are also unknown.

3.4 Quantifying trajectory and visual appearance information

Apart from calculating the trajectory features, we also quantify the amount of trajectory and visual appearance information in each frame. In this way, we can study the classifier performance variations with the amount of the available trajectory and visual appearance information in our further analysis. We define the following information quantification metrics.

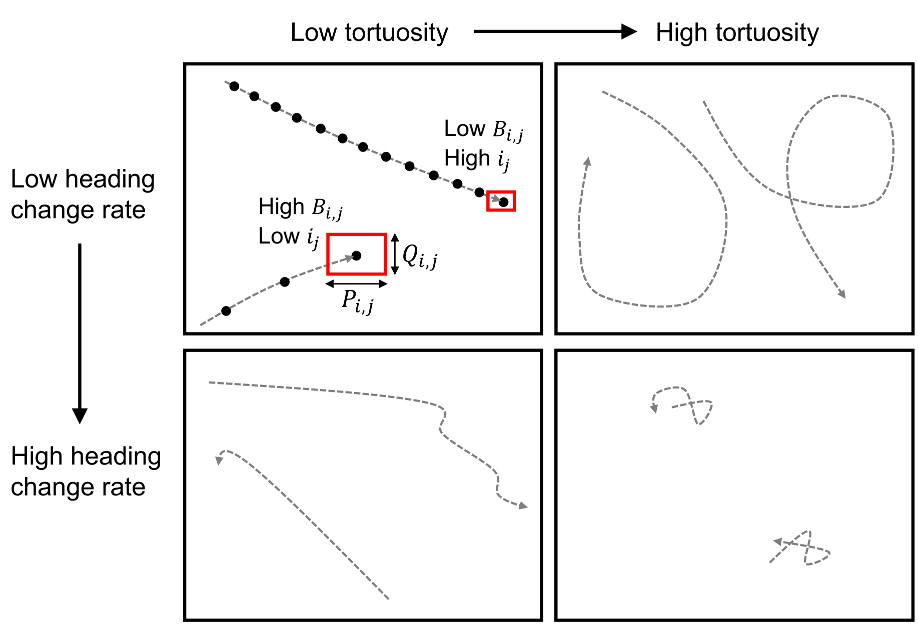


Fig. 2 Visualizations for interpretations of the heading change rate $\omega_{i,j}$, tortuosity $\tau_{i,j}$, trajectory length i_j, and ratio $B_{i,j}$

3.4.1 Trajectory length

The longer we observe an object trajectory, the more likely we pick up meaningful cues from its movement. Therefore, we propose to use the trajectory length, i.e., the frame index i_j, where j denotes the trajectory index, as a simple metric to quantify the amount of the trajectory information.

3.4.2 Ratio $B_{i,j}$

The amount of the object's visual appearance information depends on its apparent size in the the video frame. We therefore propose the ratio B, defined the bounding-box area divided by the frame area, both in pixels squared, as a metric to quantify the amount of visual information for each frame. For an arbitrary frame i in an arbitrary trajectory j, the ratio $B_{i,j}$ is

$$B_{i,j} = \frac{\text{Bounding-box area } (i,j)}{\text{Frame area}} = \frac{P_{i,j} \times Q_{i,j}}{\text{Frame area}}. \tag{7}$$

Visualizations to help interpret the trajectory length i_j and the ratio $B_{i,j}$ are also provided in Fig. 2, in the top left part.

3.5 Classifier architecture

3.5.1 Visual-appearance-feature classifier

The visual-appearance-feature classifier shown in Fig. 1c is a CNN-based classification network containing two convolution layers. Such a CNN architecture is typical for classifying small gray-scale images, such as images of clothing items in the so-called "Fashion-MNIST" dataset [59], and is therefore suitable for our application. Its input is the bounding-box cropped and resized 28 × 28 pixels image. This selected input image size is much larger than the average bounding box size in our dataset (will further be demonstrated in Fig. 9b). The risk of losing the visual appearance information due to the resizing process is therefore minimal. The first convolution layer performs convolutions of this input image using 32 5 × 5 kernel with a stride of 2, followed by a batch normalization, a Rectified Linear Unit (ReLU) activation, and a 2 × 2 max pooling with a stride of 1. The output is passed on to the second convolution layer which performs convolutions using 64 3 × 3 × 32 kernel with a stride of 2 × 2 followed by a batch normalization and a ReLU activation. Finally, the convolution-layer output is flattened and connected to a softmax function. The resulting data dimensions from the main stages of this classifier are indicated in Fig. 1c. There are in total 25,794 learnable properties in this network. Our proposed CNN architecture is relatively shallow compared to widely known architectures such as the MobileNetV2 [44] and the ResNet-18 [60], which have been implemented for the visual-appearance based drone detection [43] and drone-vs-bird classification [61] tasks. We compare performance and runtime of our CNN to these well known CNNs in Sect. 5.4.2.

3.5.2 Trajectory-feature classifier

The trajectory-feature classifier is shown in Fig. 1d. This classifier is a neural-network classifier. For frame i in trajectory j, the classifier takes six inputs, namely, $\tilde{v}_{i,j}$, $\tilde{a}_{i,j}$, $\tilde{\omega}_{i,j}$, $\tau_{i,j}$, $B_{i,j}$, and i_j. Note that $B_{i,j}$, and i_j are also included to provide information on the apparent object size and the amount of visual and trajectory information. The trajectory-feature classifier contains one hidden layer with 16 neurons. This number of neurons is chosen based on a sensitivity study. It has been found that doubling the number of neurons to 32 only increases the average classification accuracy by less than 1%. The hidden layer is followed by a batch normalization, and a ReLU activation. The output is then connected to a softmax function. There are in total 178 learnable properties in this network.

 https://doi.org/10.1007/s00521-026-12080-5

3.5.3 Combined classifier

The combined classifier is built by concatenating the flattened layer of the visual-appearance-feature classifier and the hidden layer of the trajectory-feature classifier as shown in Fig. 1e. The concatenated layer is then connected to a softmax function.

4 Datasets, training, and performance evaluation metrics

4.1 Our dataset

Our dataset contains 200 video snippets, 100 of drones and 100 of birds, filmed by a static ground-based infrared camera. The considered snippets have varying numbers of frames. In total, we consider 4,792 frames containing the drones with a variety of configurations and flight phases, and 4,226 frames containing the birds. In each frame, the ground-truth object class and location have been hand-labeled, providing a ground-truth bounding box.

The videos in our dataset were filmed during multi-day drone-detection exercises at two former air bases in the Netherlands; at the Valkenburg Air Base (29 to 30 June 2023), currently named the Unmanned Valley, and at De Peel Airbase (13 to 16 September 2023). Our filming took place during daytime. The visibility was clear and there was no rain, snow, or mist. However, the cloud cover varied largely from clear sky to overcast. We validated our weather observation with the open data of the Royal Netherlands Meteorological Institute (KNMI) [62], which reports the minimum visibility of 8 km and the cloud cover from 0 to 8 oktas at weather stations close to our measurement locations.

Our aerial surveillance camera system is shown in Fig. 1a. The infrared videos were filmed by a FLIR Boson+ uncooled long-wave infrared camera module equipped with a 73-mm lens providing a 5.5-degree horizontal field of view. Next to the infrared camera, we installed a visible-light Arducam 12MP IMX477 camera with a controllable lens providing a field of view between 33 and 96 degrees. Both the infrared and the visible-light cameras were installed on two RDS3115MG servos enabling panning and tilting. We manually controlled the servos via an Arduino Uno microcontroller board. In this work, we only consider the moments where the cameras were static. We developed a MATLAB program to sample the frames from both cameras using a sampling rate of approximately 10 Hz. For each sample, a 640×512 pixel frame from the infrared camera, a 1280×960 pixel frame from the visible-light camera, and the timestamp were collected.

Multiple rotating-wing and fixed-wing drones were flown during the exercises in different flight patterns, such as vertical take off and landing, straight line, and hover, to simulate different operational intentions, such as reconnaissance, attack, and escape. To represent realistic drone operations, we intentionally had no influence on the drone flight paths and control methods (manual or autopilot). These were decided by the operators. We only had general prior knowledge of the flight area and filmed the drones as passive observers. The considered drones and their dimensions are presented in Table 1. The composition of the drone configurations and flight phases are shown in Figs. 3a and 3b, respectively. The camera-to-drone ground distance varied up to approximately 1000 m and the altitude varied up to approximately 120 m. We also filmed birds that naturally appeared in the scene. The background of our scenes ranged from clear to cloudy skies, and occasionally trees when the drones took off or landed. During the filming, we took note of the true object class in each video as initial labeling. We also verified our initial labeling in two ways. First, by comparing the video time and the Global Positioning System (GPS) log provided by the organizers; and, second, by examining the videos filmed concurrently by the visible-light camera which provides a broader overview of the scene.

Next, labeling of the object class and location was done by manually drawing a ground-truth bounding box in each video frame using the MATLAB video labeler tool. The labeling process was done by one annotator. MATLAB's built-in temporal interpolator allowed some parts of the labeling process to be automated, such as when the object moves in a straight line with a constant velocity. After the labeling process, we regenerated the video

Table 1 Drones present in our dataset and their dimensions

Model	Diagonal length (m)
Rotating wings during forward flights	
iFlight 5″ 6S FPV	0.24
DJI Phantom 4	0.41
DJI Mavic 3	0.45
DJI Inspire 1	0.63
DJI Matrice 300	1.05
Model	Wing span (m)
Fixed wings during forward flights	
Parrot Disco	1.15
Atmos Marlyn	1.70
EOS-C VTOL	1.80
Custom-made tilting-rotor [63]	1.85
Believer Drone	1.96

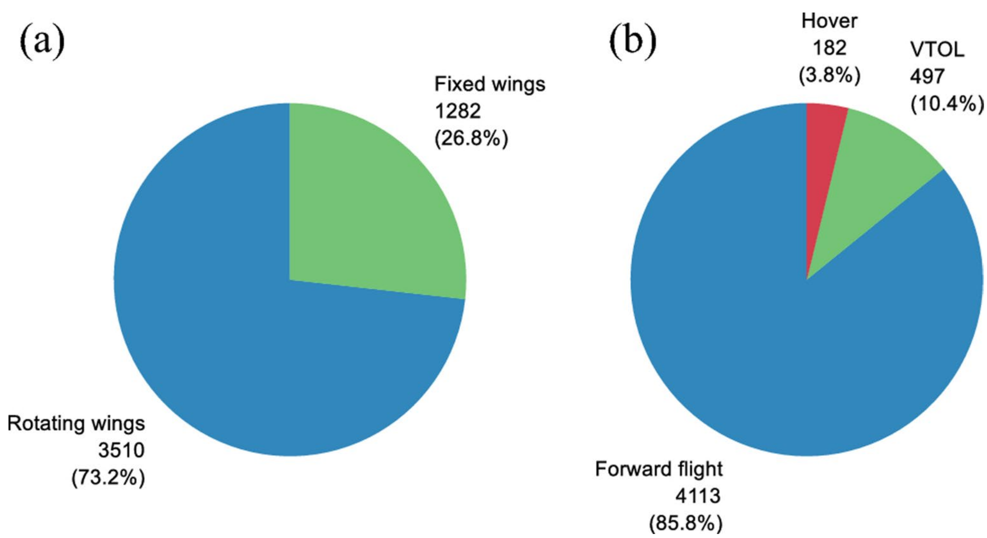


Fig. 3 Composition of the drone configurations and flight phases in our dataset. (**a**) Number of frames (percentage) and their corresponding drone configurations during forward flights. (**b**) Number of frames (percentage) and their corresponding flight phases. VTOL stands for Vertical Take Off and Landing

snippets with overlaying bounding-box labels for verification. Several examples of the bounding-box labeled drone and bird snippets and their trajectories in our dataset are visualized in Fig. 4. The bounding boxes from the first and the last frames are shown while the dots represent the bounding box centroids along the trajectory. The movement direction is indicated by the arrows, and the background is taken from the first frame of each snippet.

