Delft University of Technology

Bio-inspired enhancement for optical detection of drones using convolutional neural networks

Luesutthiviboon, Salil; de Croon, Guido C.H.E.; Altena, Anique V.N.; Snellen, Mirjam; Voskuijl, Mark

DOI
10.1117/12.2673788

Publication date
2023

Document Version
Final published version

Published in
Artificial Intelligence for Security and Defence Applications

Citation (APA)
Luesutthiviboon, S., de Croon, G. C. H. E., Altena, A. V. N., Snellen, M., & Voskuijl, M. (2023). Bio-inspired enhancement for optical detection of drones using convolutional neural networks. In H. Bouma, J. Dijk, R. Prabhu, R. J. Stokes, & Y. Yitzhaky (Eds.), *Artificial Intelligence for Security and Defence Applications* Article 127420F (Proceedings of SPIE - The International Society for Optical Engineering; Vol. 12742). SPIE. https://doi.org/10.1117/12.2673788

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Bio-inspired enhancement for optical detection of drones using convolutional neural networks

Salil Luesutthiviboon[a], Guido C. H. E. de Croon[a], Anique V. N. Altena[a], Mirjam Snellen[a], and Mark Voskuijl[b]

[a]Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands
[b]Faculty of Military Sciences, Netherlands Defence Academy, Het Nieuwe Diep 8, 1781 AC Den Helder, The Netherlands

ABSTRACT

Threats posed by drones urge defence sectors worldwide to develop drone detection systems. Visible-light and infrared cameras complement other sensors in detecting and identifying drones. Application of Convolutional Neural Networks (CNNs), such as the You Only Look Once (YOLO) algorithm, are known to help detect drones in video footage captured by the cameras quickly, and to robustly differentiate drones from other flying objects such as birds, thus avoiding false positives. However, using still video frames for training the CNN may lead to low drone-background contrast when it is flying in front of clutter, and omission of useful temporal data such as the flight trajectory. This deteriorates the drone detection performance, especially when the distance to the target increases. This work proposes to pre-process the video frames using a Bio-Inspired Vision (BIV) model of insects, and to concatenate the pre-processed video frame with the still frame as input for the CNN. The BIV model uses information from preceding frames to enhance the moving target-to-background contrast and embody the target's recent trajectory in the input frames. An open benchmark dataset containing infrared videos of small drones ($<$ 25 kg) and other flying objects is used to train and test the proposed methodology. Results show that, at a high sensor-to-target distance, the YOLO algorithms trained on BIV-processed frames and concatenation of the BIV-processed frames with still frames increase the Average Precision (AP) to 0.92 and 0.88, respectively, compared to 0.83 when it is trained on still frames alone.

Keywords: Drone, Detection, Infrared Camera, Convolutional Neural Network, Machine Learning, Bio-Inspired

1. INTRODUCTION

The use of Unmanned Aerial Vehicles (UAVs), also known as drones, has grown rapidly in recent years. Drones offer new possibilities for industrial, research, recreational, and other sectors. However, they can also pose threats. Drones are appealing for ill-intent uses due to their relatively low costs and ability to covertly carry harmful objects into target areas.[1] Besides, trespassing and disruptions of critical infrastructures can also (un)intentionally be made.[2,3]

Security and defence sectors worldwide have been searching for means to prevent harmful scenarios caused by drones. One of the first steps is to develop and implement drone detection systems for early warnings. Such detection systems comprise of sensors such as radars,[4,5] Radio Frequency (RF) sensors,[6,7] acoustic sensors,[8,9] and optical sensors.[10] Each sensor has its advantages and limitations. For instance, drones that neither transmit nor receive radio signals, cannot be detected by RF sensors,[11] but could still be detected by other sensors such as acoustic sensors, optical sensors, and radars. Therefore, to maximize true detections and minimize false positives, it is desirable to have a detection system in which multiple types of sensors work collaboratively.[12]

Optical sensors can detect drones and can also verify detections made by other sensors. Furthermore, they can help identify types of the drones[13,14] and (harmful) objects they may carry. Optical detection is usually

Further author information: (Send correspondence to S.L.)
S.L.: E-mail: s.luesutthiviboon@tudelft.nl

Artificial Intelligence for Security and Defence Applications, edited by Henri Bouma, Judith Dijk,
Radhakrishna Prabhu, Robert J. Stokes, Yitzhak Yitzhaky, Proc. of SPIE Vol. 12742,
127420F · © 2023 SPIE · 0277-786X · doi: 10.1117/12.2673788

done by Convolutional Neural Networks (CNNs),[15] a specialized computer model inspired by the human brain, trained to perform a task of detecting specific objects from input images, based on their extracted visual features. The use of the CNNs helps to identify the exact object types, and thus avoiding false alarms. In other words, implementing the CNN helps to only detect drones and disregard other irrelevant objects, such as birds.

However, there are several downsides to using the CNNs, and optical sensors in general, for detecting drones. First, increasing the sensor-to-target distance, i.e. smaller apparent target size, can lead to poorer detection performance.[16] Zooming in to gain more resolution for an object far away results in a reduced field of view. Second, cluttered visual backgrounds[17] or noisy video frames can lead to missed detections, or false positives,[16] i.e. detecting irrelevant objects as threats. Third, most CNNs for object detection operate in a frame-by-frame manner by default. The use of still frames disregards temporal information which could also be useful for detection. In other words, drones could have a recognizable flight trajectory[18] which discerns them from other flying objects such as airplanes.

Attempting to overcome the aforementioned limitations, this paper proposes to add an enhancement to the input frame of a CNN for the optical detection of drones. The enhancement model considered here is the Bio-Inspired Vision (BIV) model[19] mimicking photoreceptor cells of insects. The model contains a series of adaptive temporal low-pass filters and other pixel-wise operations to enhance the moving target and suppress the still background, as well as suppressing temporal noise, embodying information from the previous frames. Many researchers have shown successful applications of the BIV processing to enhance the visibility of small targets in video frames,[20, 21] yet the effects of combining the BIV processing with the CNN for object detection are still not examined. This paper therefore investigates the effects of concatenating the BIV-enhanced frames with still frames as input a CNN for drone detection applications. Variations of the detection performance with the sensor-to-target distance are investigated. Moreover, the performance of the CNN trained by BIV enhanced frames is also compared to those trained on still frames and frames trained on the more basic Frame Differencing (FD) technique.

This paper is structured as follows: Section 2 contains the literature review of algorithms, hardware, and benchmark datasets for the optical detection of drones. Section 3 explains the proposed methodology including the BIV processing in further detail. Qualitative and quantitative performance of the BIV processing and the CNN trained on BIV enhanced frames, compared to those of the still frames and FD, are presented in Section 4. Finally, Section 5 concludes this paper and provides recommendations for future steps.

2. RELATED WORK

2.1 Algorithms for Optical Detection of Drones

There are two main groups of algorithms for the optical detection of drones, and small flying objects in general: traditional computer vision and deep learning.[15]

Traditional computer vision algorithms apply a variety of filtering and segmentation techniques to incoming video frames. The main goal is to discern the target from the background. After that, detection is done when the processed pixel value in certain parts of the frame exceeds a detection threshold.[22–24] The techniques applied found in literature are, for instance, black-hat filtering,[25] top-hat filtering,[26] and morphological filtration such as erosion and dilation.[23, 27] Some authors also apply corner detection[18] or edge detection to find the horizon line and limit the search area for flying targets above the horizon.[26] The techniques mentioned so far operate in a frame-by-frame manner, relying on *spatial* features of the frames. On the other hand, some processing techniques utilize *temporal* information to discern the target and the background, i.e. employing information from the preceding frames and assuming that the target moves relative to the background.[28] These techniques are, for instance, frame differencing[23] and optical flow.[18, 22] Clearly, utilizing the temporal features can be beneficial unless the object of interest, e.g. the drone, hovers. Many authors argued and proved that combining spatial and temporal information[25, 29] can be beneficial for small flying target detection because one could benefit from the fact that drones, such as quadcopters, have a recognizable manner of movement,[18] i.e. trajectory, in comparison to other flying objects such as birds or airplanes. Many of these traditional temporal algorithms are sensitive to parameter settings. In contrast, some recent studies have introduced adaptive algorithms inspired by neuroscience that incorporate spatio-temporal feedback to process the incoming video frames.[28] One of such techniques is inspired photoreceptors of insects[30] which enhance the small moving target and suppress the background in an adaptive manner, based on the current environment. The main advantage of the traditional computer vision algorithms is that they do not require training.[23] Most of the traditional computer vision techniques mentioned are generally fast and can be deployed for real-time applications. However, they do not exploit the specific appearance of the object and may have limitations in recognizing different types of flying objects.

