Drone Surveillance using Detection, Tracking and Classification Techniques

Daitao Xing[1][0000−0001−6388−9300], Halil Utku Unlu[2][0000−0002−4368−5172], Nikolaos Evangeliou[3][0000−0003−0366−0226], and Anthony Tzes[3][0000−0003−3709−2810]

[1] Department of Computer Science and Engineering, New York University, 11201, New York, USA. daitao.xing@nyu.edu
Corresponding Author

[2] Department of Electrical and Computer Engineering, New York University, 11201, New York, USA. utku@nyu.edu

[3] Electrical Engineering, New York University Abu Dhabi and Center for Artificial Intelligence and Robotics, 129188, Abu Dhabi, UAE.
{nikolaos.evangeliou,anthony.tzes}@nyu.edu

Abstract. In this work, we explore the process of designing a long-term drone surveillance system by fusing object detection, tracking and classification methods. Given a video stream from an RGB-camera, a detection module based on YOLOV5 is trained for finding drones within its field of view. Although in drone detection, high accuracy and robustness is achieved with the underlying complex architecture, the detection speed is hindered on ultra HD-streams. To solve this problem, we integrate a high efficient object tracker to update target status while avoiding running the detection at each frame. Benefited from lightweight backbone networks with powerful Transformer design, the object tracker achieves real-time speed on standalone CPU devices. Moreover, a drone classification model is applied on the output of the detection and tracking mechanisms to further distinguish drones from other background distractors (birds, balloons). By leveraging inference optimization with TensorRT and ONNX, our system achieves extremely high inference speed on NVIDIA GPUs. A ROS package is designed to integrate the aforementioned components together and provide a flexible, end-to-end drone surveillance tool for real-time applications. Comprehensive experiments on both standard benchmarks and field tests demonstrate the effectiveness and stability of proposed system.

Keywords: Drone Detection and Classification · Object Tracking · Super Resolution

1 Introduction

The area of Unmanned aerial vehicles (UAVs) has drawn increasing attention in recent years due to its applications cross diverse fields such as aerial photography [10], mapping and surveying [28], search, rescue and emergency response [1].

The advent of low-cost small commercial drones led to their deployment into real world, while raising safety, privacy concerns and other types of challenges to the aviation industry [29], border security [8], and critical infrastructures [14]. Therefore, the demand of developing surveillance systems, especially for small drones has risen in the past few years to prevent intentionally or unintentionally misused of drones in urban environments, coastal border, airports and other safety-sensitive areas.

In recent years, there have been a lot of efforts in designing drone surveillance systems [6, 26] by adopting effective detection and countermeasure techniques including LiDAR, radio detectors, visual camera and passive acoustic sensors. Among those techniques, visual detection based on Deep Learning methods achieves remarkable progress on both effectiveness and accuracy. With the development of deep learning theory and optimization of hardware, modern object detectors obtain human-level compatible accuracy and operate in real-time speed even on mobile devices. However, drone detection is still a challenging problem due to its small size and fast maneuvers. Other factors caused by illumination change, heavy occlusion and target disappearance from the camera view further hinders drone detection.

To deal with the small object detection problem, recent works[19, 27, 5] utilize larger and deeper networks with more complex architecture to improve the model discriminative ability. However, constrained by the input size of neural networks, small drones only take less than 100 pixels in HD-frames, providing insufficient information for feature extraction and detection. On the other hand, blurred imaging of small objects from a long distance makes it harder to distinguish drones from other similar distractors like birds and airplanes. The only efficient solution is enlarging their input size to provide more useful information. However, this causes the exponential increment of computational complexity and will use most of the computational power, resulting in the processing delay and detection discontinuity in real applications.

In this work, we build a drone detection module based on YOLOV5. Due to the trade-off between complexity and precision, we choose YOLOV5-m as the base model and restrict the maximum input size to 1280 pixels. To avoid the computational overload caused by the drone detection, the used module only operates in a very low frequency(<1Hz). Considering the sparsity of drones occurrence in the field of view as well as the flying trajectory continuity, it is not necessary to run the detector on each frame and we use it as an indicator of the first appearance and disappearance of drones in camera view. Once a drone is captured by detection module, a more efficient object tracker running in real-time speed using low-resources will be initialized to update the drone status in the following frames.