It is important to acknowledge the limitations of our dataset. First, due to the diversity of drone operators, we could not establish a precise one-to-one mapping between each frame and the drone's GPS location. As a result, the camera-to-drone distance (range) is estimated based on the drone's physical size, its apparent size in the frame, and the camera specifications (see Sect. 5.1.4). Second, all recordings were made with a static camera at airbases. For future enrichment of the dataset, we plan to include moving-camera scenarios, provide pan-and-tilt kinematic information, and expand filming locations to cover critical infrastructure such as ports and urban areas. Third, as the birds naturally appeared in our scenes, species-level annotation was not performed. Hence, their diversity was not quantified.

Despite these limitations, the dataset still provides a relatively diverse set of drone models and a balanced representation of drones and birds. The two filming locations, one close to the North Sea and one inland, recorded at different times of the year, should provide natural variation in bird species. We consider the current drone and bird variations in our dataset sufficient for the present study.

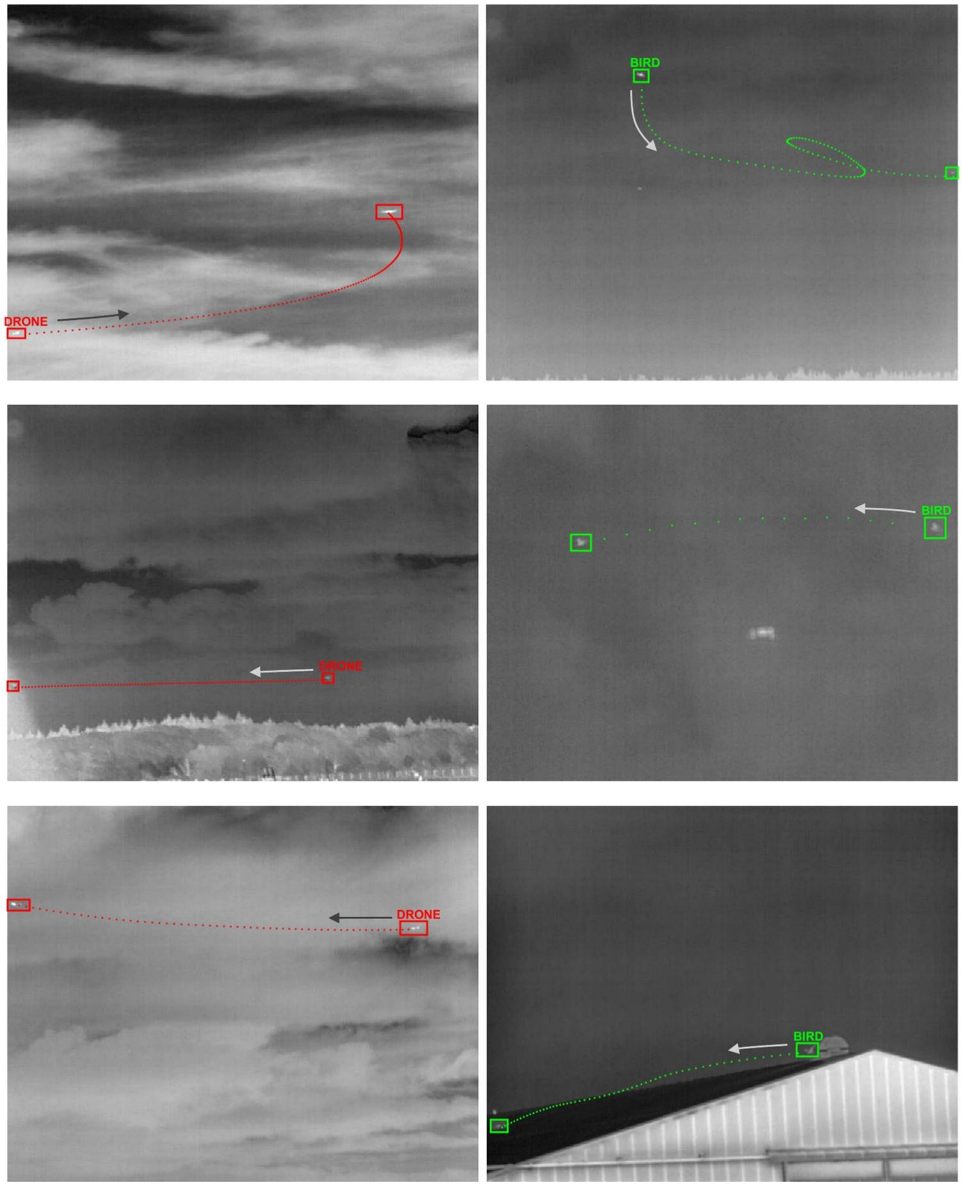


Fig. 4 Example drone (left column) and bird (right column) snippets from our dataset. The bounding boxes from the first and last frames are shown while the dots represent the bounding box centroids along the trajectory. The arrows indicate the movement direction. The background is taken from the first frame of each snippet

4.2 Data partitioning and training

We performed five data stratifications. In each, 40 (20 birds and 20 drones) out of the 200 trajectories were reserved for testing. The idea is that every trajectory is used once as a test trajectory. Having kept the test trajectories out in each stratification, the frames from the remaining trajectories were shuffled and used for training (70%) and validation (30%).

We trained the classifiers using a mini batch size of 100 for 5 epochs. We also investigated other choices of batch size, namely 50, 200, and 400. We found that the average accuracy of the trajectory-feature classifier drops by approximately 10% when we use the batch size of 400. For the other batch sizes, the accuracy change is negligible. Depending on the varying total trajectory lengths (total number of frames) in the training data, we performed approximately 350 to 400 iterations for each training. We validated our training process every 50 iterations using the pre-partitioned validation dataset. We used the adaptive moment estimator (Adam) optimizer with a constant learn rate of 0.001 to minimize the cross-entropy loss function. We found minimal changes in our results for the learning rates of 0.01 and 0.1. However, a low learning rate of 0.0001 drastically reduces the accuracy of the trajectory-feature classifier.

We employed the following cross-entropy loss function, L:

$$L = -\frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K}[\mathcal{X}(k) == Y_m]\ln s_{k,m}, \tag{8}$$

where $s_{k,m} \in [0, 1]$ is an element in the k-th row m-th column of a matrix s containing the softmax function outputs. The matrix s has a dimension of $K \times M$, where K is the number of classes and M is the mini batch size. In our case, $K = 2$ and $M = 100$. The function $\mathcal{X}(k)$ maps the class index k to the class names x. We define $\mathcal{X}(1) =$ Bird and $\mathcal{X}(2) =$ Drone. Finally, $Y_m \in \{\text{Drone}, \text{Bird}\}$ is the ground-truth class of the m-th sample in the mini batch.

4.3 Performance evaluation metrics

4.3.1 Frame-wise accuracy

For a given frame i_j in trajectory j, the output of these classifiers are the predicted object class $x_{i,j} \in \{\text{Drone}, \text{Bird}\}$ and the softmax function output $s_{i,j} \in (0.5, 1]$. We evaluated the classifier performance based on the accuracy, i.e., the number of correct predictions divided by the number of all predictions. We first evaluate the so-called frame-wise accuracy expressed as follows:

$$\text{Frame-wise accuracy} = \frac{\sum_{j=1}^{J}\sum_{i=1}^{I_j}(x_{i,j} == Y_j)}{\sum_{j=1}^{J}\sum_{i=1}^{I_j}(x_{i,j} == Y_j \text{ or } x_{i,j} \mathrel{!=} Y_j)}, \tag{9}$$

where J is the total number of trajectories, I_j is the length (number of frames) of trajectory j, and $Y_j \in \{\text{Drone}, \text{Bird}\}$ is the ground-truth class for trajectory j.

4.3.2 Trajectory-wise accuracy

To imitate real-life usage of the classifiers, in which each suspicious moving object is treated individually, we compute the trajectory-wise accuracy as follows:

$$\text{Trajectory-wise accuracy} = \frac{\sum_{j=1}^{J}(X_j == Y_j)}{\sum_{j=1}^{J}(X_j == Y_j \text{ or } X_j \mathrel{!=} Y_j)}, \tag{10}$$

where $X_j \in \{\text{Drone}, \text{Bird}\}$ is the object class prediction for trajectory j. Because each trajectory contains multiple frames and the class prediction per frame $x_{i,j}$ may vary, the trajectory class prediction X_j is obtained by frame-wise weighted voting, based on the frame-wise prediction $x_{i,j}$ and the softmax function output $s_{i,j}$, i.e.,

$$X_j = \begin{cases} \text{Drone}, & \text{for } \left[\sum_{i=1}^{I_j} s_{i,j}(x_{i,j} == \text{Drone}) \Big/ \sum_{i=1}^{I_j} s_{i,j} \right] > 0.5 \\ \text{Bird}, & \text{for } \left[\sum_{i=1}^{I_j} s_{i,j}(x_{i,j} == \text{Bird}) \Big/ \sum_{i=1}^{I_j} s_{i,j} \right] > 0.5 \end{cases}. \quad (11)$$

In other words, the class of which the summed voting score is more than half of the total voting score, considering all of the predictions in the trajectory, is concluded as the class for this trajectory.

4.4 Use of the Svanström et al. dataset

Svanström et al. have provided an open-access multi-sensor database for drone-detection research containing, among others, ground-truth-labeled video and audio files of drones and other flying objects. Extensive specifications of the dataset and its acquisition are provided in Ref. [54]. Here, we consider the ground-truth-labeled infrared-camera videos of drones and birds. It is noteworthy that the drones in this dataset are only quadrotor drones which were flown manually. These videos are grouped by the object apparent sizes resulting from the varying camera-to-target distances; namely, close, medium, and distant. Here, we consider a subset of the videos in the medium and distant groups, where the camera-to-target distances are larger than 20 m and the apparent object sizes are smaller than 15 pixels. We manually down-selected videos containing only one object and no camera movement. This resulted in 51 video snippets for birds and 51 video snippets for drones. To summarize, Table 2 provides the specifications of the down-selected Svanström et al. dataset and our dataset. We followed the same methodologies as indicated above for the trajectory construction, feature calculation, classifier structures, training, and testing. We performed 17 training rounds. In each of which, 6 trajectories were used for testing the classifiers.

