American Academic Scientific Research Journal for Engineering, Technology, and Sciences
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402
https://asrjetsjournal.org/index.php/American_Scientific_Journal/index

Utilize Dense Optical Flow for Small Flying Targets Detection and Tracking

Saad Alkentar[a*], Abdulkareem Assalem[b]

[a,b]*Albaath University, Homs, Syria*
[a]*Email: saad.zgm@gmail.com*
[b]*Email: assalem1@gmail.com*

Abstract

The detection of small targets remains a critical challenge within the field of image processing. Traditional techniques, such as image subtraction with frame-to-frame registration, suffer from high false alarm rates. Even state-of-the-art deep learning architectures, like YOLO and Masked R-CNN, exhibit limitations in this domain. In overextended distances, the inherent feature quality of small targets degrades significantly, leading to a scarcity of informative data for conventional detection algorithms. Consequently, accurate visual recognition becomes a particularly hard task.This work presents a novel detection approach that draws inspiration from the human visual attention mechanism. By leveraging dense optical flow, the model prioritizes moving objects within the scene, facilitating effective target detection. Furthermore, the proposed method employs K-Means clustering to achieve robust foreground-background separation based on color intensity characteristics. To address the limitations of dense optical flow with stationary targets, a dedicated tracking algorithm is also introduced. Our approach demonstrated a high level of accuracy (98%) when evaluated on unseen test data. Additionally, the algorithm functioned in real-time, enabling immediate processing.

Keywords: Small Target; Optical Flow; K-Means; Infrared Imaging; Real-time; Tracking.

1. Introduction

The accurate detection of small targets has great importance in multiple fields, affecting our safety, security, and knowledge in many ways. To list a few of its applications, we can mention early detection systems for drones and missiles, border surveillance, and anti-terrorism systems. there is no universal definition of a small target since it is often context-dependent

---

Received: *3/16/2024*
Accepted: *5/16/2024*
Published: *5/26/2024*

---

* Corresponding author.

Based on related reviews and research in this field, we can identify it by the number of pixels, any target with 2x2 up to 9x9 pixels is considered a small target, or by pixel ratio; any target occupies less than 15% of the whole image is considered small target [1]. This research investigates the detection of diminutive targets typically characterized by minimal dimensions, ranging from 2x2 to 9x9 pixels. These targets are devoid of substantial textural information, posing a significant challenge for contemporary machine learning algorithms, such as YOLO and Masked R-CNN, due to their limited capacity to effectively discern such minute objects.

This research draws inspiration from the human visual system's processing mechanisms. The human brain prioritizes the processing of motion and color variations within the visual field. This prioritization facilitates the detection of even the smallest targets based on their movement, followed by their recognition through the analysis of color and saturation differences compared to the background. Drawing an analogy to the human visual system's ability to detect a moving ant on the kitchen floor, we propose a system inspired by these biological principles. This system leverages dense optical flow to identify small targets based on their pixel-wise motion. Subsequently, K-Means clustering is employed to refine the candidate regions derived from the optical flow analysis. This filtering step allows us to isolate potential targets from background movements, such as those caused by swaying tree leaves or camera jitter. Within the context of this study, we will

- Share two new datasets for small target detection and recognition. Section 3
- Propose a novel small target detection system with two recognition approaches for both visual and thermal images. Section 4
- Proposes a powerful tracking algorithm that suits our targets' characteristics and the proposed detection system. Section 5
- Compare the proposed algorithm with YOLO versions 8 and 9. Section 6

To facilitate reproducibility and knowledge sharing, the code for the proposed detection algorithm and its associated network structure is publicly available. Additionally, the code used for training the YOLO model, along with the corresponding datasets, has been made accessible through an online repository.

2. Related work

Target detection is the process of finding objects of interest in images. this field has gained a lot of interest in the recent decade with the development of computer vision technologies. Many algorithms have been proposed. Convolutional neural networks (CNNs) have been used in many algorithms like SSD, YOLO, R-CNN, and faster RCNN [2, 3, 4, 5, 6, 7, 8, 9]. Lawal [10] proposed a modification to YOLO v3 to detect tomatoes in complex environments. Wu and his colleagues. [11] proposed a multiple-scaled Faster R-CNN-based face detection algorithm for small face detection. Shakarami and his colleagues. [12] used YOLO v3 for blood cell recognition. Shi and his colleagues. [13] used YOLO v4 for oil quality detection.

Small target detection is a challenging task for several reasons, to list a few: 1) Images are relatively large compared to the target size. 2) The target background could be complex and fuzzy. 3) Small targets with few pixels usually do not have enough feature information for classic detection algorithms.

A lot of methods have been developed to address small target scenarios. For instance, Lu and his colleagues. Reference [14] proposes a new object detector for remote sensing images called SAFF-SSD (Self-Attention Combined Feature Fusion-based SSD). It uses a modified EfficientNetV2-S as the backbone for feature extraction. A detection neck called CSP-PAN was employed for fusing features from different levels. The paper also proposed a new loss function using the Normalized Wasserstein Distance (NWD) instead of the commonly used Intersection over Union (IoU) for evaluating bounding boxes. An optimized YOLO v3 for detecting objects of various sizes in remote sensing imagery was proposed in [15]. To toggle objects with various sizes that clutter together, the authors replaced the feature extraction component of YOLO-V3 with DenseNet and increased the object detection scale for YOLO to 4. Many researchers explored data augmentation techniques, contextual information, and multi-scale feature learning to enhance small target detection algorithms' performance. Kisantal and his colleagues. Reference [16] proposed data augmentation method which involves randomly duplicating small targets in an image, but this resulted in scale and background mismatch issues. to overcome those issues, Chen and his colleagues. Reference [17] proposed AdaReasampling method to address background issues, and Yu and his colleagues. Reference [18] proposed the "Scale Match" method to address scaling issues. contextual information refers to the relationship between the target and background pixels, Zhao and his colleagues. Reference [19] proposed SODet network backbone that uses the attention principle [20] to detect connections between objects in images with distant targets, SODet combines both global and local image features in an adaptive way for better small object detection. it uses an adaptive fusion strategy that assigns weights to these features based on the image content. This approach aims to give more importance to features that are more informative for detecting small objects in a particular image. Cao and his colleagues. Reference [21] proposed a Feature-Fused SSD algorithm to reconstruct the image back into pixel space through deconvolution. This algorithm enhanced the connection between contexts and improved the detection accuracy for small targets. Lim and his colleagues. Reference [22] proposed a model that incorporates features from different layers of a neural network to provide context. This context helps the model understand the surrounding area of the potential object, aiding in its identification. The paper proposes using a method called feature concatenation to combine information from various layers. It also employs an attention mechanism to focus on the most relevant parts of the image. This mechanism helps prioritize the features that are most likely to belong to the small object, filtering out background noise.