On the other hand, deep learning methods, specifically Convolutional Neural Networks (CNNs), reach the highest performance for recognizing targets of interest based on their appearance. For optical detection of drones, CNNs can help discern drones from other flying objects,[16,31] and theoretically can differentiate drone types if trained properly.[13,14] CNNs need to learn from labeled training data in order to perform detection of certain objects. There are two sub-groups of deep learning algorithms: two-stage and one-stage.[15]

Two-stage CNNs[31,32] such as Region-CNN (RCNN) process multiple patches in the input image in order to propose a region of interest, where the target of interest may be present. Next, object detection is carried out. RCNNs and their variants are regarded as more accurate, but the two-stage architecture requires longer training and processing time.[33] Therefore, RCNNs may not be suitable for real-time detection.

For this purpose, one-stage CNNs, where the entire input image is processed, are more suitable. Popular one-stage CNN techniques applied in literature for optical detection of drones are the You Only Look Once (YOLO) algorithm and its variants[14,16,24,34] and the Single-Shot Detector (SSD).[33,35] Park et al.[33] compared multiple CNN algorithms for drone detection. The authors concluded that, for drone detection applications, the YOLOv2 algorithm gives the best compromise between the detection accuracy and the processing time.

By default, the CNNs process the video frame-by-frame. However, in this way, useful temporal information such as the trajectory is not used or the aforementioned temporal computer vision techniques that help to enhance target-to-background contrast are skipped. This could lead to not exploiting the full potential of the CNNs. Besides, the detection performance of the CNN techniques such as YOLO, which rely heavily on the target appearance, is commonly known to drop when the distance to the target increases.[23] To help enhance performance of the CNNs some authors suggest pre-processing the input frames for the CNN using computer vision techniques.[24,27] Some also suggest that incorporating spatio-temporal features, such as incorporating the aforementioned bio-inspired vision model could help boost the detection performance.[30]

2.2 Hardware and Datasets

Optical cameras, operating in the visible-light spectrum, are the most frequently used hardware for the optical detection of drones in literature.[14,18,24,25,27,29,31–33,36] The optical camera types and datasets vary depending on the application. In literature, applications for the optical detection of drones are roughly divided into two groups. The first application is for security and surveillance, which is the focus of this paper. In this case, the ground-based cameras are either static[14,18,25,27,31,36] or mounted on a movable pan-and-tilt platform.[33,37] An example benchmark dataset for this application is the 'Drone-vs-Bird' detection challenge,[38] containing real-world videos of drones and birds to account for the challenge of discerning birds from drones. The second application for optical detection of drones is for collision avoidance. For this purpose, the optical camera is mounted on a drone to film other flying targets.[29,32] There are extensive datasets for this application such as the 'NPS-Drones',[39] 'FL-Drones',[40] and 'AOT-Drones' datasets.[41]

Optical cameras are known to have several limitations. Detection becomes challenging when the target is in front of a cluttered visual background.[17] Furthermore, optical cameras are heavily affected by lighting conditions;[42] too bright surroundings lead to overexposure, and the target becomes invisible in the dark.

These limitations motivated some authors to opt for alternative types of cameras. Infrared cameras,[23,26,43] for instance, visualize the temperature of objects in the scenes and help to detect an object regardless of visual clutters and lighting conditions. For drone detection, this is helpful because some parts of the drone such as the engine or battery are warmer than the surroundings.[44] Some authors also demonstrated that the detection performance of drones using infrared cameras can surpass that of optical cameras.[16,43] Another type of camera is the event-based camera which monitors changes in pixel intensity with a high temporal resolution and output 'events' when the pixel intensity change exceeds a certain threshold. An event-based camera has also shown promising results in detecting small drones as demonstrated by Shu et al.[22]

Research as well as benchmarking datasets involving the alternative types of cameras are still lacking. To the best of our knowledge, the only open benchmark dataset dedicated to drone detection in security and surveillance applications containing both visual and infrared videos of drones is that of Svanström et al.,[22] which has been made available in 2021.

3. METHODOLOGY

3.1 The Proposed Method

We propose preprocessing the incoming video frames using a Bio-Inspired Vision (BIV) model[19] for drone detection with the YOLOv2[45] algorithm. Instead of using still frames in the YOLOv2 algorithm, the frames preprocessed by the BIV model are expected to help preserve temporal information and adjust the contrast of the moving target to the still background. A schematic of the processing techniques considered is shown in Fig. 1.

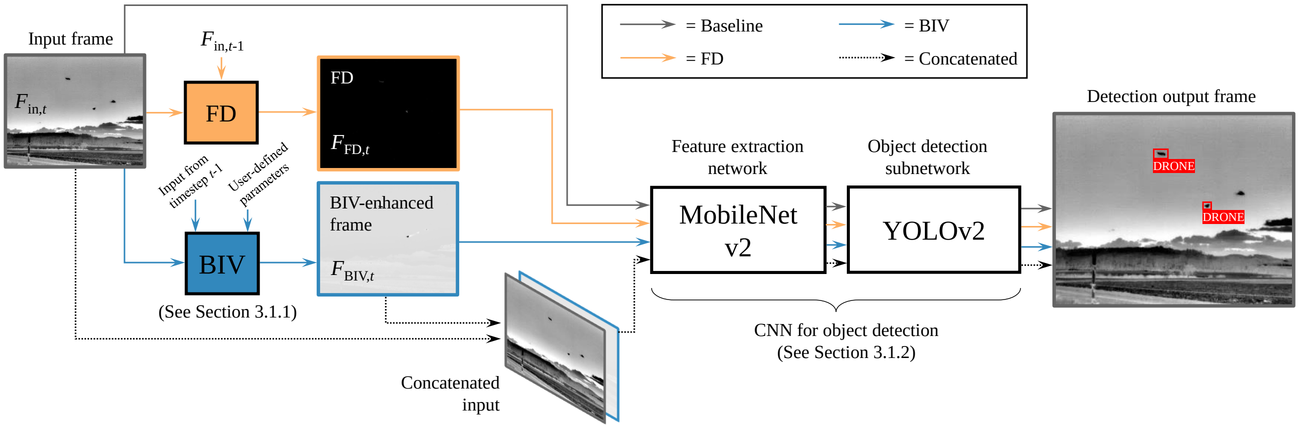


Figure 1: Schematic of the processing techniques considered.

Next to this, we propose concatenating the BIV-enhanced frame with the still frame as inputs for the YOLOv2 model. This case is called the 'Concatenated' case. As references, we also use still frames, BIV-processed frames, and frames processed by the Frame Differencing (FD) technique as separate inputs for the YOLOv2 detection. These cases are further denoted as the 'Baseline', 'BIV', and 'FD' cases, respectively. The BIV framework and the object detection framework using YOLOv2 are further explained in the following subsections.

3.1.1 The Bio-Inspired Vision (BIV) Model

The BIV model considered in this paper is based on a model of insect photoreceptor cells introduced by van Hateren.[19] A diagram of the BIV model is shown in Fig. 2. This four-stage model processes the incoming frames in a pixel-wise manner. It contains multiple adaptive Low-Pass Filters (LPFs) to suppress temporal noise, gain control, and other operations to enhance the contrast between the moving target and the background. Previous works show performance enhancement for small object detection using infrared videos when the BIV model is applied.[20,21] Furthermore, promising results have also been obtained when applying the BIV model to enhance optical flow estimation,[46] and spectrograms[30] for acoustic detection of drones. Each stage of the considered BIV model is explained in further detail in the following.