5 Results and discussion

We begin by characterizing the trajectory and visual appearance information of the drones and the birds in our dataset. We also compare our dataset to the Svanström et al. dataset. The dataset characterization and comparison are presented in Sect. 5.1. Next, we discuss the trajectory feature comparisons between the drones and the birds in Sect. 5.2. We compare the performance of the classifiers in Sect. 5.3. Finally, we present our ablation and runtime study in Sect. 5.4.

Table 2 Summary of the datasets used in our study. Note that we only include a subset of the Svanström et al. dataset that is relevant to our study

Specifications	Our dataset	Down-selected Svanström et al. dataset
Infrared camera model	FLIR Boson+	FLIR Breach PTQ136
Pixel resolution (horizontal × vertical)	640 × 512	320 × 256
Horizontal field of view (degrees)	5.5	24
Frame rate (Hz)	10	30
Number of trajectories (drone + bird)	100 + 100	51 + 51
Number of frames (drone + bird)	4792 + 4226	6018 + 6018
Average snippet length (seconds)	4	4
Drone configurations	Fixed wings and rotating wings	Rotating wings
Drone control method	Manual and autopilot	Manual
Approximate maximum target distance (m)	1000	200

5.1 Dataset characterization

5.1.1 Our dataset

The coordinate points and trajectories of the drones and the birds in our dataset are shown in Figs. 5a and b, respectively. From these figures, we can already notice collective visual differences between the drone and bird trajectories. The drone trajectories mostly appear as straight lines, while those of the birds are more tortuous. This can be attributed to the drones' ability to fly along a predefined path in a highly controlled manner using an autopilot. Our observation is in line with a similar study by Srigrarom et al. [35], who observed that drone flight paths change the heading angle less frequently, and have a lower curvature, than that of the birds.

In Fig. 6a and b, the snapshots of the drones and the birds in our dataset are shown. The snapshots are taken from the first frame of each snippet. They are cropped by the bounding box and resized to 28 $\times$ 28 pixels. The cropped snapshots are ranked based on the ratio B averaged for the entire snippet, from low to high. As B increases, we see an increase of the object visual clarity and details. For the drones, the rotors, fuselage, and wings (depending on the configuration) become clearer. For the birds, the body and wings can be seen. However, when B is relatively low, it becomes visually challenging to discern the drones and the birds. This figure therefore shows the wide variation of the object visual clarity in our dataset.

5.1.2 Svanström et al. dataset

The trajectories of the drones and the birds in the dataset of Svanström et al. are visualized in Fig. 7a and b, respectively. Generally, we observe more tortuous trajectories for the drones than in our dataset. This could be because the dataset of Svanström contains only quadrotor drones which are relatively more maneuverable than the fixed-wing drones present in our dataset. The drones in their dataset were flown manually, and strictly within the pilot's visual line of sight, instead of using an autopilot to follow a predefined flight path. The manual control of the drones is evident from the trajectories shown in Fig. 7a, and has been confirmed through personal communication with the dataset publishers.

Figure 8 shows bounding-box cropped snapshots of the drones and the birds in the Svanström et al. dataset from the first frame of each trajectory snippet. In the Svanström et al. dataset, we see less visual appearance variation compared to our dataset discussed in Fig. 6. This is due to the fewer drone models (only quadrotors) in their

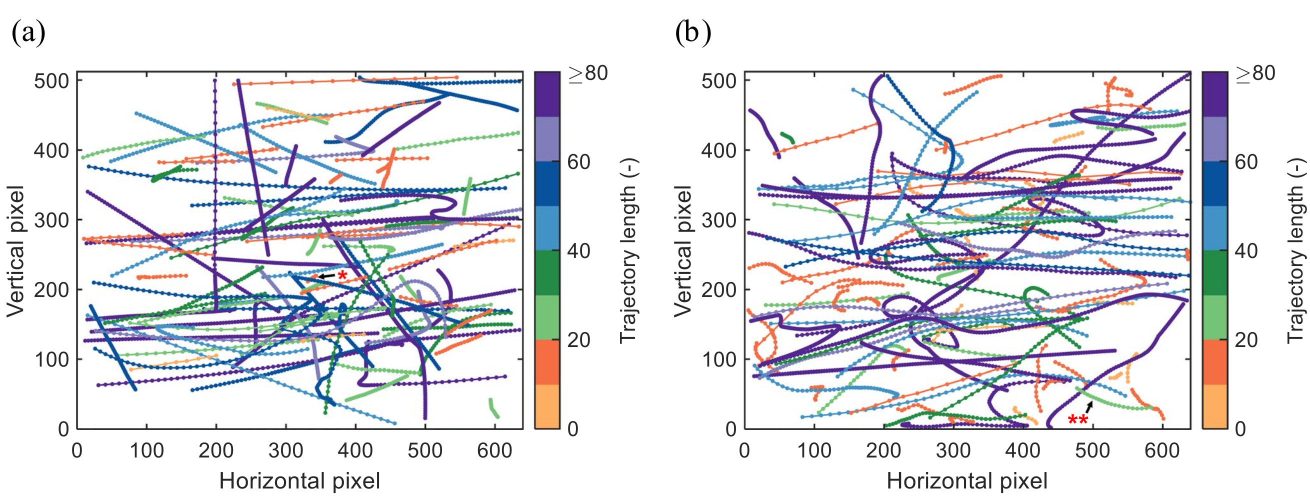


Fig. 5 Constructed trajectories and their lengths for (**a**) the drones and (**b**) and the birds in our dataset. The annotated * and ** indicate the example failed classification cases discussed in Sect. 5.3.2

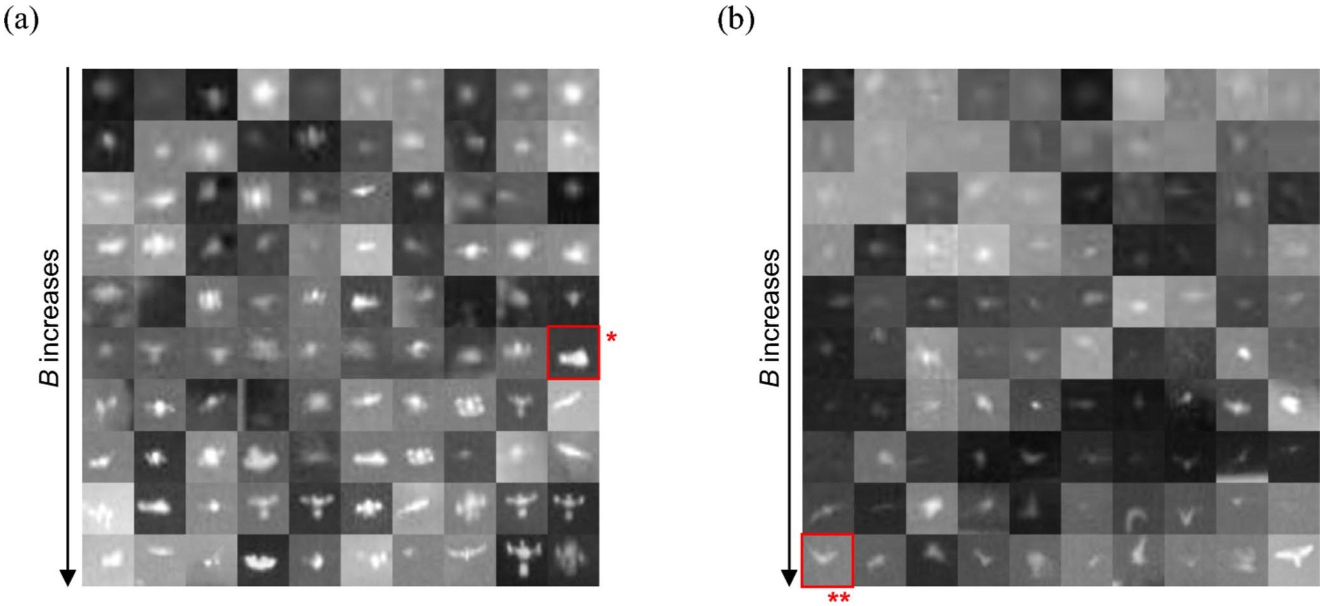


Fig. 6 Bounding-box cropped and resized images of (**a**) the drones and (**b**) the birds in our dataset. The snapshots are from the first frame of each trajectory, ranked based on the trajectory-averaged ratio B. The annotated * and ** indicate the example failed classification cases discussed in Sect. 5.3.2

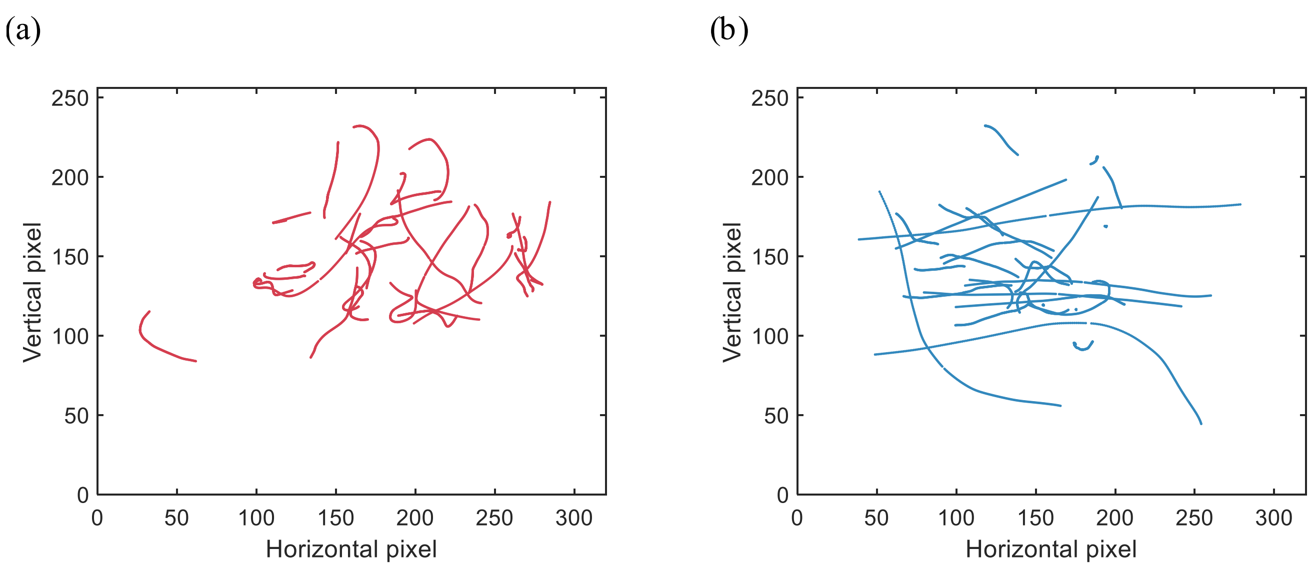


Fig. 7 Constructed trajectories for (**a**) the drones and (**b**) and the birds in the Svanström et al. dataset

dataset. Apart from that, in average, the apparent object sizes in the Svanström et al. dataset is larger than in our dataset. This is discussed in the following subsection.

