2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

AirTrack: Onboard Deep Learning Framework for Long-Range Aircraft Detection and Tracking

Sourish Ghosh[1], Jay Patrikar[1], Brady Moon[1], Milad Moghassem Hamidi[1], and Sebastian Scherer[1]

***Abstract*— Detect-and-Avoid (DAA) capabilities are critical for safe operations of unmanned aircraft systems (UAS). This paper introduces, AirTrack, a real-time vision-only detect and tracking framework that respects the size, weight, and power (SWaP) constraints of sUAS systems. Given the low Signal-to-Noise ratios (SNR) of far away aircraft, we propose using full resolution images in a deep learning framework that aligns successive images to remove ego-motion. The aligned images are then used downstream in cascaded primary and secondary classifiers to improve detection and tracking performance on multiple metrics. We show that AirTrack outperforms state-of-the art baselines on the Amazon Airborne Object Tracking (AOT) Dataset. Multiple real world flight tests with a Cessna 182 interacting with general aviation traffic and additional near-collision flight tests with a Bell helicopter flying towards a UAS in a controlled setting showcase that the proposed approach satisfies the newly introduced ASTM F3442/F3442M standard for DAA. Empirical evaluations show that our system has a probability of track of more than 95% up to a range of 700m. [Video][1]**

I. Introduction

Mid-air collision (MAC) and near mid-air collision (NMAC) risk are concerns for both manned and unmanned aircraft operations, especially in low-altitude airspace. Detect-and-Avoid (DAA), also commonly referred to as *sense* and *avoid*, is defined as "the capability of an aircraft to remain *well clear* from and avoid collisions with other airborne traffic." [1] The well clear boundary as defined by NASA [2] mathematically characterizes a volume, referred to as the Well Clear Violation, such that if aircraft pairs jointly occupy this volume, they are considered to be in a Well Clear Violation. In visual flight rules conditions, NMAC/MAC threat mitigation is carried out by a pilot visually *detecting* and *avoiding* other aircraft to remain *well clear* [3] of them. Typically for medium to large airborne systems, an active onboard collision avoidance system such as the Traffic Alert and Collision Avoidance System or the Airborne Collision Avoidance System is used and relies on transponders installed in cooperative aircraft. However, not all airborne threats can be tracked using transponders. Rogue drones, balloons, light aircraft, and inoperative transponders present a threat to reliable operations. This makes DAA an essential requirement especially for beyond visual line of sight operations in the National Airspace System.

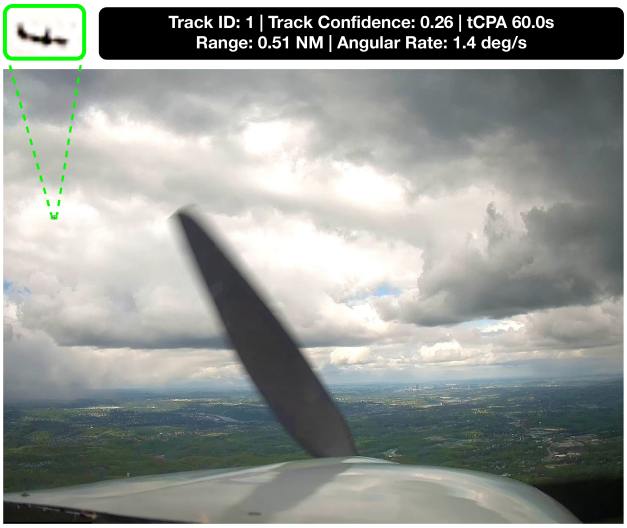

Fig. 1: Snapshot showing visual detection and tracking of an intruder in the real world tests with a Cessna 182 aircraft using our system. The bounding box width is less than 15px in an image of resolution 2448×2048 highlights the challenges of visual DAA. Top left corner shows the zoomed in crop around the detected object. The information box shows the relevant DAA parameters computed by the AirTrack framework.

Human vision is the last line of defence against a mid-air collision and is thus critical for aviation safety. Therefore, in order to assist pilots in mitigating mid-air collision threats, machine vision can be used to alert pilots of potential aircraft and objects in the sky. Due to size, weight, and power (SWaP) constraints of UASs, radar is often not a feasible solution. Machine vision has been a promising direction for research in this domain [4] based on success of CNN-based networks.

In order to standardize DAA capabilities, performance requirements F3442/F3442M - 20 have been published by ASTM [5] which define safe DAA operations of UAS as having a maximum dimension of less than or equal to 25ft, operating at airspeeds below 100kts, and of any configuration or category. This standard does not define a specific DAA architecture. According to this standard, for safe DAA operations at a particular intruder range, the probability of track must be greater than 95% and the range estimation error of the intruder must be within 15%, along with a maximum cap on the angular rate error based on ownship specifications.

In this work, we present AirTrack, a machine vision-based

*This work was supported by the Army Research Laboratory (ARL) grant: **ARL IDIQ-HA W911QX-20-F0106**. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE1745016.