The first stage of the model helps to suppress high-frequency temporal noise and enhance the target to background contrast. Let $I_{\mathrm{in},t}$ denote a generic incoming pixel value from an input frame F_t at time t. A generic diagram of the temporal LPF function f_{LPF} for any generic input I_t is shown in the lower-right corner of Fig. 2 and is expressed as follows:

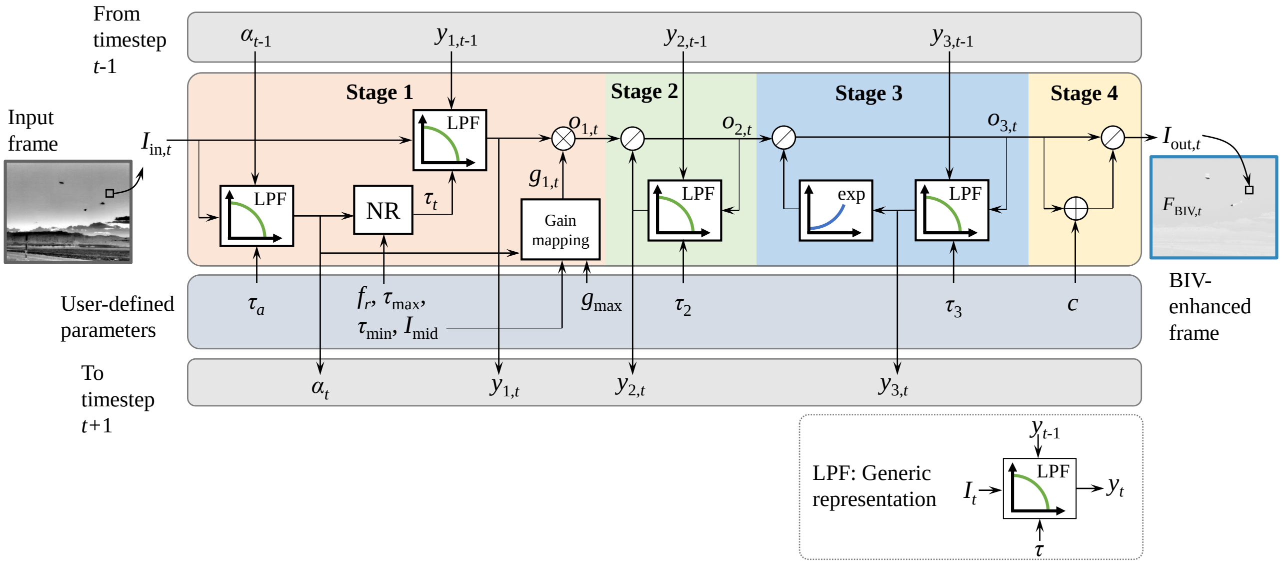


Figure 2: BIV model inspired by insect photoreceptor cells (Adapted from Fang et al.[30] and Uzair et al.[20,21]).

$$y_t = f_{\text{LPF}}(I_t, \tau, y_{t-1}) = \frac{1}{1+\tau} I_t + \left(1 - \frac{1}{1+\tau}\right) y_{t-1}, \tag{1}$$

where y_{t-1} is an input from the previous timestep $t-1$, τ is a generic time constant, and y_t is a generic output.

For timestep t in stage 1, the introduced f_{LPF} is first applied to calculate the adaptation level α_t, $\alpha_t = f_{\text{LPF}}(I_{\text{in},t}, \tau_a, \alpha_{t-1})$, where τ_a is a user-defined adaptation level time constant. Next, an adaptive time constant τ_t is calculated by a Naka-Rushton (NR) transform[47] as follows:

$$\tau_t = \frac{f_r}{2\pi[(\tau_{\max} - \tau_{\min})\left(\frac{\alpha_t}{\alpha_t - I_{\text{mid}}}\right) + \tau_{\min}]}, \tag{2}$$

where f_r is the frame rate, I_{mid}, $\tau_{\max}$, and $\tau_{\min}$ are the user-defined mid-range pixel value, maximum and minimum adaptation rates, respectively. The obtained τ_t is then used in an LPF of $I_{\text{in},t}$ yielding $y_{1,t} = f_{\text{LPF}}(I_{\text{in},t}, \tau_t, y_{1,t-1})$ as shown in the diagram.

Finally, an adaptive gain $g_{1,t}$ is applied to enhance the low pixel value, i.e. the target. The value of $g_{1,t}$ is calculated based on I_{mid} and α_t (denoted by the 'Gain mapping' block in Fig. 2),

$$g_{1,t} = (g_{\max} - 1)\left(1 - \frac{\alpha_t}{\alpha_t + I_{\text{mid}}}\right) + 1, \tag{3}$$

where $g_{\max}$ is the user-defined maximum gain. This gives the output of stage 1 $o_{1,t}$:

$$o_{1,t} = g_{1,t} y_{1,t}. \tag{4}$$

Next, stages 2 and 3 contain divisive feedback loops which provide short- and long-term adaptations of the input, respectively. The main differentiation of stage 3 to stage 2 is the exponential scaling and the relatively longer user-defined time constant τ_3. The expected output is the preservation of the moving object edges while suppressing still or slower objects in the frame.[21] Mathematically, stages 2 and 3 operate as follows:

$$o_{2,t} = \frac{o_{1,t}}{y_{2,t}}, \text{ where } y_{2,t} = f_{\text{LPF}}(o_{2,t-1}, \tau_2, y_{2,t-1}), \tag{5}$$

and

$$o_{3,t} = \frac{o_{2,t}}{\exp(y_{3,t})}, \text{ where } y_{3,t} = f_{\mathrm{LPF}}(o_{3,t-1}, \tau_3, y_{3,t-1}). \tag{6}$$

Finally, stage 4 applies a NR non-linearity to control the output range using a user-defined positive offset c. Low input values are scaled almost linearly while high input values are relatively more compressed. The output pixel value at time t, $I_{\mathrm{out},t}$, is given by

$$I_{\mathrm{out},t} = \frac{o_{3,t}}{o_{3,t} + c}. \tag{7}$$

The generic output pixels $I_{\mathrm{out},t}$ compose the BIV-enhanced frame at timestep t, $F_{\mathrm{BIV},t}$. Qualitative comparisons of F_t and $F_{\mathrm{BIV},t}$ are discussed in Section 4.1.1.

In this work, the user-defined parameters for the BIV model are selected by initially following the work of Uzair et al.[20] where the BIV model was also applied to infrared-camera videos. Some values are experimentally adjusted to suit the video dataset considered in our paper. The parameters and their set values are summarized in Table 1.

Table 1: User-defined parameters for the BIV model.

Symbols	Definitions	Set values
τ_a	Adaptation level time constant	10
f_r	Frame rate	30
$\tau_{\max}$	Maximum adaptation rate	15
$\tau_{\min}$	Minimum adaptation rate	3
I_{mid}	Mid-range pixel value	20
$g_{\max}$	Maximum gain	50
τ_2	Time constant for stage 2	10
τ_3	Time constant for stage 3	100
c	Positive offset	0.75

3.1.2 The MobileNet-v2 and YOLOv2 Object Detection Network

In this work, we generated an object detection network by combining two CNNs: MobileNet-v2[48] and YOLOv2.[45] The MobileNet-v2 was pruned and used as a feature extraction backbone and YOLOv2 was used for object detection. The motivation for such an architecture was twofold. First, such an architecture is lightweight and can be deployed in field applications using mobile devices.[49,50] Recent comparative reviews agree that YOLOv2 gives the best compromise between the processing time and the detection accuracy for real-time drone detection.[33] Successful applications of MobileNet-v2 and YOLOv2 have been presented in literature.[16,51] Secondly, such architecture has been employed in the benchmark dataset of Svanström et al.[37] The present implementation can be verified by comparing the results.