5.1.3 Trajectory and visual information in the datasets

Here, we compare the trajectory and visual appearance information distribution of the Svanström et al. data with our data. We quantify these information types based on the trajectory length and the ratio B introduced in Sect. 3.4.

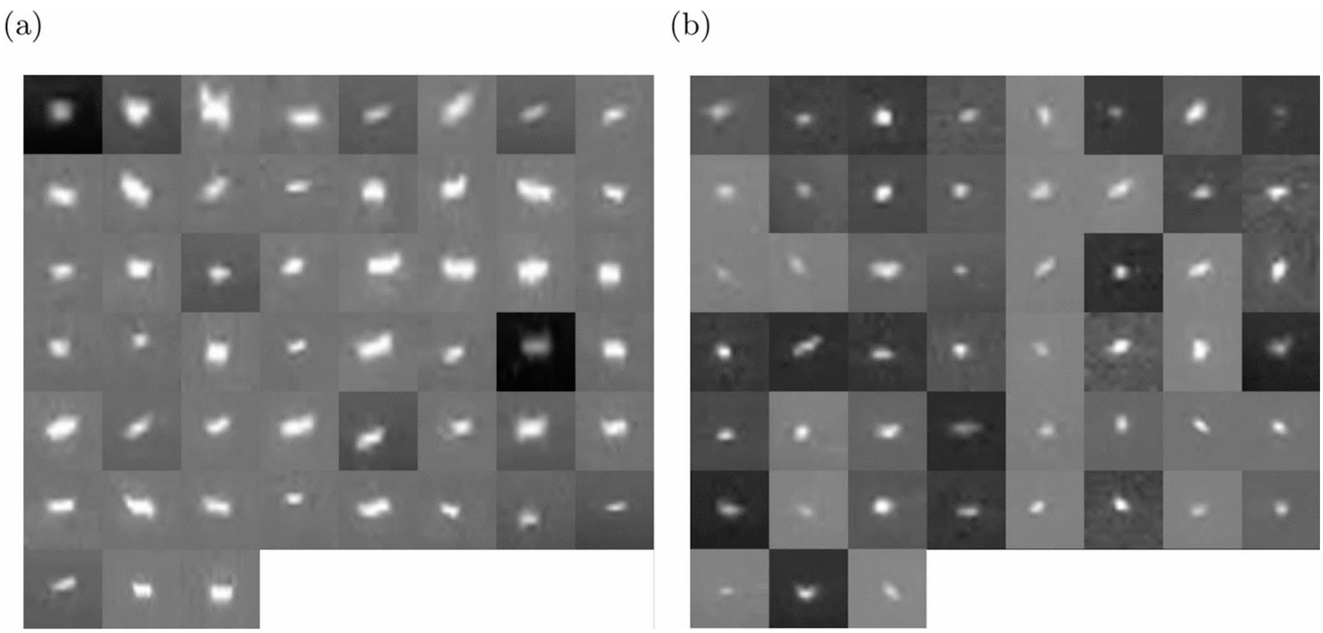


Fig. 8 Bounding-box cropped and resized images in the Svanström et al. dataset. (**a**) Drones and (**b**) birds in the first frames of each trajectory

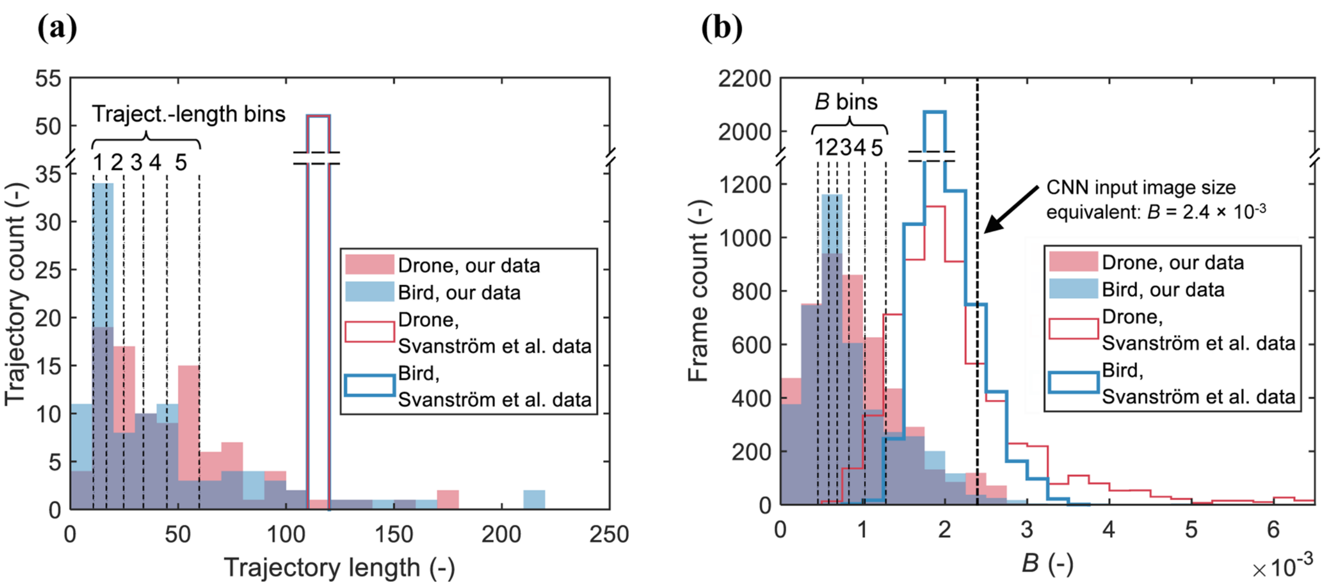


Fig. 9 (**a**) Trajectory length distributions in our dataset compared to that in the Svanström et al. dataset, and the trajectory-length bin subdivisions. (**b**) Ratio B distributions in our dataset compared to that in Svanström et al. dataset, and the ratio-B bin subdivisions

Figure 9a compares the trajectory-length distributions of the drones and the birds in our dataset. Our dataset contains varying trajectory lengths, unlike the constant trajectory length in the Svanström et al. dataset. The trajectory length values in our dataset are, in average, lower than that of Svanström et al. because of our lower frame rate. However, accounting for the frame rate difference, both our dataset and the Svanström et al. dataset observe the object's trajectory for the same average time of approximately 4 seconds.

The ratio *B* distributions of our dataset and the Svanström et al. dataset are shown in Fig. 9b. The distributions indicate that the object bounding boxes in the Svanström et al. dataset appear slightly larger than in our dataset, and are therefore likely to provide, in average, clearer object visual information, compared to our data. This confirms our previous discussion of Fig. 8. Based on the visual appearance information comparison, one would expect a classifier based only on the visual appearance feature achieve a better classification accuracy with the Svanström et al. dataset than with our dataset.

In Fig. 9b, we also indicate the ratio B equivalent to the 28 $\times$ 28 pixel input size of the visual-appearance-feature classifier. We show that most of the input images in our dataset are much smaller than the CNN input size. Therefore, we are not likely to lose visual appearance information due to the input image resizing.

Finally, for further analysis, we assign five bins to the trajectory lengths and the ratio B in our dataset. These bins are marked by the dashed lines in Figs. 9a and 9b. The bins are assigned such that each contains approximately the same number of data points (frames). These bins are used to study the classification accuracy variability with the amount of trajectory and visual appearance information, i.e., the trajectory length and B.

5.1.4 Range estimation from the ratio *B*

In the previous sections, we introduce the ratio B to quantify the amount of visual appearance information for our classifier. We further subdivided this into bins as shown in Fig. 9b for our further analysis. Here, we approximate the distance from the camera to the drone, i.e., the range, in our dataset. This is done to provide a more direct connection between the B bins and the range. Although we cannot match every video frame with the GPS location due to the high variety of drone operators during our measurements, we can still make use of our camera specifications, drone size, and their apparent sizes to approximate the range as follows:

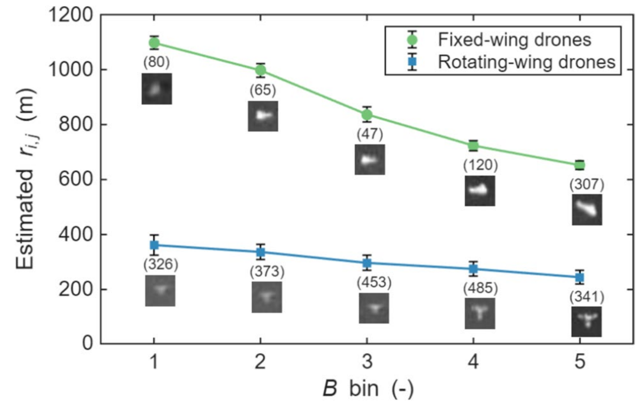
$$r_{i,j} = \frac{\bar{\ell} P_{\mathrm{cam}}}{2\varphi P_{i,j} \tan\left(\mathrm{HFOV}/2\right)}, \tag{12}$$

where $r_{i,j}$ is the estimated range, $\bar{\ell}$ is the averaged characteristic length of the drone taken from Table 1, P_{cam} is the horizontal pixel resolution of the camera (here, 640 pixels), $P_{i,j}$ is the bounding box width in pixels, φ is a multiplication factor taken as 0.7 to compensate for the fact that the bounding box is slightly wider than the drone, and HFOV stands for the camera's Horizontal Field Of View (here, 5.5 degrees).

The approximated $r_{i,j}$ is shown in Fig. 10. The numbers in brackets are the numbers of frames that fall within that B bin. When the B bin number increases, the range $r_{i,j}$ decreases. Here, we separate the approximations for the rotating-wing drones and fixed-wing drones because $\bar{\ell}$ for the fixed-wing drones is much larger than that of the rotating-wing drones (see Table 1). We found that one step in the B bin number corresponds to the distance difference of approximately 25 to 30 m for rotating-wing drones, and 100 m fixed-wing drones. The approximated range for the fixed-wing drones is larger than that of the rotating-wing drones. We estimate the range in our dataset up to approximately 1000 m, which is in line with the available GPS tracks.

In Fig. 10, we also provide bounding-box cropped example images of the drones per B bin, for both the fixed-wing and rotating-wing configurations. These cropped images are resized to 28 $\times$ 28 pixels, matching the input size of the visual-appearance-feature classifier. For the larger estimated distances $r_{i,j}$ (lower B bins), the drone appears as a blurred dot. It can be seen that as $r_{i,j}$ reduces (higher B bins), there is more visual clarity in the drone image. This demonstrates the visual appearance information variation with the ratio B and the estimated $r_{i,j}$ in our dataset.