In this study, YOLO was chosen as a reference for small target detection. YOLO, which stands for "You Only Look Once", is a real-time object detection algorithm. Unlike some object detection methods that analyze images in stages, YOLO is a single-stage detector. It processes the entire image at once, making it faster for real-time applications. It relies on CNNs, which are artificial neural networks particularly adept at image recognition. By analyzing the image through a series of filters, CNN can identify objects and their locations within the image. predicts bounding boxes around detected objects in the image. These boxes indicate the location and size of the object. Additionally, YOLO assigns a probability score to each bounding box, indicating the confidence level that the identified region contains a specific object class (e.g., caries, implant, filling). We have trained YOLOv7 [23], YOLOv8 [24], and YOLOv9 [25] on the proposed dataset to evaluate our model performance against other modern detection algorithms.

3. Materials

Figure 1: A few samples from the proposed small target detection image dataset, the first raw features optical camera images while the second one features thermal camera images

There aren't many datasets that suit small target detection in both infrared and optical images, and due to the use of dense optical flow, we cannot evaluate the proposed algorithm using individual images similar to classic detection algorithms. Svanström and his colleagues. [26] shared a dataset of 10-second videos for four target classes: airplanes, drones, birds, and helicopters. They annotated videos to close, medium, and distant videos but didn't provide further annotation to target position in frames. To evaluate the proposed algorithm, we ignored close and medium targets and used the distanced videos only. To compare the proposed algorithm with YOLO, we are sharing the "Small Target DS YOLO[1]" dataset. In which we have annotated 2027 IR and 2110 V frames with YOLO v5 annotations format. Figure 1 presents a few samples from the proposed DS after drawing target boxes.To build our classifier, we extracted dense optical flow proposed areas, reshaped them to 20x20 pixels, and sorted them in image pairs (expected target area in two consecutive frames) into target and noise classes. The dataset is shared as "Small Target Image Pairs Dataset[2]". Total number of image pairs is 6303. Figure 2 presents a few samples from the proposed image pairs DS.

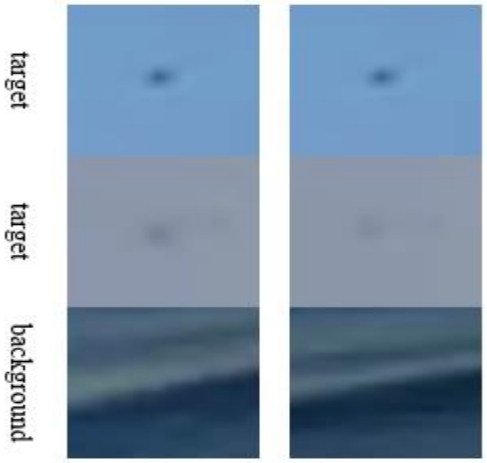

Figure 2: A few samples from the proposed small target image pairs dataset

[1] https://www.kaggle.com/datasets/saadkentar/small-target-ds-yolo
[2] https://www.kaggle.com/datasets/saadkentar/small-target-image-pairs-dataset

4. The Proposed Detection Algorithm

The proposed algorithm consists of three stages, Figure 3 shows a block diagram of the proposed algorithm. In the first stage (first row of Figure 3), we calculate the dense optical flow of two consecutive frames. If we are to express pixel intensity as a function I(x, y, t), where (x, y) is the pixel location and t is the time, a consecutive frame over a short period (dt) should have the same intensity values resulting in

$$I(x, y, t) = I'(x + dx, y + dy, t + dt) \quad (1)$$

Where I is the value in (t), and I' is the value at $(t + dt)$. By using Taylor Series Approximation and removing the common terms we can reach

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt + \cdots$$

$$\rightarrow \frac{\partial I}{\partial x}dx + \frac{\partial I}{\partial y}dy + \frac{\partial I}{\partial t}dt = 0 \rightarrow \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0 : \; u = \frac{dx}{dt}, v = \frac{dy}{dt} \quad (2)$$

$\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are the image gradients along the horizontal axis, the vertical axis, and time. solving those equations determines the movement over time. We have used Farneback's dense optical flow algorithm due to its accuracy and efficiency. It constructs a pyramid of coarsened images to handle large displacements. Within each level, local polynomial expansions capture intensity variations between corresponding pixels in consecutive frames. The algorithm optimizes these polynomials to minimize the difference between predicted and actual intensity changes under motion, making it a mainstay in computer vision tasks. The outcome of dense optical flow is restructured as an image with the same dimensions as the input frames in HSV (Hue, Saturation, and Value) color space. channel H data represents the motion direction for each pixel while channel V data represents the speed.

In the second stage, H or V channel data are processed separately based on the camera movement state. Camera movement state c(t) can be estimated based on V channel data. Moving the camera accumulated speed to all pixels, resulting in high speed-data sum

$$c(t) = \begin{cases} 1 \; if \; sum(V) > T_v \\ 0 \; if \; sum(V) < T_v \end{cases} \quad (3)$$

Where T_v is a threshold chosen based on the camera scene background. For a static or slow-moving camera, it is easier to detect potential targets using V channel data since target speed can reveal its location. But in the case of a moving camera, most pixels will have the camera speed making V channel data hard to use.

$$d(t) = \begin{cases} p_v(V) \; if \; c = 1 : \; p_v = HPF \\ p_h(H) \; if \; c = 0 : \; p_h = PSF \end{cases} \quad (4)$$

Therefore, using H channel data would be easier since the target moving direction will most likely differ from

the camera moving direction. A High-Pass Filter (HPF) was used as p_v to process V channel data, and only pass pixel data with high speed. Processing H channel data requires a Band-Stop Filter (PSF) p_h to suppress the camera's moving direction data and only keep target direction data. This stage is illustrated in row 2 of Figure 3.

The third stage should filter the areas proposed by previous stages to potential targets or background (noise). Two approaches were explored for this stage. The first one is based on image clustering. We begin by applying a custom convolutional filter (CF) to the proposed image area slice (S) to improve target features

$$S' = S * CF : CF = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix} \quad (5)$$

Where * indicates convolution, S' is the processing image slice, and the value of CF was chosen based on multiple experiments on different filters. Then we apply a clustering algorithm to make the foreground-background separation. Clustering is a method to divide a set of data into a specific number of groups. There are many types of unsupervised clustering algorithms: K-Means clustering, Fuzzy C-means clustering, mountain clustering method, and subtractive clustering method. K-Means was chosen for this study due to its computational effectiveness and high speed. K-Means [27] is an iterative algorithm in which it minimizes the sum of distances from each object to its cluster centroid, over all clusters. To cluster data using K-Means we

1. initialize cluster centers for the k^{th} cluster
2. calculate Euclidean distance between image pixels and centers using

$$d = \|p(x, y) - c_k\| \quad (6)$$

Where $p(x, y)$ is the input pixels, c_k is the center of the k^{th} cluster

3. assign pixels to the nearest center based on the value of d
4. after assigning all pixels to the nearest centers, recalculate the new center position using

$$c_k = \frac{1}{k} \sum_{y \in c_k} \sum_{x \in c_k} p(x, y) \quad (7)$$

5. repeat the process until it satisfies the tolerance or accepted error value
6. reshape the clustered pixels into an image

Since we are trying to make a foreground-background separation, a value of 2 was chosen for k. After applying K-Means, we can make target-noise decisions based on background self-coherence since we expect to have a small target surrounded by a homogeneous background.

in the second approach, after applying the same custom convolutional filter to the image slice. We pass the processed image slice S' to a custom neural network structure trained to make the target-noise separation. The network structure is illustrated in Table 1.