Unlike object detection in which the model runs through the whole frame, object tracking ,instead, identifies the target object from a local patch, resulting in efficient and accurate schemes. Moreover, modern trackers are optimized for dealing with varying challenging scenarios like fast motion, low-resolution, frequent occlusion, etc. Recent years have witnessed many successful deep learning based object trackers , especially the family of Siamese Network based trackers[4, 2, 7], which play an important role in the visual tracking community. In this work, we employ a recent Transformer based object tracker, SiamTPN [32], which achieves a desired trade-off between tracking efficiency and accuracy for dealing with varying computational demand. Specifically, the SiamTPN obtains State-Of-The-Art performance while running at real-time speed on both CPU and GPU ends. The outputs from the drone detection modules initializes the tracker module. The trackers after initialization will track the detected objects as they move around frames. Once the target is lost or out of field view, the tracker will be removed from trackers' list. Thanks to the computational efficiency design, the trackers can be easily deployed to track multiple objects in parallel way on single GPU.

However, both detector and trackers may produce false negative predictions. The detector may takes airplanes or birds from a long distance as drones. Meanwhile, the object tracker fails when the object is out of view, occupied or distractors occur. In either case, it demands a robust classifier with strong discriminative ability to determine the the final classes for outputs from detector and trackers. Only the objects with a higher confidential score will be kept. To this end, we employ a pre-trained Resnet-50 [12] model and fine tune the final layers with custom classification datasets for drones. To deal with the tiny drones with very low resolution, we deploy a light-weight super resolution method, SR-GAN [18], to generate high-resolution patches before feeding them into classifier, which further improve the stability of classification model [24]. In practice, we only apply SRGAN [18] model on small patches with size less than 50×50 pixels.

By integrating the aforementioned components, we propose an efficient, end-to-end drone surveillance system, which can be easily deployed into embedding devices with low computational resources. We further boost the effectiveness by leveraging inference optimization techniques such as TensorRT and ONNX.

2 Related Work

2.1 Object Detection

The deep learning based object detection methods include two branches, like the two-stage methods,including Faster RCNN [25], and single-stage methods using SSD [21], YOLO [15, 16] and FCOS [27]. Two-stage methods divide the detection procedure into a coarse classification problem followed by a fine-tuning step, leading to a higher accuracy. Single stage methods, instead, aim to a desired trade-off between efficiency and precision, which is preferred in the systems with limited computational power. To balance the computing resource allocation between object detector and trackers, we employ the single stage YOLOV5 as our detector.

2.2 Object Tracking

The tracking methods can be divided into: a) Discriminative Correlation Filter (DCF) based trackers and b) deep learning-based trackers. DCF based trackers [13, 23, 3] could run with real-time speed on CPU, but their performance is constrained by the feature representation ability of handcrafted features. In contrast, deep learning based trackers, like the Siamese-based trackers [4, 2, 7, 32] achieve remarkable enhancements in both accuracy and speed by utilizing a high-end GPU device.

2.3 Classification and Super Resolution

Early classification methods like AlexNet [17], and Resnet [12] get higher accuracy in using deeper and wider networks. Among those classifiers, the Resnet family is the most popular framework and is adopted in many computer vision tasks as backbone network. In this work, we use Resnet-50 as our classifier, due to its balance between efficiency and accuracy. We further boost the performance by applying a lightweight super-resolution model, SRGAN, to deal with the small drones with low resolution.

2.4 Inference Optimization

TensorRT is a C++ library that facilities high-performance inference on NVIDIA graphics processing units (GPUs). TensorRT applies graph optimizations, layer fusion, among other optimizations, while also finding the fastest implementation of that model leveraging a diverse collection of highly optimized kernels.

3 Drone Surveillance System

This section presents the drone surveillance system design and the implementation details of each component.