For the feature extraction backbone, we use the first 12 convolutional layers of the MobileNet-v2 network. After this stage, the input image with the size of 256 × 256 becomes 16 × 16, corresponding to the number of grid cells in the subsequent YOLOv2 object detection subnetwork. The extracted features are then fed into the YOLOv2 object detection neck, comprising of 3 convolutional layers with anchor boxes. For each grid cell, the bounding box locations, objectness score, and class confidence score based on a predefined number and locations of anchor boxes are determined, composing an output tensor for object detection. In this paper, the number of anchor boxes per grid cell is taken as 3, in agreement with Svanström et al.[37] This selected number of anchor boxes gives the best compromise between maximizing the mean Intersection over Union (IoU) of the anchor boxes and the ground-truth bounding boxes, and minimizing the complexity of the YOLOv2 algorithms, i.e. the dimension of the output tensor. The aspect ratios of the anchor boxes are chosen according to k-mean clustering of the ground-truth bounding boxes in the training dataset.

3.2 Training and Testing Using a Benchmark Dataset

3.2.1 The Benchmark Dataset

For training and testing the algorithms, we employ a multi-sensor dataset for drone detection of Svanström et al.,[37] which is an open-access online dataset containing, among others, videos of drones, airplanes, birds, and helicopters filmed by visible-light and infrared cameras. In our work, we utilize the infrared camera videos. The dataset contains 365 videos filmed by a FLIR Breach PTQ-136 infrared camera. Each video is approximately 10 seconds long and has a frame rate of 30 frames per second. Therefore, there are approximately 300 frames per video. The pixel resolution of the videos is 320×256 and the field of view is 24°×19° (horizontal×vertical).

Every video is accompanied by a ground-truth labeling file containing locations of the ground truth bounding boxes and object classes, namely, 'AIRPLANE', 'BIRD', 'DRONE', and 'HELICOPTER', for every frame. It is important to note that the 'DRONE' in this considered dataset concerns only three types of quadcopters. Thus, other variations of drones such as fixed-wing drones are not considered. Each video contains only one of the object classes, but some frames may contain multiple objects belonging to that class. Moreover, every video file is categorized according to three different distance bins: 'CLOSE', 'MEDIUM', and 'DISTANT', depending on the apparent size of the target object in pixels according to the Detection, Recognition, and Identification (DRI) requirement specifications.[52,53]

3.2.2 Training and Testing the CNN

We trained the CNN to recognize 2 classes of flying objects. We considered 'DRONE' and Non-Drone Flying Objects, or 'NDFO'. To achieve the intended class division, we subdivided the benchmark video dataset as shown in Table 2. We selected the number of videos for each distance bin such that the different object classes are represented uniformly. For all the videos selected in one training and testing cycle, about 80% were used for training and the remaining (and not overlapping) 20% were used for testing. Note that the 'NDFO' class does not officially exist in the benchmark dataset. Therefore, we created the 'NDFO' class by combining the videos from the 'AIRPLANE', 'BIRD', and 'HELICOPTER' classes equally. Due to an insufficient number of videos and the irrelevance of the 'CLOSE' distance bin, we do not consider this distance bin for the 'NDFO' class.

Table 2: Number of videos from the original dataset per class and distance bins used for training and testing the CNNs ('C'='CLOSE', 'M'='MEDIUM', and 'D'='DISTANT').

Training/ testing	Number of videos from the original dataset per class and distance bins											
	'AIRPLANE'			'BIRD'			'DRONE'			'HELICOPTER'		
	'C'	'M'	'D'	'C'	'M'	'D'	'C'	'M'	'D'	'C'	'M'	'D'
Training	–	16	10	–	8	10	12	32	30	–	8	10
Testing	–	4	2	–	2	2	3	8	6	–	2	2

The considered CNNs were created in MATLAB. After that, the created CNNs were trained and tested using the videos mentioned earlier. The training parameter settings are specified in Table 3.

Table 3: Training parameter settings for the CNN.

Training parameters/ options	Set values/ selected options
Optimization algorithm for network learnable parameters	Stochastic Gradient Descent with Momentum (SGDM)
Initial learn rate	0.001
Number of epochs	5
Validation ratio	0.1
Validation frequency	Every 5000 iteration
Training data shuffling frequency	Every epoch

4. RESULTS AND DISCUSSION

The results are discussed in two parts. First, the visual impacts of the BIV processing and object detection in a test video sequence are discussed qualitatively in Section 4.1. Next, quantitative evaluations of the object detection performance metrics are presented in Section 4.2.

4.1 Qualitative Performance Evaluation

4.1.1 Visual Impacts of the BIV Processing

Figure 3 presents baseline (left column) and BIV-processed (right column) infrared video frames containing various object classes from the benchmark dataset of Svanström et al.[37] The object classes as inherently defined in the dataset as well as the ground-truth bounding boxes are also shown.

In general, it can be seen that the BIV processing helps to enhance the target-to-background contrast by suppressing the still background such as trees. This is beneficial in reducing the clutter in the scene. Moreover, an irrelevant object that the detector could confuse with the relevant target, such as the cloud in Fig. 3a becomes almost invisible in Fig. 3b. The BIV processing also shows the ability to suppress noise in the video. This can be seen, for instance, from Fig. 3c which is inherently noisy. The airplane becomes more distinct after the BIV-processing as shown in Fig. 3d. It is also noteworthy from Figs. 3c and 3d that the provided ground-truth bounding box is too large and is not perfectly centered at the target. This could be due to the noisiness in this video in combination with the semi-automated ground-truth labeling.[37] The visual impacts of the BIV processing are also seen in the bird and helicopter videos from Fig. 3e to Fig. 3h. Notably, the trajectory history of these objects is also visible as light-colored trails in the BIV-processed frames in Figs. 3f and 3h.