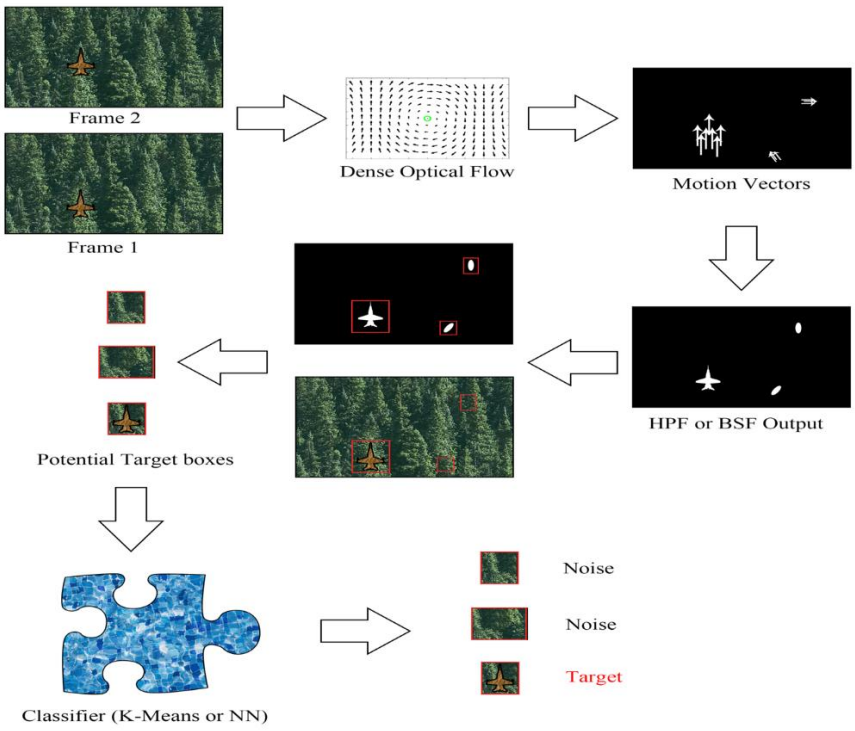

Figure 3: Proposed detector block diagram

Two other structures were proposed. The first uses 3D convolutions to improve the target feature resolution from two successive slices, and the other uses LSTM to extract temporal target information from two successive slices. However, since two successive slices tend to have similar features, those proposed structures tend to have similar performance to the single slice structure. Therefore, we didn't include them in the practical results section, yet we believe they could be further improved in future work.

Table 1: Proposed classifier structure for target-noise separation

Layer	Output shape	Params #
Convolutional 2D	(20, 20, 32)	896
Max Pooling 2D	(10, 10, 32)	0
Convolutional 2D	(10, 10, 64)	18,496
Max Pooling 2D	(5, 5, 64)	0
Batch Normalization	(5, 5, 64)	256
Global Max Pooling 2D	(64)	0
Dropout	(64)	0
Dense	(2)	130

The proposed algorithm pays more attention (assigns importance) to the moving pixels in frames which eliminates the need to process the whole image using a sliding window, improves the detection accuracy for small targets in cluttered backgrounds, and minimizes the detection processing time.

5. The proposed tracking algorithm

Dense optical flow is designed to detect the smallest movement in consecutive frames. Yet it doesn't fit stationary targets and can easily lose track of them. To overcome this disadvantage, we propose a tracking algorithm that keeps track of the last target position using the same K-Means approach mentioned in section IV to decide whether to keep or neglect the proposed target position for tracking. The proposed tracking algorithm can be summed in those three steps

1. Initialize target boxes from dense optical flow output and assign IDs to the new targets.
2. In the next frame, assign boxes to IDs based on the distance between the new proposed boxes and the old ones.
3. check target existence in old positions without new feed to overcome optical flow stationary targets disadvantage. If a target exists, self-feed the old box to the same track ID.

This tracking algorithm improves the proposed detection algorithm tolerance to stationary targets and enhances tracking and recognition performance for the proposed algorithm.

6. Practical results

We employed transfer learning to train YOLO versions 7, 8, and 9 for small object detection on the proposed dataset which combined visual and thermal images. The training process utilized 40 epochs to achieve results in Table 2, training plots and confusion matrixes for the three algorithms can be found in Appendix A

Table 2: YOLO versions 7, 8, and 9 validation metrics

Algorithm	Precision %	Recall %	mAP@0.5 %
YOLO v7	55.6	51.3	47.7
YOLO v8	95.4	96.6	99.0
YOLO v9	93.7	94.1	98.6

Analysis of the validation metrics revealed that YOLO v7 exhibited limitations in detecting targets with low contrast and quality. Conversely, YOLO v8 and v9 demonstrated significantly higher potential for success in this application. Consequently, YOLO v7 was excluded from subsequent evaluations.

The proposed classifier model was trained from scratch with 20x20 pixel images to distinguish targets from the background. An accuracy of 98% was achieved after 15 epochs of training. Figures 3 and 4 depict the corresponding accuracy and loss plots, respectively

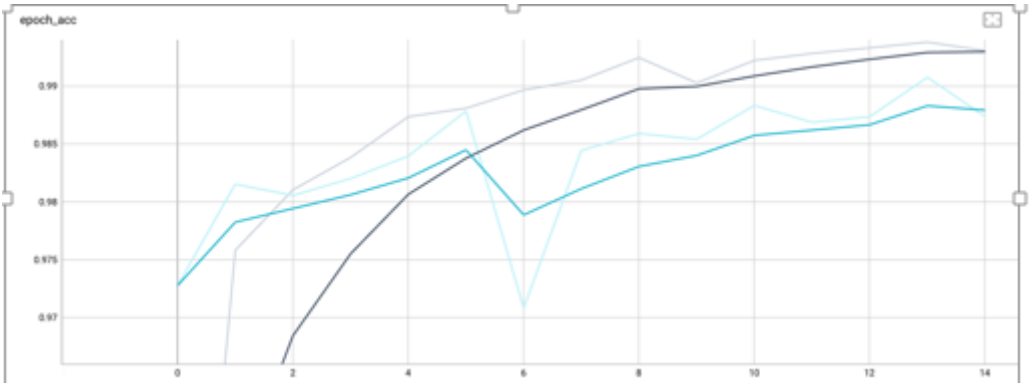

Figure 4: Proposed target-noise classifier model accuracy per epoch plot

To assess the generalizability of the proposed models, we employed unseen video data featuring drones for evaluation. These videos encompassed diverse background conditions, including clear skies, cloudscapes, and wooded areas. The target set comprised two categories: drones and birds. Notably, drones traversed across all backgrounds, while birds were confined to the wooded areas. To quantitatively gauge the models' performance, we calculated precision and recall metrics based on equations 8 and 9.