Fig. 10 Range $r_{i,j}$ approximations for the fixed-wing and rotating-wing drones with respect to the ratio B in our dataset. The numbers in brackets are the numbers of frames that fall within that B bin. Example bounding-box cropped and resized frames of the drones per B bin are also shown



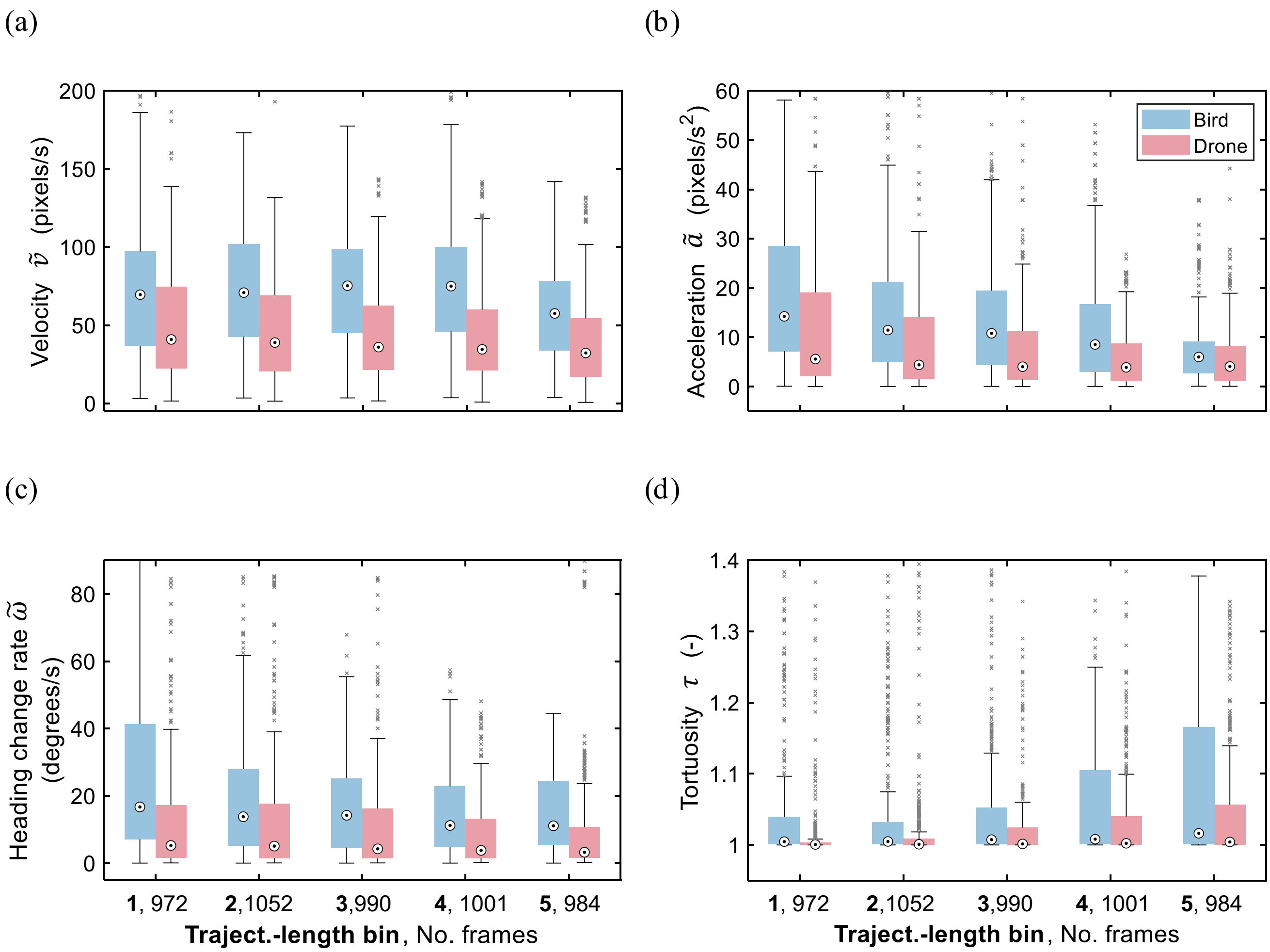


Fig. 11 Comparison between drone and bird trajectory features in our dataset with respect to the trajectory-length bins. The considered features are: (**a**) The velocity, (**b**) The acceleration, (**c**) The heading change rate, and (**d**) The tortuosity

5.2 Drone vs. bird trajectory features

5.2.1 Our dataset

In Fig. 11, we visualize the extracted trajectory features of the drones and the birds with respect to the trajectory-length bins indicated in Fig. 9a. From Fig. 11a, we see that, in our dataset, the birds have larger velocities than the drones. This result is consistent with the finding of Han et al. [34]. The explanation for this is twofold. First, most drones have the ability to hover and fly at a controlled constant (low) speed. Second, the drones are, in average, larger than the birds, and they can therefore be further away from the camera to be as visible as the birds. With this condition, even when they travel at the same absolute velocity, the apparent velocity for the drones perceived by the camera is lower. Next, Fig. 11b and c show that the birds have larger accelerations and heading change rates than the drones, also consistent with Han et al. [34]. Interestingly, the drones' median acceleration is close to 1 pixel/s^2 and their median heading change rate is close to 0 degree/s. This could be because some of the drones were programmed to fly at a constant velocity in a straight line. Consequently, the tortuosity of the birds is larger than that of the drones as shown in Fig. 11d.

The trajectory feature differences between the drones and the birds are represented by how much the drone and the bird box plots overlap. Considering the differences with respect to the trajectory length bins in Fig. 11, we observe that the difference grows larger when the trajectory length increases. This is clear especially for the trajectory-length bin number 4. We also observe that the tortuosity for both the birds and the drones increases when the trajectory length increases. The visible trajectory feature difference between the drones and the birds in our dataset confirms that these features are valuable information to consider in addition to the visual appearance features when differentiating the drones from the birds.

5.2.2 Svanström et al. dataset

Figure 12 presents the trajectory feature comparisons of the dataset of Svanström et al. [54]. Unlike in our dataset, the acceleration, heading change rate, and tortuosity of the drones are larger than those of the birds. The reason for this is due to the inherent characteristics of their dataset discussed in Sect. 5.1.2. Nevertheless, we still observe the trajectory feature difference between the drones and the birds, shown by the box plots.

5.3 Classifier performance

5.3.1 Frame-wise accuracy: our dataset

We first investigate the frame-wise accuracy provided by the considered classifiers. We study the frame-wise accuracy variation with the amount of visual appearance information by evaluating Eq. (9) per B bin. In other words, given a set $\mathbf{B}_n$ containing all $B_{i,j}$ belonging to bin n, we evaluate Eq. (9) for all $x_{i,j}$ where $B_{i,j} \in \mathbf{B}_n$. We perform this analysis only for our dataset. Results are shown in Fig. 13a. For all of the classifiers, we see an increasing trend of the frame-wise accuracy as B increases. The visual-appearance-feature classifier clearly achieves higher accuracies than the trajectory-feature classifier. This is reasonable because, for a given frame, the visual appearance feature contains much more information than the trajectory features. This indicates the indispensable role of the visual appearance information in the classification of flying objects. For the lower B bins, we see a drop in the accuracy provided by the visual-appearance-feature classifier. The sensitivity of the visual-appearance-feature classifier accuracy to B is explainable by the more limited amount of visual information, and has been reported earlier in literature [25]. Interestingly, in this area, especially for bins 3 and 4 where the accuracy of the trajectory-feature classifier is greater than 0.6, the classifier which combines the visual and trajectory information achieves higher frame-wise accuracies than the visual-appearance-feature classifier alone. This demonstrates the added value of the trajectory information when the visual information is limited.

In the same manner, we examine the frame-wise accuracy variation with the amount of trajectory information, represented by the trajectory length, in Fig. 13b. We observe more constant frame-wise accuracies for all of the classifiers compared to Fig. 13a. The trajectory-feature classifier provides an approximately constant frame-wise accuracy of 0.7, close to that of the visual-appearance-feature classifier. The relatively higher and more stable frame-wise accuracy of the trajectory-feature classifier can be attributed to the difference of the trajectory features

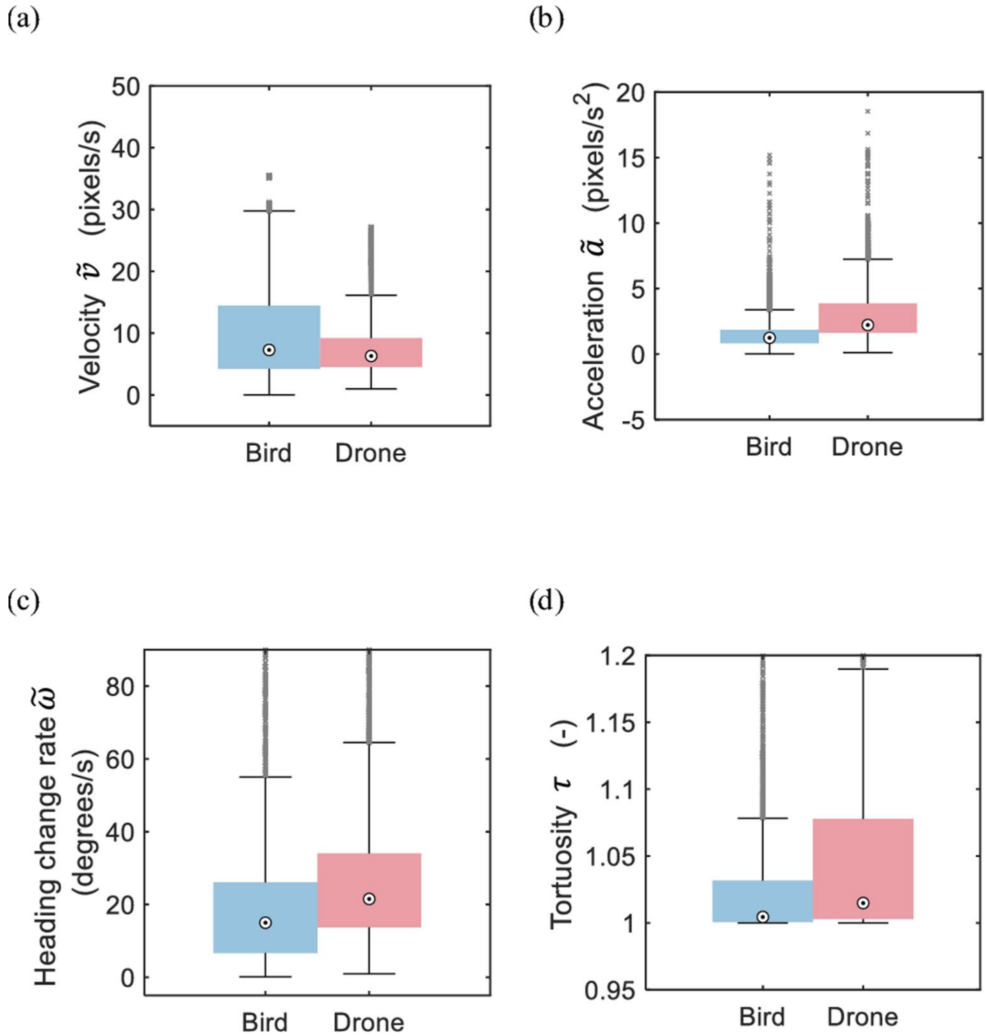


Fig. 12 Comparison between drone and bird trajectory features in the Svanström et al. dataset. The considered features are: (**a**) The velocity, (**b**) The acceleration, (**c**) The heading change rate, and (**d**) The tortuosity