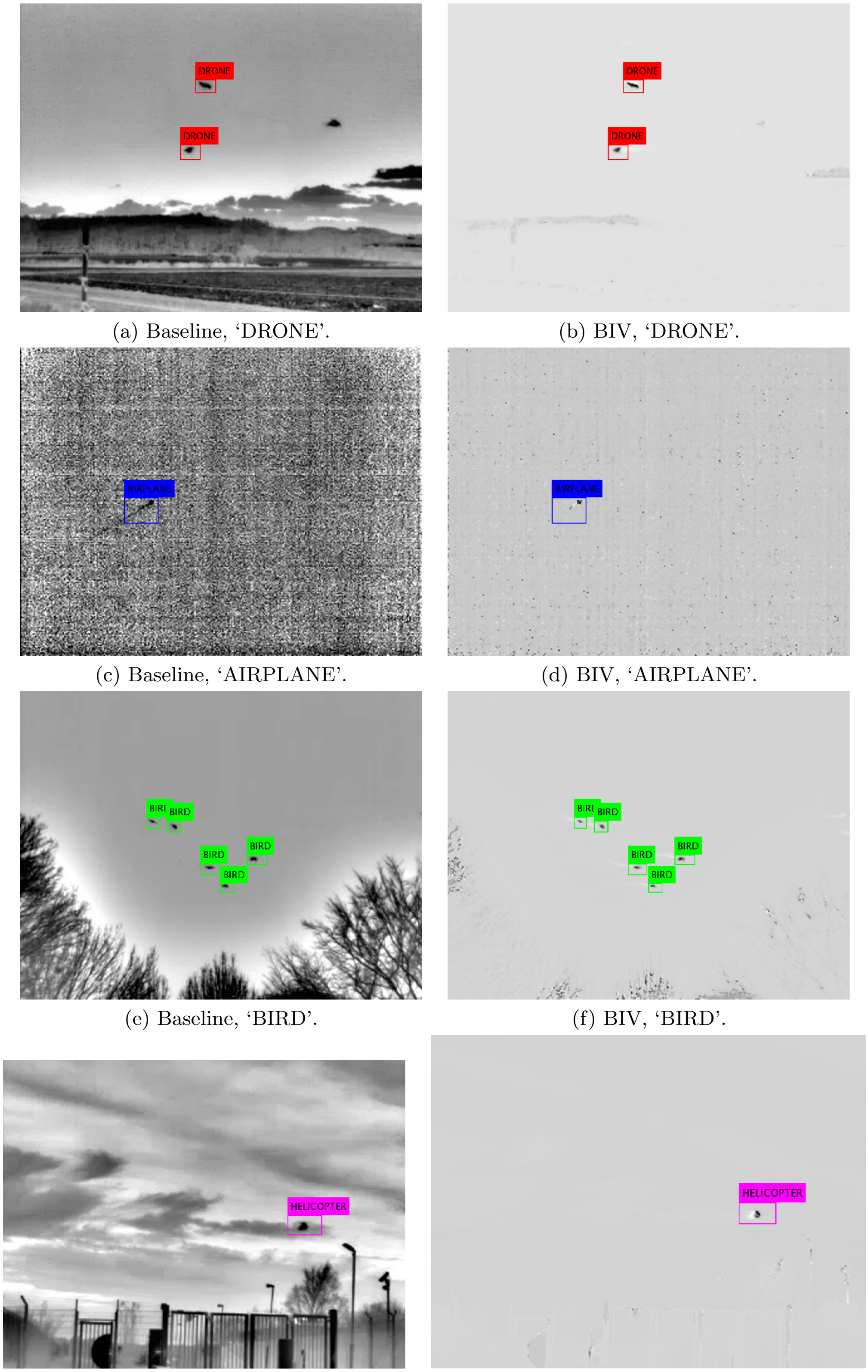
4.1.2 Detection Timeline

Detection timelines in Fig. 4 help to qualitatively examine the object detection behavior of the considered algorithms. In this figure, the detection timelines of the considered algorithms for a portion of the test videos are shown, with a focus on the test videos containing drones. The detection timelines by the considered algorithms are shown and color-coded in terms of the confusion matrix, namely True Positives (TPs), False Positives (FPs), False Negatives (FNs), and True Negatives (TNs), for the object class 'DRONE'. The ground-truth class and distance bin of each video are also annotated above. Here, and where applicable in this paper, the detection confidence score threshold is taken as 0.5, while the Intersection-over-Union (IoU) threshold which quantifies the overlap of the ground-truth bounding boxes and the detection bounding boxes for the correct detection is taken at 0.35. The latter is a rather relaxed criterion chosen based on observations that the ground-truth bounding boxes in the dataset are sometimes too large.

Generally, the detection performance of most of the algorithms is satisfactory and comparable. There are continuous stripes of TPs for the ground-truth class 'DRONE' and TNs for the ground-truth class 'NDFO'. However, there are some videos with particularly more concentrated FPs. The FPs may also occur even for the ground-truth class 'DRONE'. This occurs when a detection lies elsewhere in the frame, i.e. when the IoU of the detection bounding box and the ground-truth bounding box is below the set IoU threshold. Notably, the CNN trained on the FD frames has the highest number of FNs, i.e. missed detections, especially when the sensor-to-target distance is large. This is likely because the FD algorithm relies on the visual difference between two adjacent frames, and thus the target's apparent motion. In some instances, such as when the drone hovers, the FD frames are completely dark (see, for instance, Fig. 1) and it is almost impossible to detect any object.

Two detection snapshots from two test videos with a high concentration of FPs (indicated by '1' and '2' in Fig. 4) are examined further in Fig. 5. In this analysis, the CNNs trained on Baseline and BIV-processed frames are considered in the left and right columns, respectively. The ground-truth bounding boxes are shown and, when an object is detected, the detection bounding boxes and the detected object class annotation with the confidence score are shown.

In snapshot 1, a bird appears as a small dot. Although the bird is correctly detected as 'NDFO', a tree is detected as 'DRONE' by the CNN trained on Baseline frames as shown in Fig. 5a. The BIV processing helps to keep the bird visible while suppressing the tree visibility in the frame as shown in Fig. 5b. The CNN trained on BIV-processed frames suppresses this false positive. In snapshot 2, the helicopter and another irrelevant object



(a) Baseline, 'DRONE'.

(b) BIV, 'DRONE'.

(c) Baseline, 'AIRPLANE'.

(d) BIV, 'AIRPLANE'.

(e) Baseline, 'BIRD'.

(f) BIV, 'BIRD'.

(g) Baseline, 'HELICOPTER'.

(h) BIV, 'HELICOPTER'.

Figure 3: Baseline (left column) and BIV-processed frames (right column) for various object classes annotated together with the ground-truth bounding boxes.

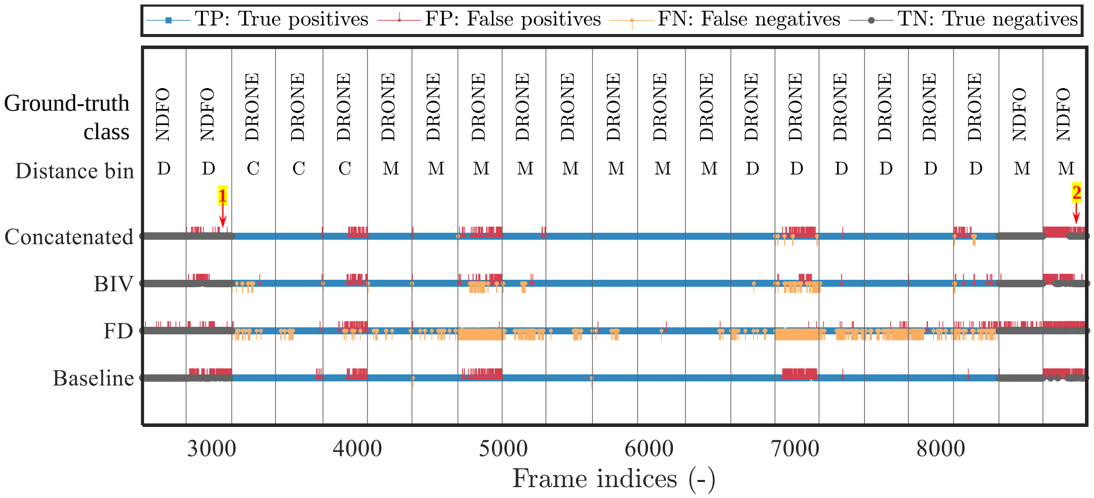
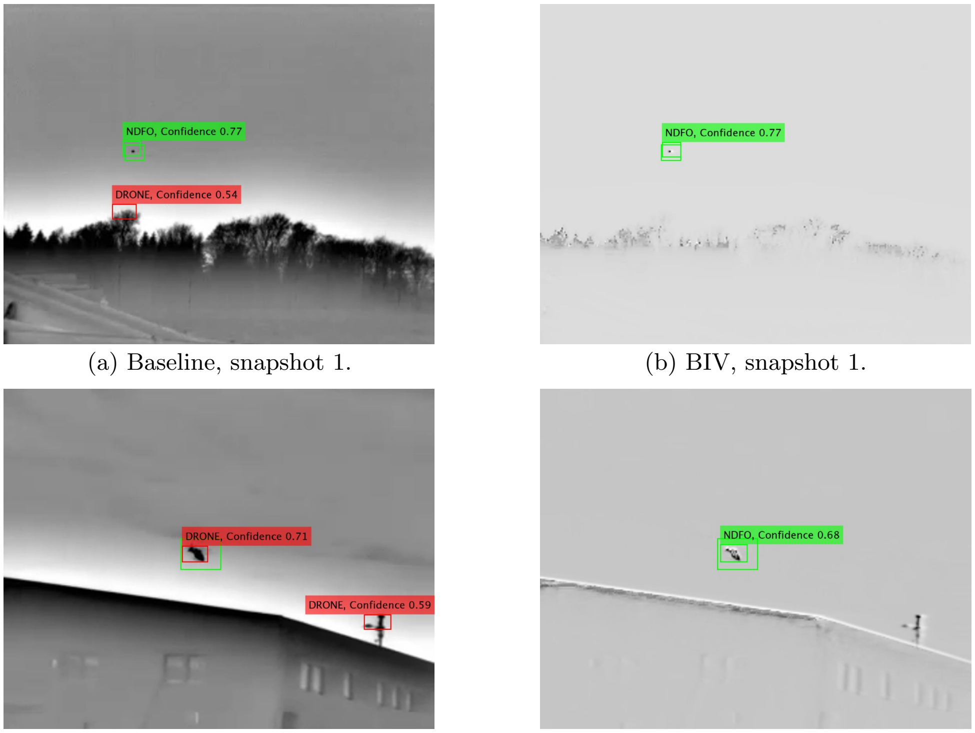


Figure 4: Detection timelines of the considered algorithms for a sequence of test videos, with a zoom-in on a part containing drones. The detection timelines are color-coded in terms of the confusion matrix for the object class 'DRONE'. The ground-truth class and distance bin of each video are also annotated ('C'='CLOSE', 'M'='MEDIUM', and 'D'='DISTANT').

are detected as 'DRONE' by the CNN trained on Baseline frames as shown in Fig. 5c, while the CNN trained on BIV-processed frames detects this helicopter correctly as 'NDFO'. Based on these observations, it is expected that incorporating the BIV processing will help to suppress false positives and thereby improve the detection precision by the CNN.