3.1 System Overview

Given the frame f at time t, we first resize the image without crop and maintain the aspect ratio before feeding them into detector $\mathcal{D}$. For images of size 1920×1080 or less, we resize images so that the longer edge equals to 1280 pixels. The object detector returns a new set of recognized drones as $\mathbf{d} = \{\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_n\}$, where $\mathbf{d}_i$ is represented as the concatenation of the bounding box coordinates $\{x, y, w, h\}$ and confidential score $s_{\mathcal{D}}$. For images with higher resolution, we follow the image tilling strategy [30] in which the image is divided into multiple tiles of a fixed size. The tiled images are processed with same detector in a batch manner. The final prediction is the aggregation of outputs from each tile. The detector is set to run at low frequency (<1Hz) for inference efficiency.

Each tracker $\mathcal{T}$ is responsible for a specific object and returns the updated status as $\mathbf{t}_i = \{x, y, w, h, s_{\mathcal{T}}\}$ where $s_{\mathcal{T}}$ is its confidential score. Together, we have a set of drone candidates $\{\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_n\, \mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_m\}$ from the detector and trackers outputs. We crop patches according to those candidates and feed

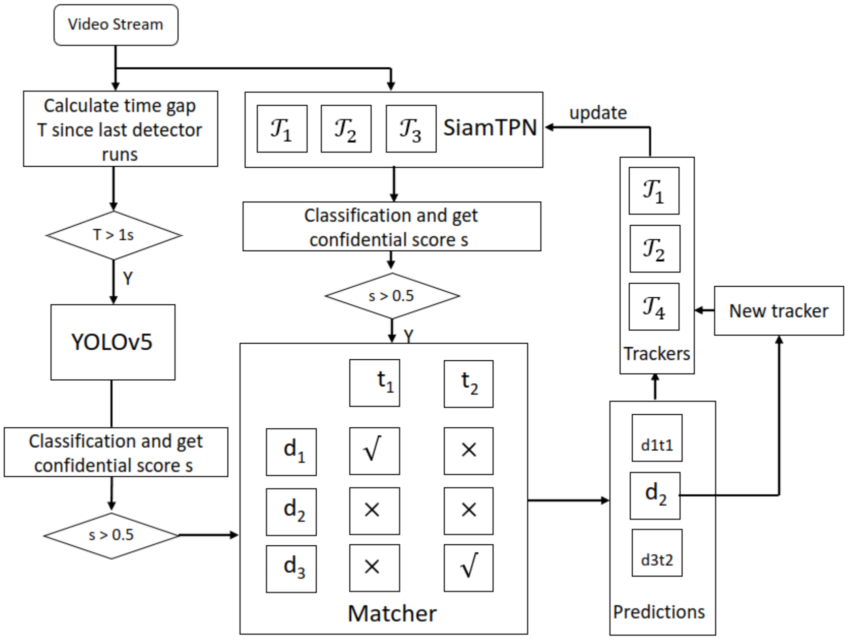

Fig. 1. Drone surveillance system overview

them into the drone classifier $\mathcal{C}$ for further discrimination. A confidential score $s_{\mathcal{C}}$ is provided for each candidate. Overall, the final confidential score is calculate as $s = s_{\mathcal{C}} \times s_{\mathcal{D},\mathcal{T}}$; only candidates with this score s higher than a threshold will be kept for further processing, while for those candidates whose confidential score falls below the threshold, the corresponding tracker will be removed from tracker list.

A matcher, based on the criterion of maximum Intersection Over Union (IOU), is designed to match the candidates from $\mathbf{d}$ to $\mathbf{t}$. As shown in Figure 1, the process is similar to non maximum supppression (NMS). Specifically, if a candidate $\mathbf{d}_i$ is matched to $\mathbf{t}_j$, or vice versa, the two instances will be merged and use the one with higher confidential score as final output. If no matches found for instances from $\mathbf{d}$, a new tracker $\mathcal{T}$ will be initialized and added into the tracker list. We should mention that the matcher only works when both detector and trackers are active.