$$P = \frac{TP}{TP+FP} \quad (8)$$

$$R = \frac{TP}{TP+FN} \quad (9)$$

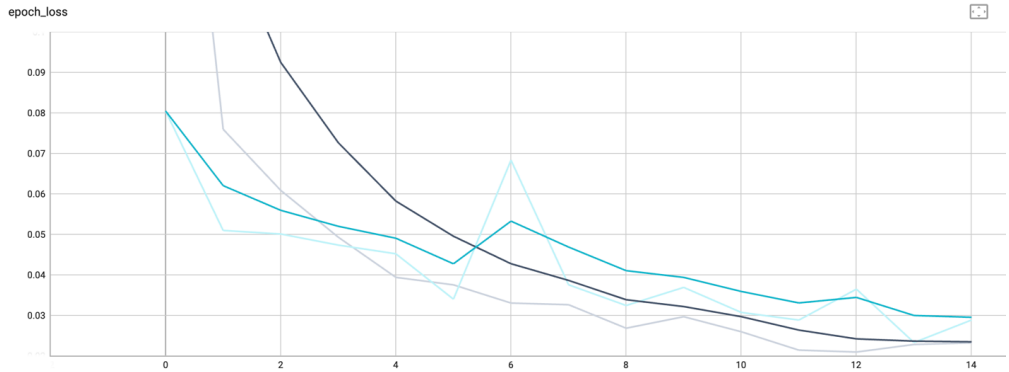

Figure 5: Proposed target-noise classifier model loss per epoch plot

A True Positive (TP) is achieved when there's a spatial overlap between the bounding box predicted by the algorithm and the ground truth location of the target object. Conversely, a False Negative (FN) occurs when the algorithm fails to detect an actual target present within the frame. Lastly, a False Positive (FP) arises when the algorithm erroneously predicts a target object in the frame that is demonstrably absent. Table 3 presents the test results.

The first row of Table 3 shows that YOLOv8 prioritizes speed (33 fps) at the expense of accuracy compared to other algorithms. This trade-off suggests a potentially less complex model or training data skewed towards simpler scenes. However, YOLOv8 demonstrates strength in controlled environments with clear skies or thermal imagery. Its limitations in handling complex visual elements like clutter or varying illumination

necessitate further investigation for broader applicability. On the other hand, YOLOv9 exhibits significant improvements in both precision and recall, demonstrating its ability to handle small targets across thermal and visual imagery. However, this enhanced accuracy comes at a cost. The model generates a higher rate of false positives, requiring further refinement. Additionally, its computational demands far exceed those of v8, rendering it unsuitable for resource-constrained devices. This highlights the need to explore techniques for optimizing YOLOv9's efficiency while maintaining its superior detection capabilities.

Table 3: Proposed models comparison with YOLO

Algorithm	Precession %	Recall %	Time (ms/frame)
YOLO v8	42.52	37.21	30
YOLO v9	87.81	82.96	580
Ours (K-Means)	95.70	59.58	20
Ours (NN)	98.26	67.67	40

Our proposed algorithms achieve remarkable precision, surpassing YOLO in minimizing false positives. This signifies their strength in accurately distinguishing true objects from background clutter, leading to more reliable detections.

The proposed two approaches reinforce the trade-off between model complexity and accuracy. While the K-Means approach offers real-time processing suitable for resource-constrained devices, its lower recognition rate compared to neural networks highlights its limitations. This emphasizes the ongoing need for lightweight, high-accuracy models that can bridge the gap between computational efficiency and robust object detection, particularly for real-world applications on devices with limited resources.

Our lightweight neural network approach for target-background discrimination demonstrates a promising trade-off between accuracy and efficiency. Compared to the K-Means-based approach, it achieves a notable 3% improvement in precision and a 7% improvement in recall. This signifies a significant reduction in false positives and a better ability to capture true targets. Importantly, these enhancements are achieved while maintaining real-time processing constraints, making the model suitable for deployment on resource-limited devices. This finding highlights the potential of lightweight neural networks for real-world applications where both high accuracy and fast processing are crucial.

Our proposed neural network approach achieves impressive accuracy; however, a current limitation lies in its recall compared to YOLO v9. The test videos included a significant number of initially stationary targets, leading to a 14% higher recall rate for YOLO v9. This disparity highlights a key difference in the approaches: our method utilizes optical flow, which inherently misses stationary objects at the beginning of recordings. While real-world surveillance systems likely involve fewer initially stationary targets (especially for flying objects), this finding underscores the need for further exploration. Future work should investigate incorporating complementary methods to address this limitation.

7. Conclusion

This study proposed a three-stage algorithm for small target detection. Simulating human attention mechanisms and giving more weight (importance) to moving areas produced encouraging results in both precession and recall. This approach can be further improved with the use of small fast ML models like the one used for recognition. The main drawback of optical flow is depending solely on movement and not paying stationary targets much attention. To overcome this drawback, a special tracking algorithm was proposed, that continues tracking stationary targets after the first detection. Future work can focus on some potential avenues such as:

- Target appearance modeling: Integrating an appearance-based object detection module alongside the optical flow would allow the algorithm to identify stationary objects even if they haven't moved yet.
- Background subtraction: Implementing background subtraction techniques could help identify initially stationary objects by detecting deviations from the background model.
- Hybrid approaches: Combining optical flow with other motion detection techniques, like frame differencing, could offer a more comprehensive solution for capturing both moving and initially stationary objects.

By exploring these strategies, future research can aim to bridge the current gap in recall and achieve robust object detection across various targets, especially in real-world surveillance applications. both the detector and tracker have room to grow in both speed and accuracy and can be worked on independently.