(a) Baseline, snapshot 1.

(b) BIV, snapshot 1.

(c) Baseline, snapshot 2.

(d) BIV, snapshot 2.

Figure 5: Detection snapshots by the Baseline and BIV CNN object detectors. The snapshot numbers correspond to that indicated by '1' and '2' in Fig. 4. The ground-truth bounding boxes are shown and, when an object is detected, the detection bounding boxes and the detected object class annotation with the confidence score are shown.

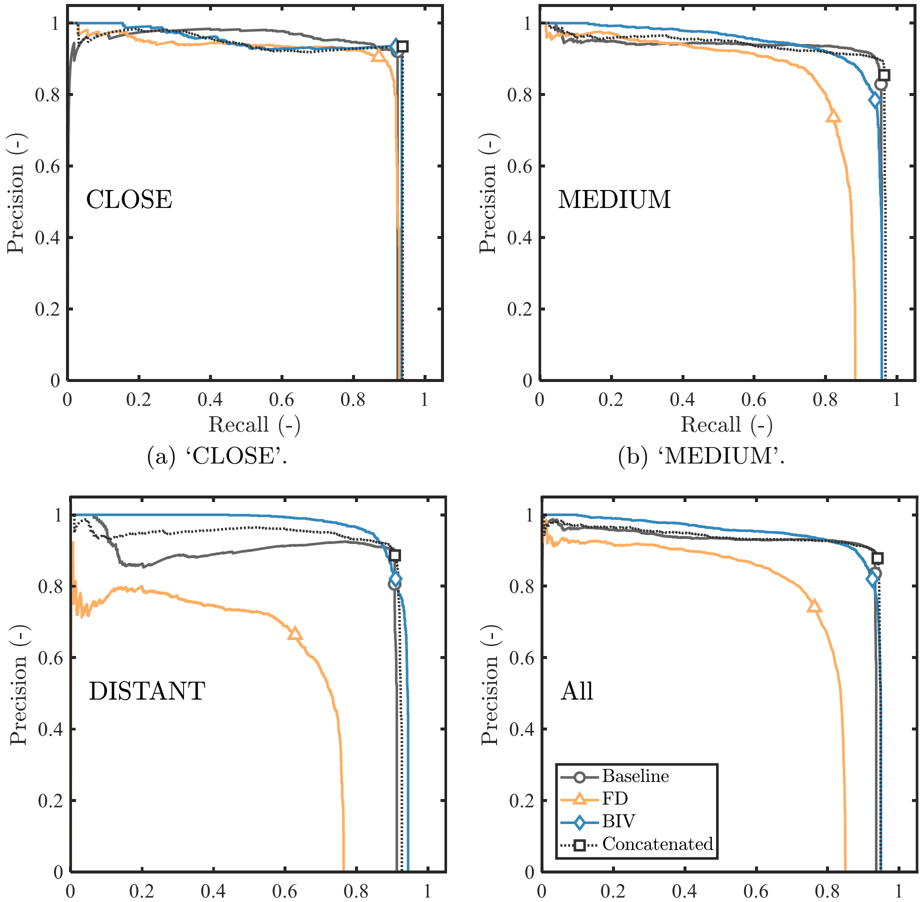
4.2 Quantitative Performance Evaluation

Performance metrics based on the confusion matrix, namely the precision, recall, and False-Positive Rate (FPR), are used to quantitatively evaluate and compare the performance of the considered algorithms. For an object class k, the precision, recall, and FPR are defined as follows:

$$\text{Precision}_k = \frac{n(\text{TP}_k)}{n(\text{TP}_k) + n(\text{FP}_k)} \quad (8) \qquad \text{Recall}_k = \frac{n(\text{TP}_k)}{n(\text{TP}_k) + n(\text{FN}_k)} \quad (9) \qquad \text{FPR}_k = \frac{n(\text{FP}_k)}{n(\text{FP}_k) + n(\text{TN}_k)} \quad (10)$$

where $n(\dots)$ means 'the number of . . . '.

The precision and recall vary inversely with each other, depending on the detection threshold. Therefore, it is more holistic to consider the precision and recall together using, for example, Precision-Recall (PR) curves or F1 score (the geometric mean of the precision and recall).



(a) 'CLOSE'.

(b) 'MEDIUM'.

(c) 'DISTANT'.

(d) 'All'.

Figure 6: PR curves for the considered algorithms for the object class 'DRONE' subdivided according to the labeled distance bins of the test data. The markers indicate the precision and recall values for the detection confidence score threshold of 0.5.

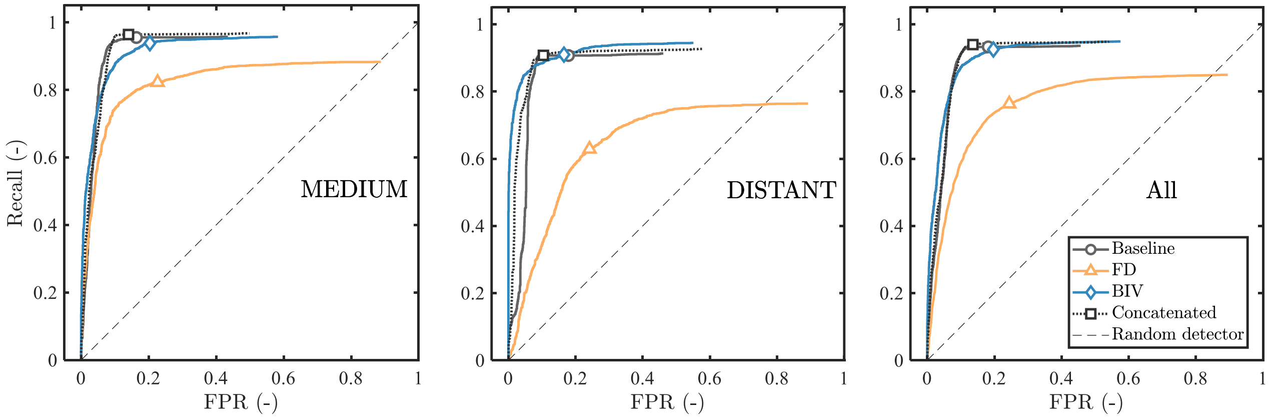
The PR curves for the considered algorithms for the object class 'DRONE' subdivided according to the labeled distance bins of the test data are shown in Fig. 6. Correspondingly, the so-called Receiver's Operating Characteristics (ROC) curves, showing the recall and the FPR for the considered algorithms are also shown in

Fig. 7. For both of the figures, the markers indicate the values of the precision, recall, and FPR when the detection confidence threshold is taken as 0.5.

Ideal performance is achieved when the precision and recall are both 1, i.e. the PR curve reaches the top right corner at $(1, 1)$, and when the ROC curve is far above the random detector (indicated by the dashed lines in Fig. 7), i.e. close to $(0, 1)$.

From the 'CLOSE' and 'MEDIUM' distance bins, the Baseline case, the BIV case, and the Concatenated case have comparable performances. They all have excellent precision and recall as well as low FPRs. Distinctively, the FD case has the lowest precision and recall as well as the least ideal performance according to the ROC curves. This is already expected based on the observation made in the detection timeline in Fig. 4.