3.2 Detection Module

We select the single-stage object detector YOLOv5 [16] for its efficiency and speed on object detection tasks. Specifically, the COCO [20] pre-trained YOLOv5-m model with input size of 1280 is adopted. In all experiments, the networks were trained using 40 epochs on 4-GPUs with 16 images per batch. We use the ADAM [22] optimizer with the initial learning rate of 10^{-4}. For the training dataset, we consider the image sequences from Drone vs. Bird Competition [26] and USC drone detection and tracking dataset [31]. We uniformly sampled frames with a fixed rate (5 fps) from each sequence and extracted 32067 images in total for training. We select 4 videos from Drone vs. Bird training dataset for detection and tracking validation purpose.

3.3 Tracking Module

For object tracking, we employ a real-time Siamese Network based deep learning tracker, SiamTPN [32], for its robust performance and real-time speed. As shown in Figure 2, the SiamTPN utilizes a lightweight backbone and optimized transformer based pyramid network to learn discriminative features from both template and search images. The final prediction is returned after the cross correlation layer. The template image is cropped from image when the detector recognizes a new drone which is not tracked yet. The search image is cropped from the following frames and resized into 256 × 256. Benefited from the small input size and optimized architecture, the tracker runs at 50 FpS on CPU and over 100 FpS on GPU, where more details can be found in [32]. We compare the performance between the SiamTPN with default trackers provided by OpenCV in Section 4. For inference, we directly use the pre-trained model from SiamTPN without further finetuning since the tracker is designed to track any generic objects specified by the template.

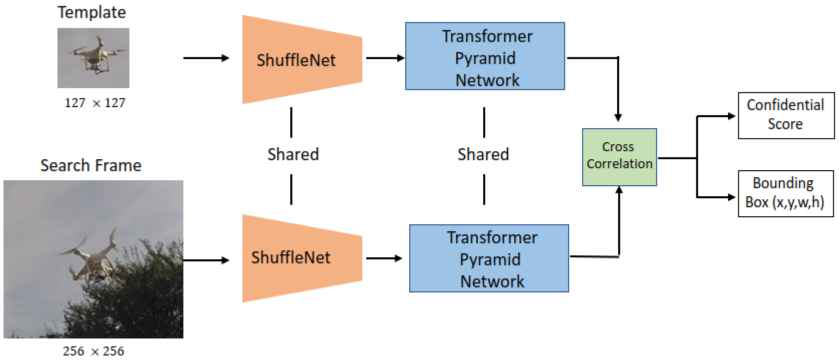

Fig. 2. SiamTPN: Architecture Overview

3.4 Classification Module

The drone classification module is fine-tuned on a pre-trained Resnet-50 [12] model. We cropped drones patches and resize them into 224 × 224, yielding

33733 positive samples. For negative images, we randomly select images from the ImageNet [9] dataset. We found that the negative images are easy to be classified due to the low similarity to drones. Therefore, we run the drone detector on the training dataset from the object detection part and select the false positive predictions as false samples for classification, resulting in a robust performance. During training, we freeze the whole network except the final fully-connected layers for fine-tuning. The network is trained for 50 epochs with 128 images per batch.

4 Experimental Studies

This section first presents the effects of proposed components in the aspects of accuracy and speed. We further apply the system in the field test videos to validate its performance in real-world applications.

4.1 Overall Performance

In order to compare the effect of each proposed component regarding accuracy and speed, we choose 4 videos from the Drone vs. Bird traning dataset as validation dataset. We adopt the Average Precision (AP) metric which is extensively used in an object detection task. The prediction outputs are counted as correct when its IOU score with a ground truth bounding box is higher than a threshold. In this study, we test the AP score under varies criterion, including AP under different threshold value (AP, AP_{50}, AP_{75}) and AP for drones with different sizes (AP_S, AP_M, AP_L). All AP scores are calculated with COCO API. Table 1 shows the overall performance on validate set when different components are active. YOLOv5 detector alone shows relative poor performance on video detection, having an AP of 43.4. We notice that the YOLOv5 detector is sensitive to the complex scenarios like object deformation, illumination and object occupation. Since object detector treats videos as independent frames, the predictions shows inconsistency even in adjacent frames. The tracker boosts the performance by 50% by guaranteeing the prediction continuity between frames. The AP_{50} achieves 89 on the validated dataset. The classification module brings relative small performance changes, but it provides an additional check which is useful when the trackers lost the target and return false positive outputs.