Appendix A

Training plots and confusion matrixes for YOLO models

YOLO v7

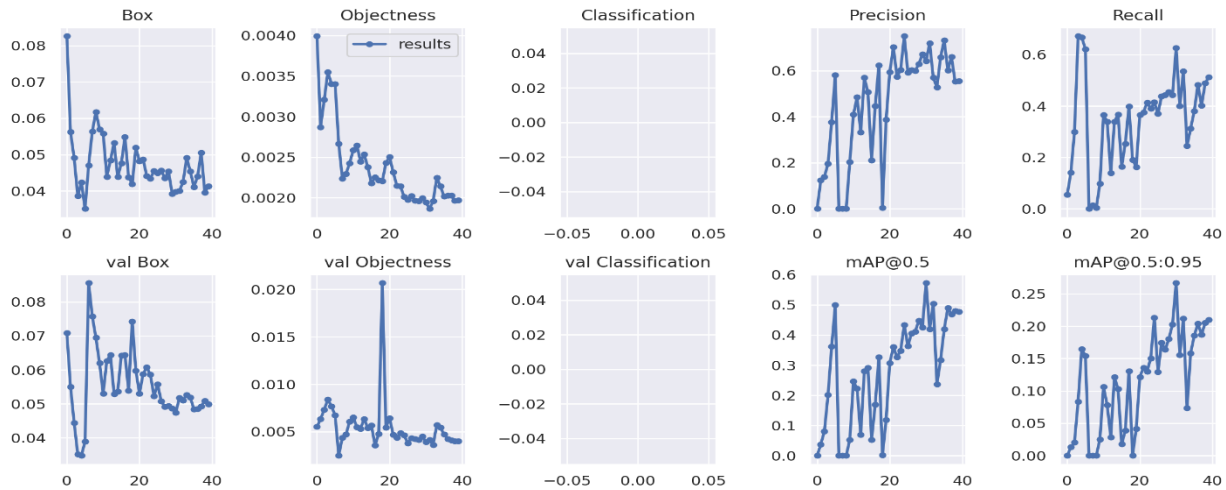

Figure 6

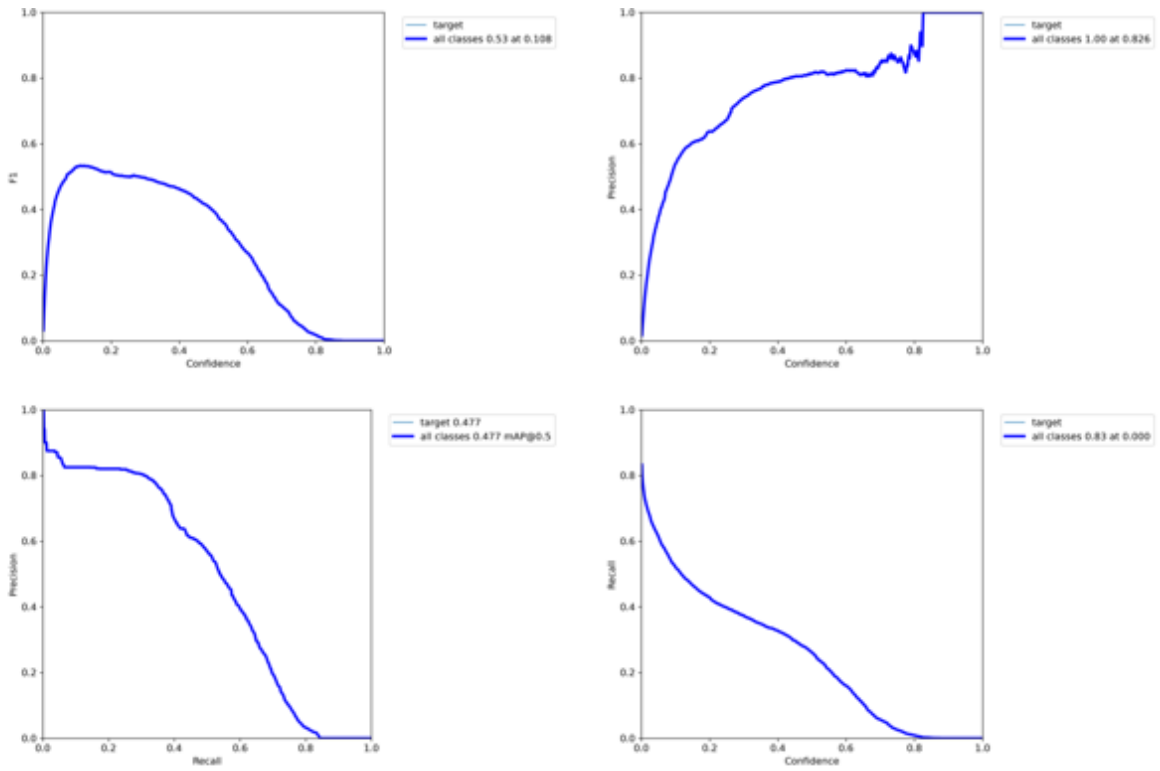

Figure 7

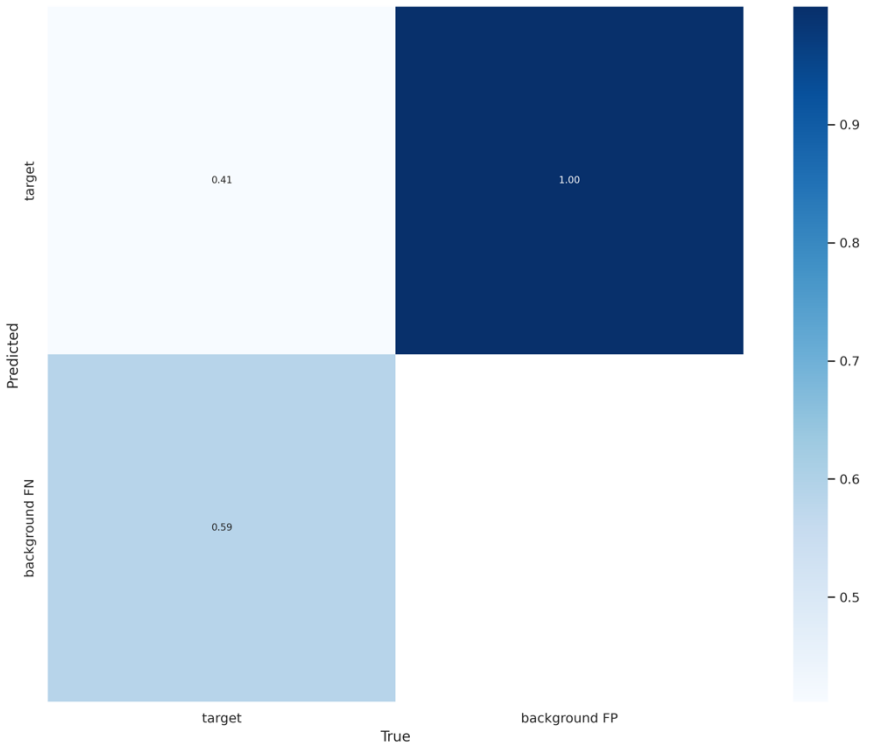

Figure 8

YOLO v9