For the 'DISTANT' distance bin, the precision and recall of most of the considered cases drop as shown in Fig. 6c. This is reasonable because the apparent object size, i.e. the number of pixels it occupies, is smaller when it is at this distance. Therefore, it is more challenging to be recognized and detected. Interestingly, the BIV and the Concatenated cases have higher precision than the Baseline case, especially if the detection confidence score threshold is set above 0.5. This confirms the added value of concatenating the BIV-processed frame in the object detection CNN in visually challenging cases, such as when the object of interest is far away, which is crucial for providing early warnings. The PR curve in Fig. 6c also indicates that the BIV-processed frame may help improve the recall if the detection confidence score threshold is set slightly under 0.5.



(a) 'MEDIUM'. (b) 'DISTANT'. (c) 'All'.

Figure 7: ROC curves for the considered algorithms for the object class 'DRONE' subdivided according to the labeled distance bins of the test data. The markers indicate the recall and the FPR values for the detection confidence score threshold of 0.5.

Finally, the Average Precisions (APs) which quantify the area under the PR curves are summarized in Table 4 for the considered algorithms, distance bins, and object classes. The APs when all of the distance bins are considered together are presented in the 'All' column and the mean APs (mAPs) for all of the object classes are presented in the rightmost column.

Despite the excellent performance at closer distances, the Baseline algorithm APs worsen at large sensor-to-target distances. Concatenation of the BIV-processed frames help to retain the relatively high AP. For the 'DISTANT' distance bin, the APs of the CNN trained on BIV-processed frames and BIV-processed frames concatenated with baseline frames are 0.92 and 0.88, respectively, compared to 0.83 for the CNN trained on baseline frames alone. In general, the highest APs are achieved by either the BIV case or the Concatenated case. For all of the distance bins combined, the Concatenated case has the highest AP. Considering the added value when the object of interest is at a distance, it is recommended to concatenate the BIV-processed frames to detect drones using a CNN.

Table 4: APs for the considered algorithms for the object classes ‘NDFO’ and ‘DRONE’ subdivided according to the labeled distance bins, with the mean APs (mAPs), i.e. for all classes and distance bins in the rightmost column. The values in **bold** indicate the highest AP of each distance bin.

Algorithms	AP class ‘NDFO’				AP class ‘DRONE’				mAP
	CLOSE	MEDIUM	DISTANT	All	CLOSE	MEDIUM	DISTANT	All	
Baseline	–	0.80	0.56	0.70	**0.89**	0.90	0.83	0.88	0.79
FD	–	0.25	0.20	0.22	0.87	0.80	0.55	0.73	0.48
BIV	–	0.64	0.48	0.57	**0.89**	**0.91**	**0.92**	**0.91**	0.74
Concatenated	–	**0.81**	**0.62**	**0.74**	**0.89**	**0.91**	0.88	0.89	**0.81**

5. CONCLUSIONS AND FUTURE WORK

This work proposes to employ a Bio-Inspired Vision (BIV) model of small insects to pre-process video frames for drone detection using a Convolutional Neural Network (CNN). The BIV model uses information from precedent frames to enhance the drone-to-background contrast and embody the drone’s recent trajectory in the input frames. The performance of a CNN trained on the concatenation of BIV-enhanced and still frames is compared to that of CNNs trained on still frames and frame difference alone. Results show that, at close to medium sensor-to-target distances, the CNN trained on the concatenation of the BIV-enhanced frames and still frames has comparable performance to that trained on still frames only. The added benefit of concatenating the BIV-enhanced frame becomes clear at high sensor-to-target distance which is critical for providing early warnings, i.e. the apparent target size is small. In this case, the APs of the CNN trained on BIV-processed frames and BIV-processed frames concatenated with baseline frames are 0.92 and 0.88, respectively, compared to 0.83 for the CNN trained on baseline frames alone. On average, the AP of the proposed algorithm is higher than that of the baseline one. It is therefore recommended to concatenate the BIV-processed frames to detect drones using a CNN.

The next steps include an addition of Long Short-Term Memory (LSTM) to further exploit temporal data and optimization of the BIV operating parameters. The additional steps will be done carefully to preserve a low processing time. This is done to ensure the applicability of the proposed method in real-time applications.

ACKNOWLEDGMENTS

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. The authors also would like to thank Fredrik Svanström for answering questions regarding the dataset.

REFERENCES

[1] Miasnikov, E., “Threat of terrorism using unmanned aerial vehicles: technical aspects,” tech. rep., Moscow, Russia: Center for Arms Control, Energy, and Environmental Studies, Moscow Institute of Physics and Technology (2005).

[2] Daily Mail Reporter, “Attack of the drones: The amateur enthusiasts crowding the sky with miniature stealth planes like the CIA’s.” MailOnline, 2 April 2012 https://www.dailymail.co.uk/news/article-2123813. (Accessed: 10 August 2023).

[3] BBC, “Gatwick drone shutdown.” BBC News https://www.bbc.com/news/topics/cnx1xjxwp51t. (Accessed: 10 August 2023).

[4] Jian, M., Lu, Z., and Chen, V. C., “Drone detection and tracking based on phase-interferometric Doppler radar,” in [*2018 IEEE Radar Conference (RadarConf18)*], 1146–1149 (2018).

[5] Liu, Y., Wan, X., Tang, H., Yi, J., Cheng, Y., and Zhang, X., “Digital television based passive bistatic radar system for drone detection,” in [*2017 IEEE Radar Conference (RadarConf)*], 1493–1497 (2017).

[6] Nguyen, P., Ravindranatha, M., Nguyen, A., Han, R., and Vu, T., “Investigating cost-effective RF-based detection of drones,” in [*Proceedings of the 2nd workshop on micro aerial vehicle networks, systems, and applications for civilian use*], 17–22 (2016).

[7] Al-Emadi, S. and Al-Senaid, F., "Drone detection approach based on radio-frequency using convolutional neural network," in [*2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*], 29–34, IEEE (2020).

[8] Busset, J., Perrodin, F., Wellig, P., Ott, B., Heutschi, K., Rühl, T., and Nussbaumer, T., "Detection and tracking of drones using advanced acoustic cameras," in [*Unmanned/Unattended Sensors and Sensor Networks XI; and Advanced Free-Space Optical Communication Techniques and Applications*], **9647**, 53–60, SPIE (2015).

[9] Bernardini, A., Mangiatordi, F., Pallotti, E., and Capodiferro, L., "Drone detection by acoustic signature identification," *Electronic Imaging* **2017**(10), 60–64 (2017).

[10] Park, S., Kim, H. T., Lee, S., Joo, H., and Kim, H., "Survey on anti-drone systems: Components, designs, and challenges," *IEEE Access* **9**, 42635–42659 (2021).

[11] Chiper, F.-L., Martian, A., Vladeanu, C., Marghescu, I., Craciunescu, R., and Fratu, O., "Drone detection and defense systems: Survey and a software-defined radio-based solution," *Sensors* **22**, 1453 (2 2022).

[12] Guvenc, I., Koohifar, F., Singh, S., Sichitiu, M. L., and Matolak, D., "Detection, tracking, and interdiction for amateur drones," *IEEE Communications Magazine* **56**, 75–81 (4 2018).

[13] Wisniewski, M., Rana, Z. A., and Petrunin, I., "Drone model classification using convolutional neural network trained on synthetic data," *Journal of Imaging* **8**, 218 (8 2022).

[14] Samadzadegan, F., Javan, F. D., Mahini, F. A., and Gholamshahi, M., "Detection and recognition of drones based on a deep convolutional neural network using visible imagery," *Aerospace* **9** (2022).

[15] Elsayed, M., Reda, M., Mashaly, A. S., and Amein, A. S., "Review on real-time drone detection based on visual band electro-optical (EO) sensor," in [*2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS)*], 57–65, IEEE (12 2021).

[16] Svanström, F., Englund, C., and Alonso-Fernandez, F., "Real-time drone detection and tracking with visible, thermal and acoustic sensors," in [*2020 25th International Conference on Pattern Recognition (ICPR)*], 7265–7272, IEEE (1 2021).

[17] Trapal, D. D. C., Leong, B. C. C., Ng, H. W., Zhong, J. T. G., Srigrarom, S., and Chan, T. H., "Improvement of vision-based drone detection and tracking by removing cluttered background, shadow and water reflection with super resolution," in [*2021 6th International Conference on Control and Robotics Engineering (ICCRE)*], 162–168, IEEE (4 2021).