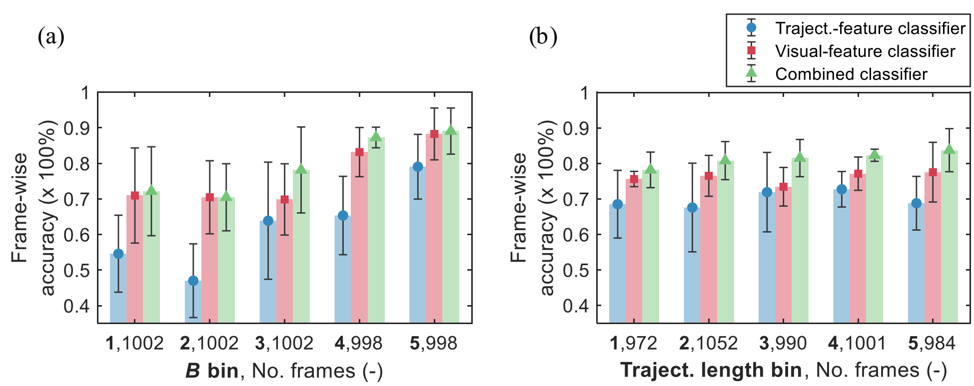


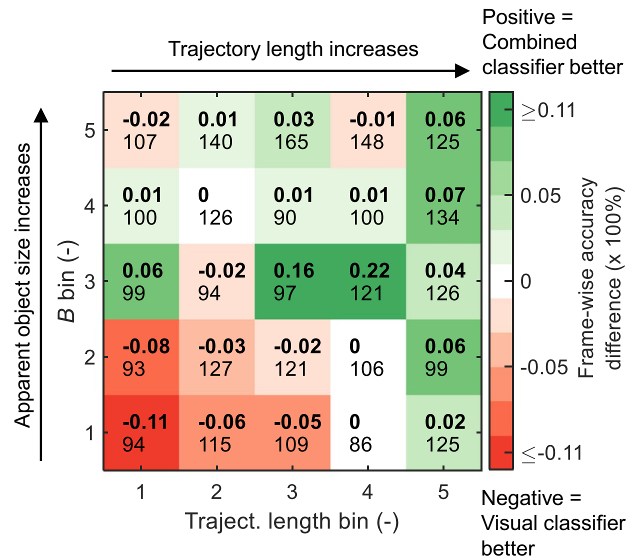
Fig. 13 Frame-wise drone and bird classification accuracies of the considered classifiers. (**a**) Frame-wise accuracies with respect to the B bins. (**b**) Frame-wise accuracies with respect to the trajectory-length bins

discussed in Fig. 11. When combining the trajectory with visual appearance features, the highest frame-wise accuracy is achieved.

Next, we identify regions where the combined classifier improves the frame-wise accuracy, compared to the visual-appearance-feature classifier. We do this by examining the frame-wise accuracy difference (combined classifier minus visual-appearance-feature classifier). The result is shown in Fig. 14, where a positive value means the combined classifier has a better frame-wise accuracy than the visual-appearance-feature classifier, and

 https://doi.org/10.1007/s00521-026-12080-5

Fig. 14 Accuracy differences (in **bold**) between the combined visual-trajectory features classifier and the visual-alone classifier with respect to the B and trajectory-length bins. The numbers below the bold accuracy differences indicate the number of data points (frames) considered



a negative value means the opposite. We investigate the accuracy difference for all possible combinations of the B and trajectory-length bins. One can infer that the bottom-left corner of this plot, where B and the trajectory length are low, is the most challenging region for the classifiers because both the visual appearance and trajectory information are relatively limited. While progressing diagonally towards the top-right corner, the amount of both the trajectory and visual appearance information increases, making the top-right corner the least challenging region for the classifiers.

From Fig. 14, we observe that the longer the trajectory, meaning the more the trajectory information, the more likely the combined classifier will outperform the visual-appearance-feature classifier. We also identify an area where the trajectory information is relatively high (trajectory-length bins 3 and 4) and visual information is relatively low (B bin 3). In this area, up to 22% increase in the frame-wise accuracy is achieved. This result suggests that when the trajectory information is sufficiently high, including the trajectory features can help improve the classifier performance when the apparent object size is small. Based on our range estimation in Fig. 10, one step of B bin corresponds to the range step of approximately 25 m for rotating-wing drones and 100 m for fixed-wing drones. This implies that the inclusion of the trajectory features can ultimately help extend the operational sensor-to-target distance of aerial surveillance cameras.

5.3.2 Trajectory-wise accuracy: our dataset and Svanström et al. dataset

We summarize the trajectory-wise accuracy as well as frame-wise accuracy in Table 3. This is done for all of the considered classifiers applied to both our dataset and the Svanström et al. dataset. The trajectory-wise accuracy is calculated as described in Sect. 4.3.2.

For our dataset, we show that the combined classifier increases the total number of correctly predicted frames by 5%, compared to the visual-appearance-feature classifier. This also increases the number of correctly predicted trajectories by 4% and thereby makes the combined classifier achieve the highest trajectory-wise accuracy. We further investigated specific cases where the visual-appearance classifier fails, while the trajectory-feature and combined classifiers correctly classify the object class. Examples include the fixed-wing drone marked by * and the bird marked by ** in Fig. 6. Their corresponding trajectories are indicated by the same symbols in Fig. 5. We observed that correct classifications by the combined classifier, despite misclassifications by the visual-appearance classifier, often occur when the drone moves at very low apparent velocity or hovers (see the trajectory marked by * in Fig. 5a). Similar improvements occur when the bird exhibits acceleration while changing

Table 3 Frame-wise and trajectory-wise drone and bird classification accuracies provided by the considered classifiers applied to our dataset and the Svanström et al. dataset. The highest accuracies are highlighted in **bold**

Our data				
Classifiers	No. correctly predicted frames (Out of 9018)	Frame-wise accuracy (×100%)	No. correctly predicted trajects. (Out of 200)	Traject.-wise accuracy (×100%)
Traject.	6115	0.68 ± 0.06	141	0.70 ± 0.10
Visual	6846	0.76 ± 0.03	159	0.80 ± 0.02
Combined	**7265**	**0.81 ± 0.03**	**168**	**0.84 ± 0.05**
Svanström et al. data				
Classifiers	No. correctly predicted frames (Out of 12036)	Frame-wise accuracy (×100%)	No. correctly predicted trajects. (Out of 102)	Traject.-wise accuracy (×100%)
Traject.	8227	0.68 ± 0.14	72	0.71 ± 0.20
Visual	11479	0.95 ± 0.05	99	0.97 ± 0.07
Combined	**11633**	**0.97 ± 0.04**	**100**	**0.98 ± 0.06**

its heading (see the trajectory marked by ** in Fig. 5b). Hence, these motion patterns appear to provide strong supporting discriminative cues, for challenging visual-appearance classifications.

Finally, we apply our methodology to the open dataset of Svanström et al. The resulting frame-wise and trajectory-wise accuracies are presented in the lower half of Table 3. Using this dataset, the visual-appearance-feature classifier already achieves excellent frame-wise accuracy of 95% and trajectory-wise accuracy of 97%. This can be explained by the relatively higher amount of visual information in this dataset, compared to our dataset, as discussed in Fig. 9b. Moreover, all of the drones in this dataset are quadrotor drones. They therefore have relatively less visual variability, compared to the drones in our dataset. Nonetheless, we still see a slight improvement in the frame-wise and trajectory-wise accuracies when using the combined classifier, compared to the visual-appearance-feature classifier. This confirms the generalizability of our finding.

5.4 Ablation and runtime study

5.4.1 Trajectory feature ablation

We have demonstrated that combining the visual-appearance and trajectory features is beneficial in the classification of drones and birds in aerial surveillance. In this ablation study, we investigate the contribution of each trajectory feature in more detail.

Figure 15 shows the accuracies of the trajectory-feature classifier and combined classifier when different combinations of the trajectory features are used. The combination presented so far where all features are considered is named as "Baseline". Then we use numbers to represent the different variations of feature combinations, which are explained in the following:

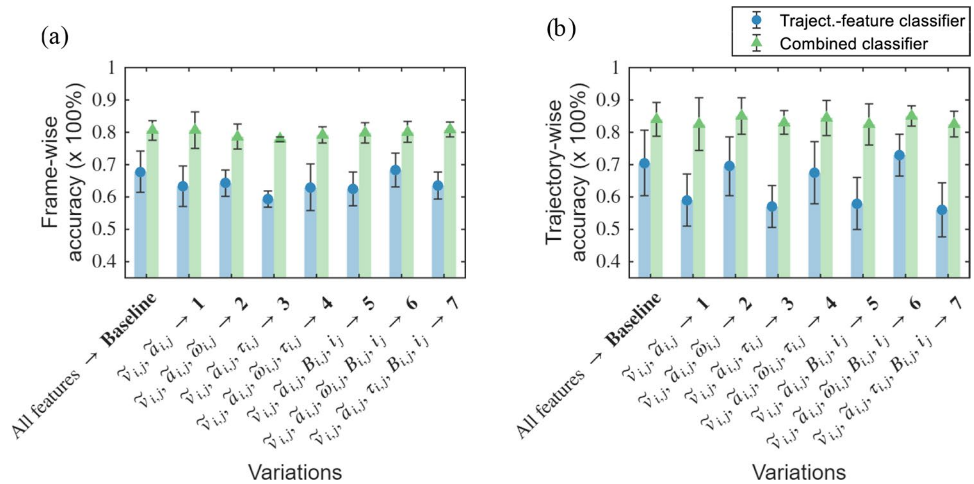
- Variation 1: Only the velocity $\tilde{v}_{i,j}$ and acceleration $\tilde{a}_{i,j}$. (The most minimalistic variation)
- Variations 2, 3, and 4: Same as variation 1, with the added heading change rate $\tilde{\omega}_{i,j}$, tortuosity $\tau_{i,j}$, and both $\tilde{\omega}_{i,j}$ and $\tau_{i,j}$, respectively.
- Variations 5, 6, and 7: Same as variations 2, 3, and 4, with the added ratio $B_{i,j}$ and trajectory length i_j.

We show both the frame-wise and trajectory-wise accuracies in Figs. 15a and b, respectively.

For the frame-wise accuracy in Fig. 15a, most feature reductions with respect to the baseline reduces the accuracy of the trajectory-feature classifier. The clearest drop in the accuracy is found for variation 3, where only the velocity, acceleration, and tortuosity are used. Inclusion of the bounding box size ratio B and trajectory length helps to improve the accuracy (see variation 3 vs 7). This suggests that the tortuosity is meaningful when the

 https://doi.org/10.1007/s00521-026-12080-5

Fig. 15 Ablation study: (**a**) frame-wise and (**b**) trajectory-wise accuracies of the trajectory-feature and combined classifiers when different trajectory feature combinations are used. The combination employed in the rest of our study is named "Baseline" while the other combinations are represented by variation numbers shown in **bold**



bounding box size and trajectory length is available, and should not be used alone. The same observation also holds for the heading change rate (see variation 2 vs 6). Interestingly, for variation 6 where only the tortuosity is excluded, the resulting accuracy is comparable to the baseline.