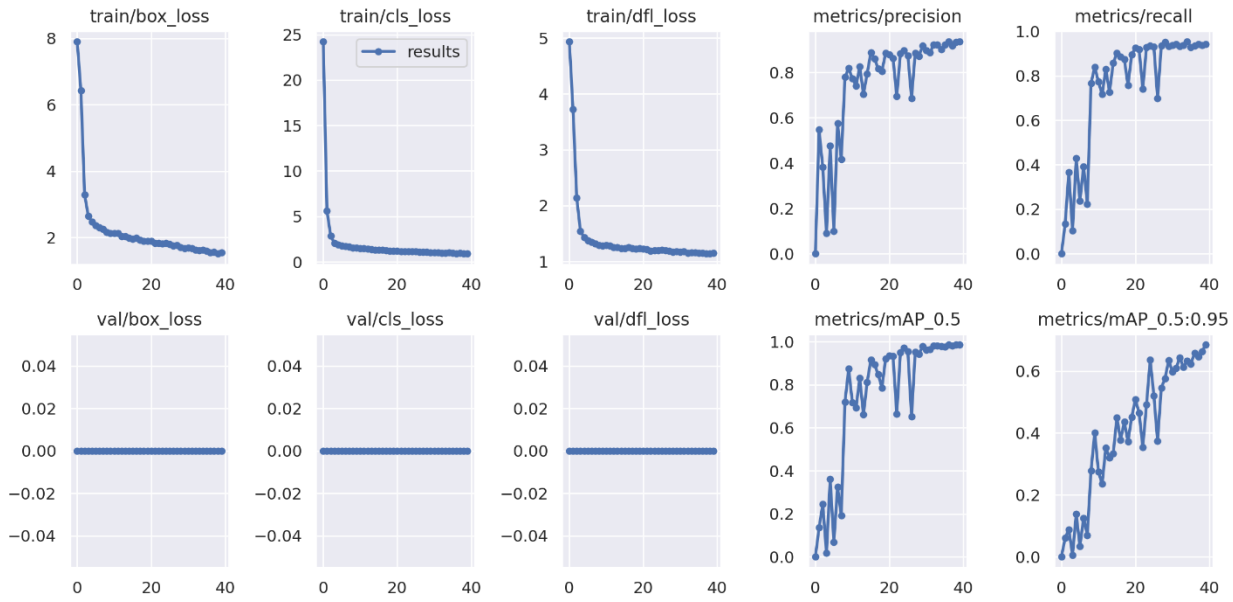

Figure 9

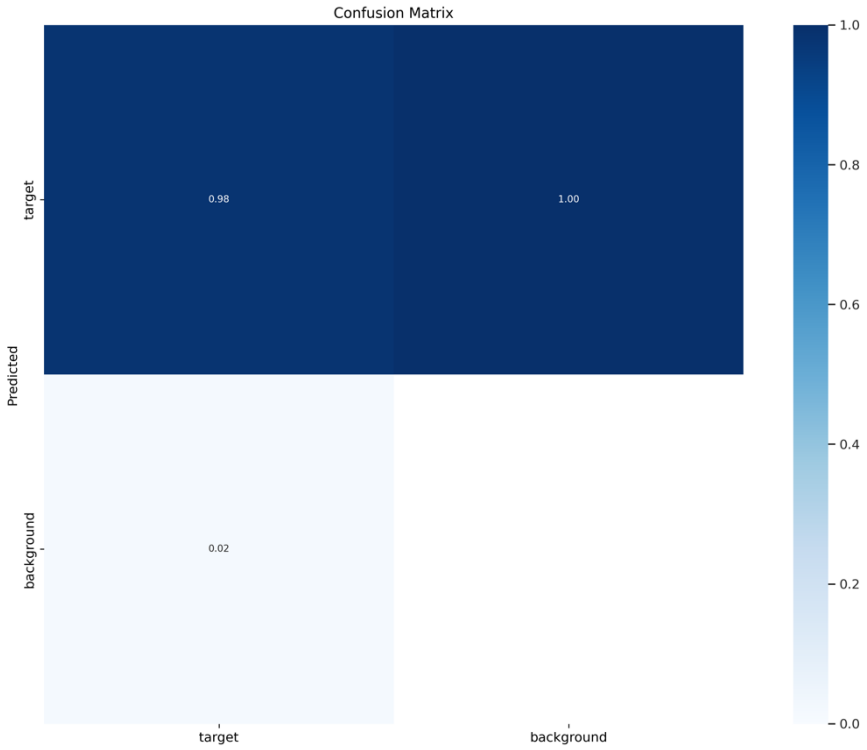

Figure 10

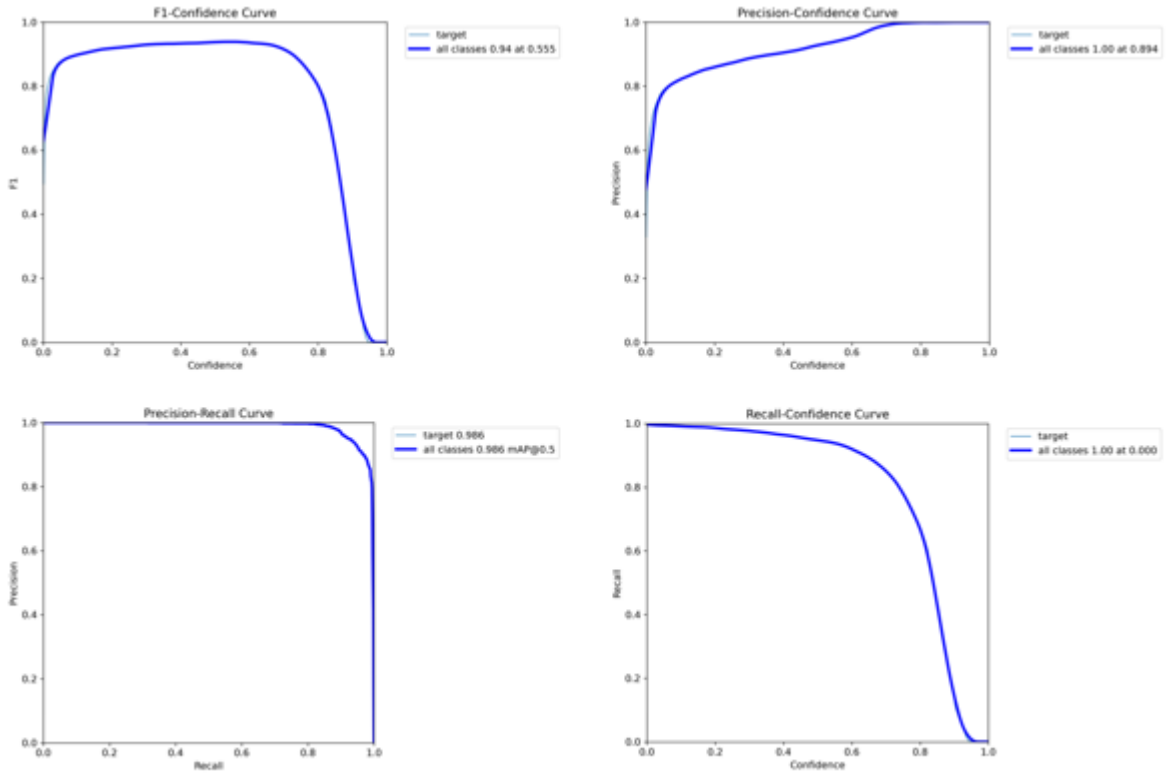

Figure 11

YOLO v8

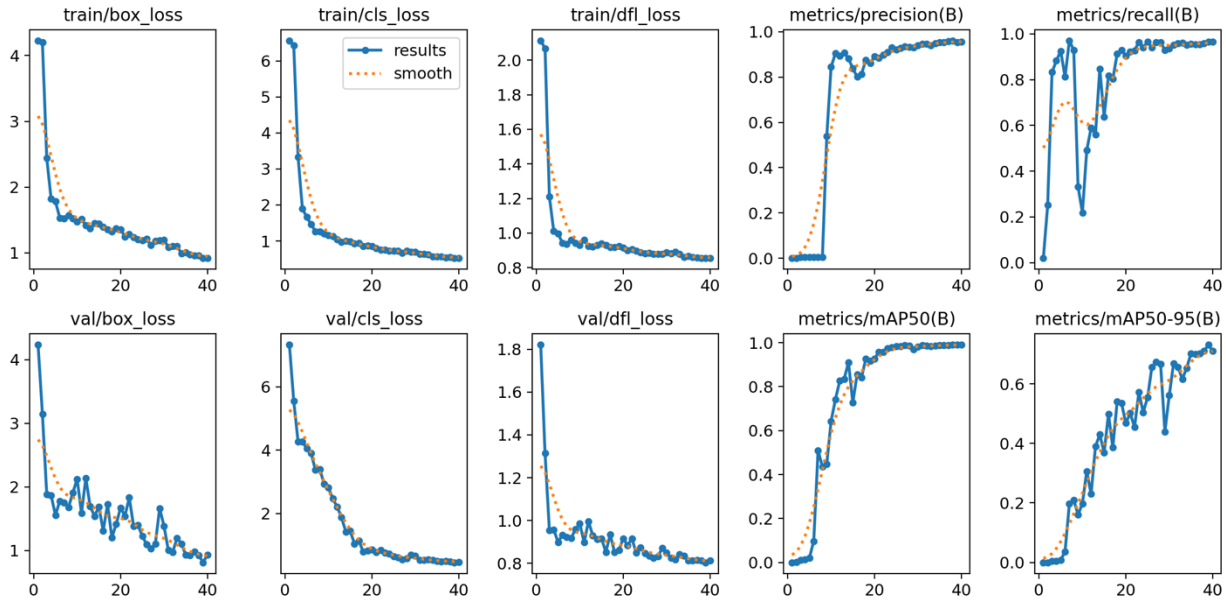

Figure 12

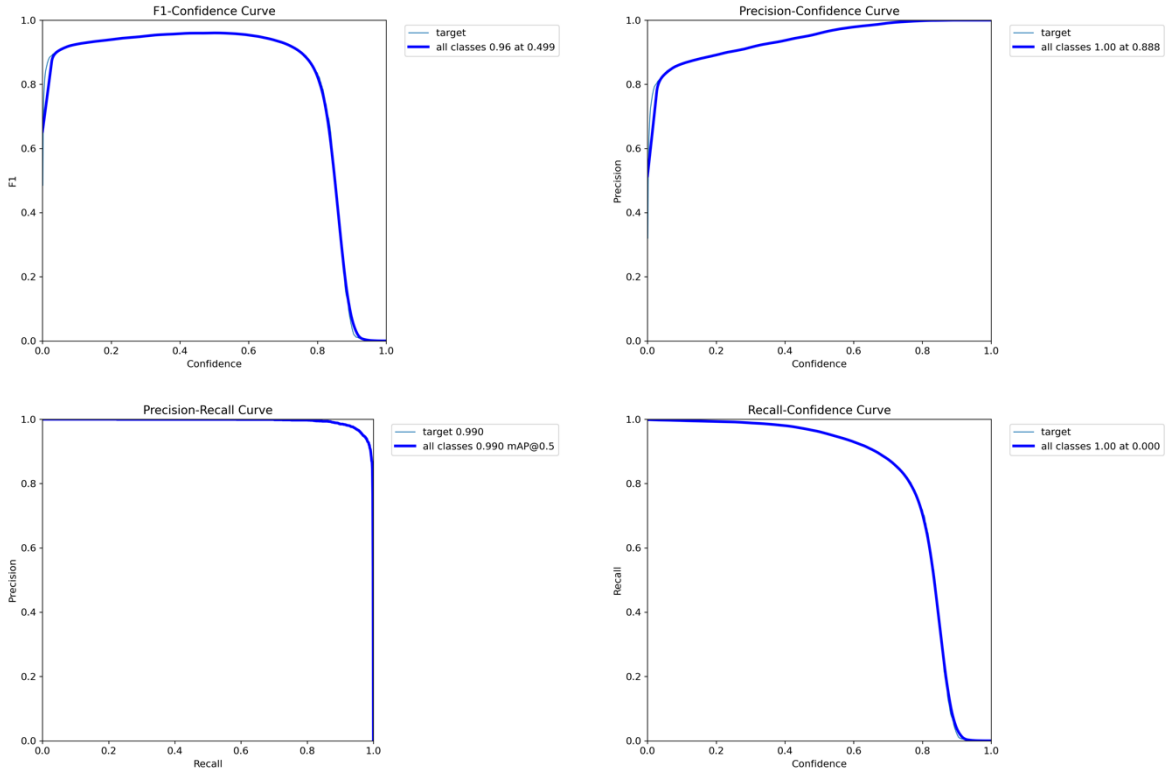

Figure 13