[18] Thai, V.-P., Zhong, W., Pham, T., Alam, S., and Duong, V., "Detection, tracking and classification of aircraft and drones in digital towers using machine learning on motion patterns," in [*2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*], 1–8, IEEE (4 2019).

[19] van Hateren, J. and Snippe, H., "Information theoretical evaluation of parametric models of gain control in blowfly photoreceptor cells," *Vision Research* **41**, 1851–1865 (6 2001).

[20] Uzair, M., Brinkworth, R. S., and Finn, A., "Bio-inspired video enhancement for small moving target detection," *IEEE Transactions on Image Processing* **30**, 1232–1244 (2021).

[21] Uzair, M., Brinkworth, R. S., and Finn, A., "A bio-inspired spatiotemporal contrast operator for small and low-heat-signature target detection in infrared imagery," *Neural Computing and Applications* **33**, 7311–7324 (7 2021).

[22] Shu, Y., Sui, Y., Zhao, S., Cheng, Z., and Liu, W., "Small moving object detection and tracking based on event signals," *2021 7th International Conference on Computer and Communications (ICCC)* , 792–796, IEEE (12 2021).

[23] Kwan, C. and Larkin, J., "Detection of small moving objects in long range infrared videos from a change detection perspective," *Photonics* **8**, 394 (9 2021).

[24] Zheng, Y., Zheng, C., Zhang, X., Chen, F., Chen, Z., and Zhao, S., "Detection, localization, and tracking of multiple MAVs with panoramic stereo camera networks," *IEEE Transactions on Automation Science and Engineering* , 1–18 (2022).

[25] Xie, J., Yu, J., Wu, J., Shi, Z., and Chen, J., "Adaptive switching spatial-temporal fusion detection for remote flying drones," *IEEE Transactions on Vehicular Technology* **69**, 6964–6976 (7 2020).

[26] Carrio, A., Lin, Y., Saripalli, S., and Campoy, P., "Obstacle detection system for small uavs using ads-b and thermal imaging," *Journal of Intelligent Robotic Systems* **88**, 583–595 (12 2017).

[27] Seidaliyeva, U., Akhmetov, D., Ilipbayeva, L., and Matson, E. T., "Real-time and accurate drone detection in a video with a static background," *Sensors* **20**, 3856 (7 2020).

[28] Wang, H., Zhong, Z., Lei, F., Jing, X., Peng, J., and Yue, S., "Spatio-temporal feedback control of small target motion detection visual system," *arXiv preprint arXiv:2211.10128* (11 2022).

[29] Sangam, T., Dave, I. R., Sultani, W., and Shah, M., "Transvisdrone: Spatio-temporal transformer for vision-based drone-to-drone detection in aerial videos," in [*2023 IEEE International Conference on Robotics and Automation (ICRA)*], 6006–6013, IEEE (10 2023).

[30] Fang, J., Finn, A., Wyber, R., and Brinkworth, R. S. A., "Acoustic detection of unmanned aerial vehicles using biologically inspired vision processing," *The Journal of the Acoustical Society of America* **151**, 968–981 (2 2022).

[31] Nalamati, M., Kapoor, A., Saqib, M., Sharma, N., and Blumenstein, M., "Drone detection in long-range surveillance videos," in [*2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*], 1–6, IEEE (9 2019).

[32] Ashraf, M. W., Sultani, W., and Shah, M., "Dogfight: Detecting drones from drones videos," in [*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*], 7067–7076 (3 2021).

[33] Park, J., Kim, D. H., Shin, Y. S., and Lee, S., "A comparison of convolutional object detectors for real-time drone tracking using a PTZ camera," in [*2017 17th International Conference on Control, Automation and Systems (ICCAS)*], 696–699, IEEE (10 2017).

[34] Delleji, T., Slimeni, F., Fekih, H., Jarray, A., Boughanmi, W., Kallel, A., and Chtourou, Z., "An upgraded-yolo with object augmentation: Mini-uav detection under low-visibility conditions by improving deep neural networks," *Operations Research Forum* **3**, 60 (9 2022).

[35] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., "SSD: Single shot multibox detector," in [*Computer Vision–ECCV 2016: 14th European Conference*], 21–37, Springer (12 2016).

[36] Elsayed, M., Mashaly, A. S., Reda, M., and Amein, A. S., "Visual drone detection in static complex environment," in [*2022 13th International Conference on Electrical Engineering (ICEENG)*], 154–158, IEEE (3 2022).

[37] Svanström, F., Alonso-Fernandez, F., and Englund, C., "A dataset for multi-sensor drone detection," *Data in Brief* **39**, 107521 (12 2021).

[38] Coluccia, A., Fascista, A., Schumann, A., Sommer, L., Dimou, A., Zarpalas, D., Akyon, F. C., Eryuksel, O., Ozfuttu, K. A., Altinuc, S. O., Dadboud, F., Patel, V., Mehta, V., Bolic, M., and Mantegh, I., "Drone-vs-bird detection challenge at IEEE AVSS2021," in [*2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*], 1–8, IEEE (11 2021).

[39] Li, J., Ye, D. H., Chung, T., Kolsch, M., Wachs, J., and Bouman, C., "Multi-target detection and tracking from a single camera in unmanned aerial vehicles (UAVs)," in [*2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*], 4992–4997, IEEE (10 2016).

[40] Rozantsev, A., Lepetit, V., and Fua, P., "Detecting flying objects using a single moving camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**, 879–892 (5 2017).

[41] AIcrowd, "Airborne object tracking challenge." AICrowd https://www.aicrowd.com/challenges/airborne-object-tracking-challenge. (Accessed: 10 August 2023).

[42] Fu, Z., Zhi, Y., Ji, S., and Sun, X., "Remote attacks on drones vision sensors: An empirical study," *IEEE Transactions on Dependable and Secure Computing* **19**, 3125–3135 (9 2022).

[43] Andraši, P., Radišić, T., Muštra, M., and Ivošević, J., "Night-time detection of UAVs using thermal infrared camera," *Transportation Research Procedia* **28**, 183–190 (2017).

[44] Ganti, S. R. and Kim, Y., "Implementation of detection and tracking mechanism for small UAS," *2016 International Conference on Unmanned Aircraft Systems (ICUAS)* , 1254–1260, IEEE (6 2016).

[45] Redmon, J. and Farhadi, A., "YOLO9000: Better, faster, stronger," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 7263–7271 (12 2017).

[46] Skelton, P. S., Finn, A., and Brinkworth, R. S., "Consistent estimation of rotational optical flow in real environments using a biologically-inspired vision algorithm on embedded hardware," *Image and Vision Computing* **92**, 103814 (12 2019).

[47] Naka, K. I. and Rushton, W. A. H., “S-potentials from luminosity units in the retina of fish (cyprinidae),” *The Journal of Physiology* **185**, 587–599 (8 1966).

[48] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., “MobileNetV2: Inverted residuals and linear bottlenecks,” in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 4510–4520 (1 2018).

[49] Deng, P., Wang, K., and Han, X., “Real-time object detection based on YOLO-v2 for tiny vehicle object,” *SN Computer Science* **3**, 329 (7 2022).

[50] Zhou, Y., Wen, S., Wang, D., Meng, J., Mu, J., and Irampaye, R., “MobileYOLO: Real-time object detection algorithm in autonomous driving scenarios,” *Sensors* **22**, 3349 (4 2022).

[51] Ayob, A. F., Khairuddin, K., Mustafah, Y. M., Salisa, A. R., and Kadir, K., “Analysis of pruned neural networks (MobileNetV2-YOLO v2) for underwater object detection,” in [*Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019: NUSYS’19*], 87–98 (2021).

[52] Chevalier, P., “On the specification of the DRI requirements for a standard NATO target,” *Researchgate publication* (2016).

[53] Infinitioptics, “Whitepaper on thermal DRI.” Infinitoptics https://www.infinitioptics.com/dri. (Accessed: 10 August 2023).