Figure 15b shows trajectory-wise accuracies for the same feature ablation variations. In this figure, we notice a clear contribution of the heading change rate to the accuracy of the trajectory-feature classifier. This can be seen in variations 2, 4, and 6 which mutually include the heading change rate. A clear improvement of the trajectory-feature classifier can be seen compared to when the heading change rate is absent in variations 1, 3, 5, and 7.

For both the frame-wise and trajectory-wise accuracies, we see that the combined classifier's accuracy is slightly influenced by the trajectory-feature classifier's accuracy. The difference in the accuracies among the different cases is however less pronounced.

Based on this ablation study, we recommend that the velocity, acceleration, and heading change rate should be prioritized if the computational resource is limited. Furthermore, if the tortuosity is used, information on the trajectory length and the bounding box size should also be provided for a complete context.

5.4.2 Runtime vs CNN complexity

We study the consequence of replacing our proposed CNN-based visual-appearance-feature classifier by generally known CNN architecture, namely, the MobileNetV2 [44] and ResNet-18 [60], on the classification accuracies. Because our input images are gray-scale 28 × 28 pixels images, smaller than the default input size of 224 × 224 pixels for the MobileNetV2 and ResNet-18, we modified these networks by adjusting the input size and removing the deeper layers of these networks. We used the first 3 residual bottleneck layers of the MobileNetV2, and up to the end of ResNet-18 conv2_x layer for feature extraction.

Changing the CNN architecture also impacts the runtime of our classification process, which is critical for real-time deployment for aerial surveillance. We therefore address the impact of varying the CNN architecture on both the accuracy and runtime altogether, such that a trade-off can be made.

We performed this study using our dataset only. The frame-wise and trajectory-wise accuracies for both the visual-appearance-feature and combined classifiers given by the different CNN architectures are shown in Table 4. Note that the results for "Our CNN" is repeated from Table 3. The corresponding classification runtime per frame are also shown. Additionally, the numbers of learnable parameters for the visual-appearance-feature classifier based on each CNN architecture are provided. All of our experiments were done on MATLAB R2023a with the Deep Learning Toolbox. We used an Intel Xeon CPU E5-1620 v3 @3.50 GHz and 16 GB RAM.

We see that, despite this adjustment, both the MobileNetV2 and ResNet-18 still have a more complicated architecture compared to our proposed CNN architecture. This is represented by the larger number of learnable

Table 4 Frame-wise and trajectory-wise drone and bird classification accuracies for the visual-appearance-feature and combined classifiers given by the different CNN architectures applied to our dataset. The corresponding classification runtime per frame and the numbers of learnable parameters for the visual-appearance-feature classifier are also provided. The shortest runtime and highest accuracies for the visual-appearance-feature and combined classifiers are highlighted in **bold**

CNN architecture	No. learnables in visual classifier ($\times 1000$)	Classifiers	Frame-wise accuracy ($\times 100\%$)	Trajectory-wise accuracy ($\times 100\%$)	Classification runtime per frame (μs)
Our CNN	25.8	Combined	$\mathbf{0.81 \pm 0.03}$	$\mathbf{0.84 \pm 0.05}$	$\mathbf{268 \pm 16}$
		Visual	0.76 ± 0.03	0.80 ± 0.02	$\mathbf{69 \pm 4}$
MobileNetV2	73.5	Combined	0.78 ± 0.03	0.82 ± 0.05	321 ± 17
		Visual	0.75 ± 0.02	$\mathbf{0.81 \pm 0.03}$	146 ± 11
ResNet-18	151.6	Combined	0.77 ± 0.04	$\mathbf{0.84 \pm 0.08}$	354 ± 16
		Visual	$\mathbf{0.78 \pm 0.02}$	0.80 ± 0.08	178 ± 10

parameters. In general, we found that replacing our CNN by the modified MobileNetV2 and ResNet-18 have minimal effect on the trajectory-wise accuracy. We found that the Resnet-18 slightly improves the frame-wise accuracy of the visual-appearance-feature classifier. However, the implementation of both the MobileNetV2 and ResNet-18 costs more runtime per frame due to their more complicated architectures compared to our CNN. This result suggests that, our lightweight CNN architecture is sufficient for the visual-appearance classification of objects with small apparent pixel sizes. The runtime per frame is minimal compared to typical frame rates of cameras (10 to 60 Hz). This suggests that our proposed classification method could be incorporated in a full detection-free tracking pipeline, with a negligible computational overhead.

6 Conclusion and recommendations

This work tackles the problem of classifying suspicious small flying objects in aerial surveillance videos, with a specific application to differentiating drones from birds. To deal with such a problem, frame-wise extraction and processing of visual appearance features are commonly performed. However, this approach could fail when the apparent object size is small and the visual appearance feature information is limited. We argue that, in addition to the commonly used frame-wise visual appearance features, the trajectory features of the different flying objects, such as the median apparent velocity and acceleration, could contain meaningful cues that help to differentiate these objects. To demonstrate this, we train and test classifiers based on trajectory features, visual appearance features, and both trajectory and visual appearance features combined using our labeled video snippets of drones and birds. We show that combining the trajectory features helps to improve the classification accuracy, compared to when employing the frame-wise visual appearance feature alone. We also identify the cases with relatively low visual information and sufficiently high trajectory information where the largest accuracy improvement of 22% by the combined classifier occurs. This results in a 4% increase of the correctly classified trajectories in our dataset. We have demonstrated that this improvement could be translated to the camera-to-target distance extension of 25 m and 100 m for rotating-wing drones and fixed-wing drones, respectively. This is because the improvement happens in the cases with a relatively small apparent object size.

More generally, our work employs inter-frame temporal information, in this case, the trajectory features, in addition to the frame-wise processing of the visual appearance feature to help enhance the suspicious flying object classification accuracy.

For future extensions of the present dataset and method, we propose to: (1) incorporate camera eigenmotion descriptions to enable trajectory feature calculation for moving-camera scenarios, (2) expand the dataset to include nighttime and urban scenes, and (3) exploit additional inter-frame temporal information, such as changes in object appearance over time.

Acknowledgements The authors would like to thank Fernando Alonso-Fernandez and Fredrik Svanström for answering questions regarding the dataset.

Author contributions S.L.: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Visualization, Writing – original draft; G.C.: Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Writing – review & editing; A.A.: Data curation, Investigation, Project administration, Writing – review & editing; M.S. Funding acquisition, Project administration, Supervision, Writing – review& editing; M.V.: Funding acquisition, Project administration, Supervision, Writing – review & editing.

Funding This work is part of the "ACTION" [ACoustic detecTION of class I (< 25 kg) unmanned aircraft systems supported by optical sensors] project, funded by the Dutch Ministry of Defence under *Nationaal Technologie Project* (NTP) N21/005.

Data availability Data will be made available on reasonable request.

Declarations

Competing interests The authors declare no competing interests.

References

1. Kapustina L, Izakova N, Makovkina E, Khmelkov M (2021) The global drone market: main development trends. SHS Web of Conferences 129:11004. https://doi.org/10.1051/shsconf/202112911004
2. Altawy R, Youssef AM (2016) Security, privacy, and safety aspects of civilian drones: a survey. ACM Trans Cyber-Phys Syst 1(2):1–25. https://doi.org/10.1145/3001836
3. BBC: Gatwick drone shutdown. BBC News. (Accessed: 27 November 2024). https://www.bbc.com/news/topics/cnx1xjxwp51t
4. The Guardian: Danish PM: airport drone incursion a serious attack on critical infrastructure. The Guardian. (Accessed: 16 October 2025). https://www.theguardian.com/world/2025/sep/23/drone-sightings-cause-disruption-delays-norway-denmark-airports
5. Seidaliyeva U, Ilipbayeva L, Taissariyeva K, Smailov N, Matson ET (2023) Advances and challenges in drone detection and classification techniques: a state-of-the-art review. Sensors 24(1):125. https://doi.org/10.3390/s24010125
6. Guvenc I, Koohifar F, Singh S, Sichitiu ML, Matolak D (2018) Detection, tracking, and interdiction for amateur drones. IEEE Commun Mag 56:75–81. https://doi.org/10.1109/MCOM.2018.1700455
7. Semenyuk V, Kurmashev I, Lupidi A, Alyoshin D, Kurmasheva L, Cantelli-Forti A (2025) Advances in uav detection: integrating multi-sensor systems and ai for enhanced accuracy and efficiency. Int J Crit Infrastruct Prot 49:100744. https://doi.org/10.1016/j.ijcip.2025.100744
8. Park S, Kim HT, Lee S, Joo H, Kim H (2021) Survey on anti-drone systems: components, designs, and challenges. IEEE Access 9:42635–42659. https://doi.org/10.1109/ACCESS.2021.3065926
9. Hommes A, Shoykhetbrod A, Noetel D, Stanko S, Laurenzis M, Hengy S, Christnacher F (2016) Detection of acoustic, electro-optical and RADAR signatures of small unmanned aerial vehicles. SPIE. https://doi.org/10.1117/12.2242180
10. Christnacher F, Hengy S, Laurenzis M, Matwyschuk A, Naz P, Schertzer S, Schmitt G (2016) Optical and acoustical UAV detection. Demonstrated benefits of active imaging for UAV detection in complex background. https://doi.org/10.1117/12.2240752
11. Chang X, Yang C, Wu J, Shi X, Shi Z (2018) A surveillance system for drone localization and tracking using acoustic arrays. IEEE. https://doi.org/10.1109/sam.2018.8448409
12. Busset J, Perrodin F, Wellig P, Ott B, Heutschi K, Rühl T, Nussbaumer T (2015). Detection and tracking of drones using advanced acoustic cameras. https://doi.org/10.1117/12.2194309