	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
YOLOv5	43.4	62.3	52.4	30.5	48.1	64.7
YOLOv5 + SiamTPN	63.1	89.0	77.8	45.9	69.3	91.2
YOLOv5 + SiamTPN + Classification	63.5	89.9	77.9	47.2	69.1	91.5

Table 1. Overall performance on Validation dataset. Average Precision (AP) scores are calculated with the COCO API. AP_{50}, AP_{75} represent AP with IOU above 0.5, 0.75 respectively. AP_S, AP_M, AP_L represent AP for small, medium, large objects respectively.

To further investigate the tracker's performance, we compare the SiamTPN with 3 default trackers from OpenCV, which are: a) CSRT tracker [23], b) KCF tracker [13], and c) MIL tracker [11]. We perform an One-Pass Evaluation (OPE) and measure the precision and success of different tracking algorithms on validating videos. Different from Average Precision, in OPE, the precision is computed by comparing the distance between tracking result and ground truth bounding box in pixels. The success is computed as the the IOU scores between tracking result and ground truth bounding box at different threshold levels. Finally, we rank the tracking methods using the Area Under the Curve (AUC). As shown in Figure 3, the KCF performs poorly on both precision and accuracy. CSRT provides compatible results on precision scores but still have a large gap on success rate compared with SiamTPN. Table 2 shows the speed comparison between those trackers, in which, MIL, KCF and CRST only support CPU while SiamTPN support both CPU and GPU. Overall, SiamTPN achieves best performance in the aspects of speed, accuracy and robustness.

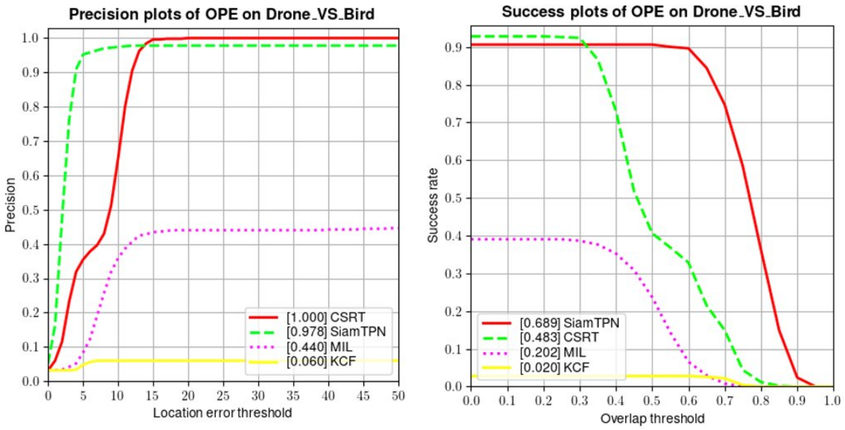

Fig. 3. Tracking performance comparison with OpenCV trackers

	MIL	KCF	CSRT	SiamTPN	SiamTPN(GPU)
FpS	6	240	45	52	102

Table 2. Speed comparison between trackers

4.2 Speed Analysis

In Table 3, we compare the inference speed of each modules and their combination performance. Due to the large input size (1280), YOLOv5m only operates at around 12 FpS on GPU after inference optimization, which is not suitable for applications with real-time requirements. Instead, the optimized SiamTPN achieves 100+ FpS on GPU. Benefiting from small network size and smaller input size, the classification module and super resolution model, SRGAN, require much less computation resources compared with other detection and tracking models. By combining the YOLOv5 and SiamTPN and constraining the detector operation frequency, the inference achieves a real-time speed of 37 FpS. The speed decline comes from the heavy detection module and multiple trackers running in parallel. Nevertheless, the combination of detector and trackers obtains a desired trade-off between accuracy and speed. The classification module and SRGAN introduces a slight computational burden to the system.