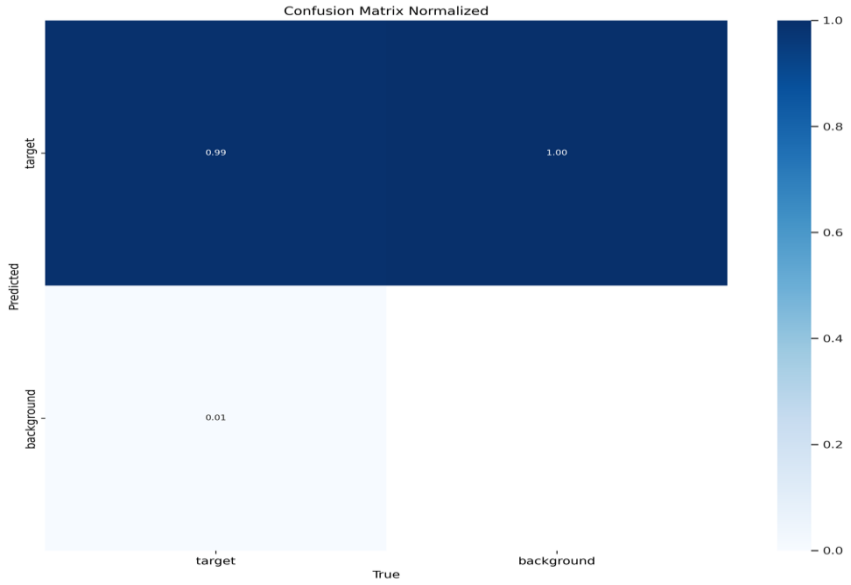

Figure 14

References

[1] X. Mao and W.-h. Diao, "Criterion to evaluate the quality of infrared small target images," *Journal of*

Infrared, Millimeter, and Terahertz Waves, vol. 30, no. 1, pp. 56--64, 2009.