13. Al-Sa'd MF, Al-Ali A, Mohamed A, Khattab T, Erbad A (2019) RF-based drone detection and identification using deep learning approaches: an initiative towards a large open source drone database. Future Gener Comput Syst 100:86–97. https://doi.org/10.1016/j.future.2019.05.007
14. Sazdić-Jotić B, Pokrajac I, Bajčetić J, Bondžulić B, Obradović D (2022) Single and multiple drones detection and identification using RF based deep learning algorithm. Expert Syst Appl 187:115928. https://doi.org/10.1016/j.eswa.2021.115928
15. Jian M, Lu Z, Chen VC (2018). Drone detection and tracking based on phase-interferometric Doppler radar. https://doi.org/10.1109/RADAR.2018.8378723
16. Liu Y, Wan X, Tang H, Yi J, Cheng Y, Zhang X (2017). Digital television based passive bistatic radar system for drone detection. https://doi.org/10.1109/RADAR.2017.7944443
17. Wisniewski M, Rana ZA, Petrunin I (2022) Drone model classification using convolutional neural network trained on synthetic data. J Imaging 8:218. https://doi.org/10.3390/jimaging8080218
18. Samadzadegan F, Javan FD, Mahini FA, Gholamshahi M (2022) Detection and recognition of drones based on a deep convolutional neural network using visible imagery. Aerospace 9:8. https://doi.org/10.3390/aerospace9010031
19. Elsayed M, Reda M, Mashaly AS, Amein AS (2021) Review on real-time drone detection based on visual band electro-optical (EO) sensor. IEEE. https://doi.org/10.1109/ICICIS52592.2021.9694151
20. Isaac-Medina, B.K.S., Poyser, M., Organisciak, D., Willcocks, C.G., Breckon, T.P., Shum, H.P.H.: Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark (2021) 10.48550/ARXIV.2103.13933
21. 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. IEEE (2017). 10.23919/ICCAS.2017.8204318
22. Delleji T, Slimeni F, Fekih H, Jarray A, Boughanmi W, Kallel A, Chtourou Z (2022) An upgraded-YOLO with object augmentation: mini-UAV detection under low-visibility conditions by improving deep neural networks. Oper Res Forum 3:60. https://doi.org/10.1007/s43069-022-00163-7
23. Trapal DDC, Leong BCC, Ng HW, Zhong JTG, Srigrarom S, Chan TH (2021) Improvement of vision-based drone detection and tracking by removing cluttered background, shadow and water reflection with super resolution. IEEE. https://doi.org/10.1109/ICCRE51898.2021.9435671
24. Ganti SR, Kim Y (2016) Implementation of detection and tracking mechanism for small UAS. IEEE. https://doi.org/10.1109/ICUAS.2016.7502513
25. Svanström F, Alonso-Fernandez F, Englund C (2022) Drone detection and tracking in real-time by fusion of different sensing modalities. Drones 6(11):317. https://doi.org/10.3390/drones6110317
26. Andraši P, Radišić T, Muštra M, Ivošević J (2017) Night-time detection of UAVs using thermal infrared camera. Transp Res Procedia 28:183–190. https://doi.org/10.1016/j.trpro.2017.12.184
27. Shu Y, Sui Y, Zhao S, Cheng Z, Liu W (2021) Small moving object detection and tracking based on event signals. IEEE. https://doi.org/10.1109/ICCC54389.2021.9674247
28. Uzair M, Brinkworth RS, Finn A (2021) Bio-inspired video enhancement for small moving target detection. IEEE Trans Image Process 30:1232–1244. https://doi.org/10.1109/TIP.2020.3043113
29. Luesutthiviboon S, de Croon G, Altena A, Snellen M, Voskuijl M (2023) Bio-inspired enhancement for optical detection of drones using convolutional neural networks. SPIE. https://doi.org/10.1117/12.2673788
30. Kwan C, Larkin J (2021) Detection of small moving objects in long range infrared videos from a change detection perspective. Photonics 8:394. https://doi.org/10.3390/photonics8090394
31. Uzair M, Brinkworth RS, Finn A (2021) A bio-inspired spatiotemporal contrast operator for small and low-heat-signature target detection in infrared imagery. Neural Comput Appl 33:7311–7324. https://doi.org/10.1007/s00521-020-05206-w
32. Coluccia A, Fascista A, Schumann A, Sommer L, Dimou A, Zarpalas D, Méndez M, Iglesia D, González I, Mercier J-P, Gagné G, Mitra A, Rajashekar S (2021) Drone vs. bird detection: deep learning algorithms and results from a grand challenge. Sensors 21(8):2824. https://doi.org/10.3390/s21082824
33. Guo H, Zheng C, Zhao S (2024) Motion-guided small MAV detection in complex and non-planar scenes. Pattern Recognit Lett 186:98–105. https://doi.org/10.1016/j.patrec.2024.09.013
34. Han Y, Han W, Li L, Zhang T, Wang Y (2023) Identifying critical kinematic features of animate motion and contribution to animacy perception. IScience 26(9):107658. https://doi.org/10.1016/j.isci.2023.107658
35. Srigrarom, S., Hoe Chew, K., Meng Da Lee, D., Ratsamee, P.: Drone versus Bird Flights: Classification by Trajectories Characterization. IEEE (2020). 10.23919/sice48898.2020.9240313
36. Chan JJX, Srigrarom S, Cao J, Wang P, Ratsamee P (2021) Small flying object classifications based on trajectories and support vector machines. J Robot Mechatron 33(2):329–338. https://doi.org/10.20965/jrm.2021.p0329
37. Luo W, Xing J, Milan A, Zhang X, Liu W, Kim T-K (2021) Multiple object tracking: a literature review. Artif Intell 293:103448. https://doi.org/10.1016/j.artint.2020.103448
38. Yang B, Nevatia R (2012) Online learned discriminative part-based appearance models for multi-human tracking. Springer. https://doi.org/10.1007/978-3-642-33718-5_35

39. Svanström F, Englund C, Alonso-Fernandez F (2021) Real-Time Drone Detection and Tracking With Visible. Thermal and Acoustic Sensors IEEE. https://doi.org/10.1109/ICPR48806.2021.9413241
40. Zheng Y, Zheng C, Zhang X, Chen F, Chen Z, Zhao S (2022) Detection, localization, and tracking of multiple MAVs with panoramic stereo camera networks. IEEE Trans Autom Sci Eng 1–18. https://doi.org/10.1109/TASE.2022.3176294
41. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: Single Shot MultiBox Detector, Amsterdam. The Netherlands. https://doi.org/10.1007/978-3-319-46448-0_2
42. Lee D-H (2020) CNN-based single object detection and tracking in videos and its application to drone detection. Multimedia Tools Appl 80(26–27):34237–34248. https://doi.org/10.1007/s11042-020-09924-0
43. Seidaliyeva U, Akhmetov D, Ilipbayeva L, Matson ET (2020) Real-time and accurate drone detection in a video with a static background. Sensors (Basel) 20:3856. https://doi.org/10.3390/s20143856
44. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C.: MobileNetV2: Inverted Residuals and Linear Bottlenecks (2018)
45. Poiesi F, Cavallaro A (2016) Detection of fast incoming objects with a moving camera. British Machine Vision Association. https://doi.org/10.5244/c.30.146
46. Kwan C, Budavari B (2020) Enhancing small moving target detection performance in low-quality and long-range infrared videos using optical flow techniques. Remote Sens (Basel) 12(24):4024. https://doi.org/10.3390/rs12244024
47. Yin J, Liu L, Li H, Liu Q (2016) The infrared moving object detection and security detection related algorithms based on W4 and frame difference. Infrared Phys Technol 77:302–315. https://doi.org/10.1016/j.infrared.2016.06.004
48. Thai V-P, Zhong W, Pham T, Alam S, Duong V (2019) Detection, Tracking and Classification of Aircraft and Drones in Digital Towers Using Machine Learning on Motion Patterns. IEEE. https://doi.org/10.1109/ICNSURV.2019.8735240
49. Dong Y, Wu F, Zhang S, Chen G, Hu Y, Yano M, Sun J, Huang S, Liu F, Dai Q, Cheng Z-Q (2025) Securing the Skies: A comprehensive survey on anti-UAV methods, benchmarking, and future directions. arXiv. https://doi.org/10.48550/ARXIV.2504.11967
50. Bingwei Hui, Zhiyong Song, Hongqi Fan, Zhong, P., Weidong Hu, Xiaofeng Zhang, Jianguo Lin, Hongyan Su, Jin, W., Yongjie Zhang, Yaxi Bai: A dataset for infrared image dim-small aircraft target detection and tracking under ground / air background. Science Data Bank (2019). 10.57760/SCIENCEDB.902
51. Zhao J, Zhang J, Li D, Wang D (2022) Vision-based anti-UAV betection and tracking. arXiv. https://doi.org/10.48550/ARXIV.2205.10851
52. Jiang N, Wang K, Peng X, Yu X, Wang Q, Xing J, Li G, Zhao J, Guo G, Han Z (2021) Anti-UAV: A large multi-modal benchmark for UAV tracking. arXiv. https://doi.org/10.48550/ARXIV.2101.08466
53. Huang B, Li J, Chen J, Wang G, Zhao J, Xu T (2024) Anti-uav410: a thermal infrared benchmark and customized scheme for tracking drones in the wild. IEEE Trans Pattern Anal Mach Intell 46(5):2852–2865. https://doi.org/10.1109/tpami.2023.3335338
54. Svanström F, Alonso-Fernandez F, Englund C (2021) A dataset for multi-sensor drone detection. Data Brief 39:107521. https://doi.org/10.1016/j.dib.2021.107521
55. Shortis MR, Clarke TA, Short T (1994) Comparison of some techniques for the subpixel location of discrete target images. SPIE. https://doi.org/10.1117/12.189136
56. Vlachos M, Gunopulos D, Das G (2004) Rotation invariant distance measures for trajectories. ACM. https://doi.org/10.1145/1014052.1014144
57. Ghanbarian B, Hunt AG, Ewing RP, Sahimi M (2013) Tortuosity in porous media: a critical review. Soil Sci Soc Am J 77(5):1461–1477. https://doi.org/10.2136/sssaj2012.0435
58. Miller C, Christman MC, Estevez I (2011) Movement in a confined space: estimating path tortuosity. Appl Anim Behav Sci 135(1–2):13–23. https://doi.org/10.1016/j.applanim.2011.09.002
59. Kayed, M., Anter, A., Mohamed, H.: Classification of garments from fashion MNIST dataset using CNN LeNet-5 architecture (2020). 10.1109/ITCE48509.2020.9047776
60. He K, Zhang X, Ren S, Sun J (2015) Deep residual learning for image recognition. arXiv. https://doi.org/10.48550/ARXIV.1512.03385
61. Oh, H.M., Lee, H., Kim, M.Y.: Comparing Convolutional Neural Network (CNN) models for machine learning-based drone and bird classification of anti-drone system. IEEE (2019). 10.23919/iccas47443.2019.8971699
62. Koninklijk Nederlands Meteorologisch Instituut (KNMI): Uurwaarnemingen. (Accessed: 27 September 2025). https://daggegevens.knmi.nl/klimatologie/uurgegevens
63. Mancinelli A, Smeur EJJ, Remes B, de Croon G (2022) Dual-axis tilting rotor quad-plane design, simulation, flight and performance comparison with a conventional quad-plane design. IEEE. https://doi.org/10.1109/icuas54217.2022.9836063

Authors and Affiliations

Salil Luesutthiviboon[1,2] · Guido C. H. E. de Croon[1] · Anique Altena[1] · Mirjam Snellen[1] · Mark Voskuijl[3]

✉ Salil Luesutthiviboon
salil.luesutthiviboon@tno.nl

Guido C. H. E. de Croon
g.c.h.e.decroon@tudelft.nl

Anique Altena
a.altena@tudelft.nl

Mirjam Snellen
m.snellen@tudelft.nl

Mark Voskuijl
m.voskuijl@mindef.nl

1 Department Control and Operations, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands

2 Present address: TNO Defence, Safety & Security, Oude Waalsdorperweg 63, 2597 AK The Hague, The Netherlands

3 Faculty of Military Sciences, Netherlands Defence Academy, Het Nieuwe Diep 8, 1781 AC Den Helder, The Netherlands