#	YOLOv5	SiamTPN	Classification	SRGAN	FpS
1	✓				12
2		✓			102
3			✓		178
4				✓	205
5	✓	✓			37
6	✓	✓	✓		32
7	✓	✓	✓	✓	29

Table 3. Speed comparison between individual modules and difference configurations. All models are accelerated with GPU and TensorRT.

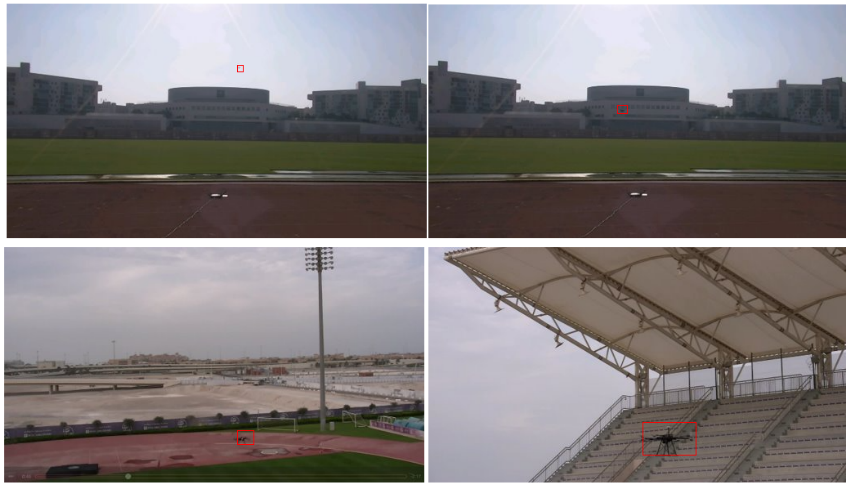

Fig. 4. Visualization of drone surveillance system in field test. The drones are captured by a still camera on the ground (first row) or a camera mounted on a flying drone (second row)

4.3 Field Test Analysis

To validate the reliability of the proposed drone surveillance system in real-world scenarios, we set up several field tests with challenging factors including

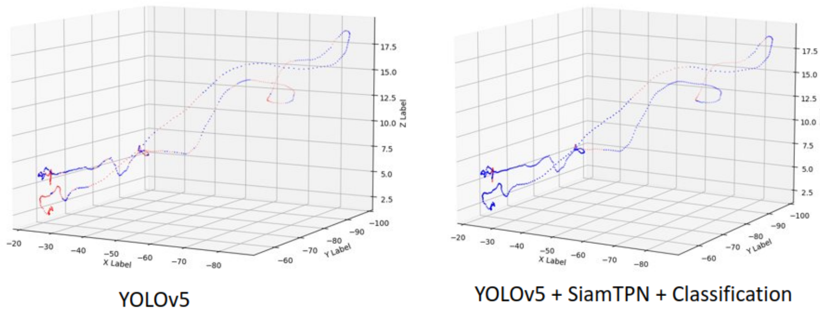

Fig. 5. Drone surveillance performance in real world test.

scale variance, out-of-view, object deformation and partial occlusion. Figure 4 shows the flight status and detection results in difference scenarios. To verify the advantage of proposed system over single detection modules, we compare the drone trajectories coverage percentages based on their prediction results. As shown in Figure 5. For better visualization, we recorded GPS data and plot the trajectory in 3D as red dots. The dots are labeled as blue only if the drone in this position is correctly recognized. The configuration with YOLOv5, SiamTPN and Classification obtains more consistent predictions than detector alone.

5 Conclusions

In this work, we propose a long-term drone surveillance system which consists of a YOLOv5 based drone detector, real-time object tracker, drone classifier and other auxiliary modules. Those modules are integrated in an efficient way and are optimized with inference acceleration techniques (TensorRT and ONNX) to achieve best performance. Our method ranked second in the 2022 Drone vs. Bird detection challenge. We have also verified our system in real-world test with the preliminary results from both field tests and competition demonstrating the effectiveness of the proposed system.