[2]. He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in Proceeding of IEEE Conference on Computer Vision and Pattern Recognition (IEEE), 770–778.

[3]. Li, S., Xu, Y., Zhu, M., Ma, S., and Tang, H. (2019). Remote sensing airport detection based on end-to-end deep transferable convolutional neural networks. IEEE Geosci. Remote Sens. Lett. 16, 1640–1644. doi: 10.1109/LGRS.2019.29 04076

[4]. Hou, B., Ren, Z., Zhao, W., Wu, Q., and Jiao, L. (2020). Object detection in high- resolution panchromatic images using deep models and spatial template matching. IEEE Trans. Geosci. Remote Sens. 58, 956–970. doi: 10.1109/TGRS.2019.2942103

[5]. Zhong, Y., and Zheng, Z. Ma, A., Lu, X., and Zhang, L. (2020). Color: cycling, offline learning, and online representation framework for airport and airplane detection using gf-2 satellite images. IEEE Trans. Geosci. Remote Sens. 58, 8438–8449. doi: 10.1109/TGRS.2020.2987907

[6]. Fan, J., Lee, J. H., Jung, I. S., and Lee, Y. K. (2021). "Improvement of object detection based on Faster R-CNN and YOLO," in 2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (Jeju), 1–4.

[7]. Tu, J., Gao, F., Sun, J., Hussain, A., and Zhou, H. (2021). Airport detection in sar images via salient line segment detector and edge-oriented region growing. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 14, 314–326. doi: 10.1109/JSTARS.2020. 3036052

[8]. Dong, X., Tian, J., and Tian, Q. (2022). A feature fusion airport detection method based on the whole scene multispectral remote sensing images. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 15, 1174–1187. doi: 10.1109/JSTARS.2021.3139926

[9]. Mikriukov, G., Ravanbakhsh, M., and Demir, B. (2022). "Deep unsupervised contrastive hashing for large-scale cross-modal text-image retrieval in remote sensing," in 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (Singapore: IEEE), 4463–4467.

[10]. Lawal, M. O. (2021). Tomato detection based on modified YOLOv3 framework. Sci. Rep. 11, 1–11. doi: 10.1038/s41598-021-81216-5

[11]. Wu, W., Yin, Y., Wang, X., and Xu, D. (2018). Face detection with different scales based on faster R-CNN. IEEE T. Cybern. 49, 4017–4028. doi: 10.1109/TCYB.2018. 2859482.

[12]. Shakarami, A., Menhaj, M. B., Mahdavi-Hormat, A., and Tarrah, H. (2021). A fast and yet efficient YOLOv3 for blood cell detection. Biomed. Signal Process. Control 66, 102495. doi: 10.1016/j.bspc.2021.102495

[13]. Shi, P., Jiang, Q., Shi, C., Xi, J., Tao, G., Zhang, S., et al. (2021). Oil well detection via large-scale and high-resolution remote sensing images based on improved YOLO v4. Remote Sens. 13, 3243. doi: 10.3390/rs13163243

[14]. Lu, X., Ji, J., Xing, Z., and Miao, X. (2021). Attention and feature fusion SSD for remote sensing object detection. IEEE Trans. Instrum. Meas. 70, 1–9. doi: 10.1109/TIM.2021.3118092

[15]. Xu, D. and Wu, Y. (2020). Improved YOLO-V3 with DenseNet for multi-scale remote sensing target detection. Sensors 20, 4276. doi: 10.3390/s20154276

[16]. Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J.; Cho, K. Augmentation for small object detection. arXiv 2019, arXiv:1902.07296.

[17]. Chen, C.; Zhang, Y.; Lv, Q.; Wei, S.; Wang, X.; Sun, X.; Dong, J. Rrnet: A hybrid detector for object detection in drone-captured images. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4917–4926.

[18]. Yu, X.; Gong, Y.; Jiang, N.; Ye, Q.; Han, Z. Scale match for tiny person detection. In Proceedings of the IEEE/CVF Winter conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1257–1265.

[19]. Zhao L.; Liu, S.P. Small Target Detection Algorithm Based on Adaptive Fusion of Global and Local Image Features. 2022. Available online: https://xueshu.baidu.com/usercenter/paper/show?paperid=1d2w06s0an6r0rw01k660ex0kj632154&site=xueshu_se (accessed on 14 June 2023)

[20]. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; p. 30.

[21]. Cao, G.; Xie, X.; Yang, W.; Liao, Q.; Shi, G.; Wu, J. Feature-fused SSD: Fast detection for small objects. In Proceedings of the Ninth International Conference on Graphic and Image Processing (ICGIP 2017), Qingdao, China, 14–16 October 2017; Volume 10615, pp. 381–388.

[22]. Lim, J.S.; Astrid, M.; Yoon, H.J.; Lee, S.I. Small object detection using context and attention. In Proceedings of the 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Republic of Korea, 13–16 April 2021; pp. 181–186.

[23]. Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 7464–7475, 2022. 3, 6, 7, 9, 1

[24]. Xianzhe Xu, Yiqi Jiang, Weihua Chen, Yilun Huang, Yuan Zhang, and Xiuyu Sun. DAMO-YOLO: A report on real-time object detection design. arXiv preprint arXiv:2211.15444, 2022. 3, 7, 2, 4

[25]. Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. arXiv preprint arXiv:2402.13616, 2024. 2

[26]. Svanström F, Englund C and Alonso-Fernandez F. (2020). Real-Time Drone Detection and Tracking With Visible, Thermal and Acoustic Sensors

[27]. Aimi Salihai Abdul, Mohd Yusuff Masor and Zeehaida Mohamed, Colour Image Segmentation Approach for Detection of Malaria Parasiter using Various Colour Models and k-Means Clustering, In WSEAS Transaction on Biology and Biomedecine., vol. 10, January (2013).

Availability of data and materials

Demo videos for the proposed algorithm and YOLO can be found at https://www.youtube.com/playlist?list=PLvPINJRuTlZ9F-Pmst9B41tKxFpVgWYgn

The custom classifier and the detector code can be found on

https://github.com/saad4software/small_target_classifier

https://github.com/saad4software/small_target_detector

Competing interests

Funding

Authors' contributions

Saad Alkentar: 85%
Abdulkareem Assalem: 15%

Acknowledgements

Authors' information

Saad Alkentar: PhD student at Al Baath University, saad.zgm@gmail.com

Abdulkareem Assalem: Professor at Al Baath University, assalem1@gmail.com