References

1. Ajith, V., Jolly, K.: Unmanned aerial systems in search and rescue applications with their path planning: a review. In: Journal of Physics: Conference Series. vol. 2115, p. 012020. IOP Publishing (2021)
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional Siamese networks for object tracking. In: European Conference on Computer Vision. pp. 850–865. Springer (2016)

3. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2544–2550. IEEE (2010)
4. Cao, Z., Fu, C., Ye, J., Li, B., Li, Y.: SiamAPN++: Siamese Attentional Aggregation Network for Real-Time UAV Tracking. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1–7 (2021)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020)
6. Coluccia, A., Fascista, A., Schumann, A., Sommer, L., Dimou, A., Zarpalas, D., Akyon, F.C., Eryuksel, O., Ozfuttu, K.A., Altinuc, S.O., et al.: Drone-vs-Bird Detection Challenge at IEEE AVSS2021. In: 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–8. IEEE (2021)
7. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Atom: Accurate tracking by overlap maximization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4660–4669 (2019)
8. De Cubber, G., Shalom, R., Coluccia, A., Borcan, O., Chamrád, R., Radulescu, T., Izquierdo, E., Gagov, Z.: The safeshore system for the detection of threat agents in a maritime border environment. In: IARP Workshop on Risky Interventions and Environmental Surveillance (2017)
9. Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009)
10. Gotovac, D., Gotovac, S., Papić, V.: Mapping aerial images from uav. In: 2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech). pp. 1–6 (2016)
11. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: Bmvc. vol. 1, p. 6. Citeseer (2006)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **37**(3), 583–596 (2014)
14. de la Iglesia, D., Mendez, M., Dosil, R., Gonzalez, I.: Drone detection cnn for close-and long-range surveillance in mobile applications. Proceedings of the AVSS, Taipei, Taiwan pp. 18–21 (2019)
15. Jiang, Z., Zhao, L., Li, S., Jia, Y.: Real-time object detection method based on improved yolov4-tiny. arXiv preprint arXiv:2011.04244 (2020)
16. Jocher, G.: ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements (Oct 2020)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)
18. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017)
19. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)

20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision. pp. 740–755. Springer (2014)
21. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
22. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
23. Lukezic, A., Vojir, T., ˇCehovin Zajc, L., Matas, J., Kristan, M.: Discriminative correlation filter with channel and spatial reliability. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6309–6318 (2017)
24. Magoulianitis, V., Ataloglou, D., Dimou, A., Zarpalas, D., Daras, P.: Does deep super-resolution enhance uav detection? In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). pp. 1–6. IEEE (2019)
25. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
26. Svanström, F., Englund, C., Alonso-Fernandez, F.: Real-time drone detection and tracking with visible, thermal and acoustic sensors. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 7265–7272. IEEE (2021)
27. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (2019)
28. Tokekar, P., Vander Hook, J., Mulla, D., Isler, V.: Sensor planning for a symbiotic uav and ugv system for precision agriculture. IEEE Transactions on Robotics **32**(6), 1498–1511 (2016)
29. Tsoukalas, A., Xing, D., Evangeliou, N., Giakoumidis, N., Tzes, A.: Deep learning assisted visual tracking of evader-UAV. In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS). pp. 252–257. IEEE (2021)
30. Unel, F., Ozkalayci, B., Cigla, C.: The power of tiling for small object detection. 2019 ieee. In: CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA. https://ieeexplore. ieee. org/document/9025422 (2019)
31. Wang, Y., Chen, Y., Choi, J., Kuo, C.C.J.: Towards visible and thermal drone monitoring with convolutional neural networks. APSIPA Transactions on Signal and Information Processing **8** (2019)
32. Xing, D., Evangeliou, N., Tsoukalas, A., Tzes, A.: Siamese transformer pyramid networks for real-time uav tracking. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2139–2148 (2022)