[1]The authors are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA 15213. {sourishg, jaypat, bradym, mmoghass, basti}@andrew.cmu.edu

[1]Video: https://youtu.be/bMw5nUGL5GQ

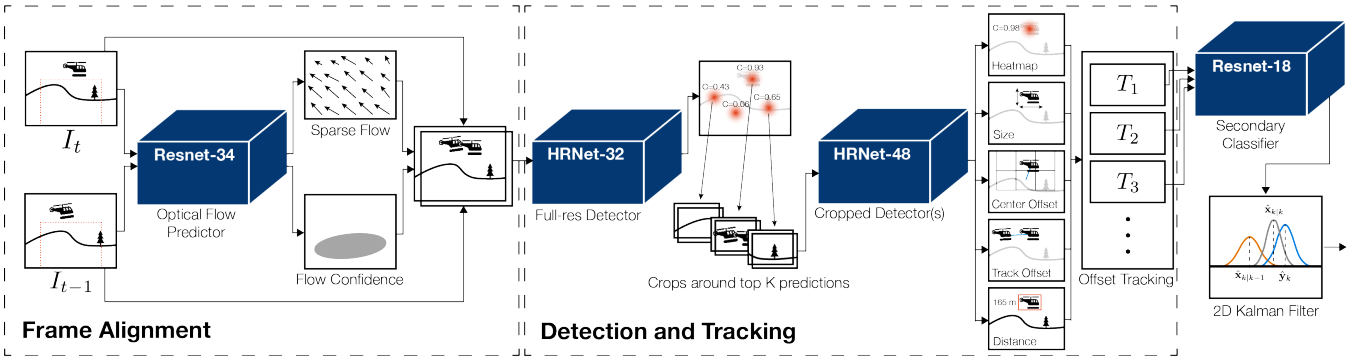

Fig. 2: AirTrack architecture showing the various internal components. The architecture uses frame alignments, cascaded detection modules and a secondary classifier. Finally a 2D Kalman filter is used to get the relevant intruder DAA parameters. Shaded gray boxes are the modules are the white boxes are the input/output variables.

solution for detecting and tracking aircraft that aims to satisfy the ASTM performance standards. We propose an end-to-end deep learning solution that leverages recent advances in machine vision to enable a real-time SWaP-C solution for DAA. Given the low Signal-to-Noise ratios (SNR) of the detection, we propose using full resolution images in a deep learning framework that aligns successive images to remove ego-motion. The aligned images are then used downstream in cascaded primary and secondary classifiers to improve detection performance on multiple metrics. The key contribution of this work is three-fold: (1) an end-to-end vision-based aircraft detection and tracking system that uses cascaded detection modules, (2) a system of primary and secondary classification modules that maintain high precision in novel scenarios via false-positive mining and retraining, and (3) real-world flight tests with onboard SWaP hardware on both a general aviation Cessna 182 performing standard general aviation flight behaviors as well as on a UAS performing dedicated near-collision experiments with a Bell helicopter intruder flying towards the UAS in controlled settings, with both showcasing satisfaction of ASTM F3442/F3442M standards for DAA.

II. Related Work

Vision-based DAA has a rich literature background spanning classical pipe-line based techniques to modern era end-to-end deep learning techniques.

A. Classical Approaches

Traditionally, the main solutions for visual DAA included a modular design with various well-known techniques commonly used in classical computer vision [6], [7], [8], [9], [10], [11]. The primary modules are the following in sequential order: frame stabilization, background-foreground estimation using morphological operations, temporal filtering, detection, and tracking. The frame stabilization step is typically handled using either optical flow-based approaches [12] or image registration using feature matching [13], [14]. [15] used regression-based motion compensation both in the horizontal and vertical directions, and the morphological operations are used to enhance the signal of the intruder aircraft. [6], [7], [9] used background subtraction. Machine learning based approaches have been also used to learn descriptors of the aircraft [15], [7], [16] using methods such as SVMs. Naively using morphological operations would result in false positives, hence a temporal filtering stage is used. Track-before-detect is a commonly used technique for detecting low-SNR targets in infrared imagery [17] and several have tried using it in vision-based DAA. Approaches for tracking include using Kalman filter banks [18], Hidden Markov Models [6], [19], [20], and Viterbi-based filtering [9], [6].

B. Deep Learning Approaches

1) Detection: Modern visual DAA approaches rely on deep neural networks based object detection and tracking methods. Detection is typically accomplished using convolution neural networks (CNN)s [21], [22], [23], [24]. The key challenge is visually detecting small objects in large resolution images. Due to the very low SNR, standard anchor-based methods such as YOLO and R-CNN are not ideal for small object detection. Keypoint-based architectures [25] are much more suitable for small objects. Related work can also be found in the domain of face detection [26], aerial imagery [27], and pedestrian tracking [28]. Since computational efficiency is a key requirement for DAA systems, state of the art approaches include fully convolution networks with heatmap prediction [24], [29].

2) Tracking: Tracking-by-detection is the commonly used paradigm for aircraft tracking [30]. A tracking management system is typically maintained to account for birth and death of new tracks, and associate new detections to existing tracks using the Hungarian algorithm [6], [7]. In multi-object tracking, typically a metric like Intersection-over-Union (IoU) for bounding boxes is used to determine the assignment, but for small objects the IoU metric becomes too sensitive to slight deviations in bounding box positions. In that case, an integrated detection and tracking approach such as [31] is a better choice, which our approach is based on.

This manuscript is organised as follows: Section II details prior approaches in visual DAA including classical approaches along with modern state-of-the-art deep learning-

based approaches. Section III explains the proposed approach in detail, and Section IV includes the specifics to our implementation. Section V contains an evaluation of the proposed method based on relevant metrics and real-world testing. Finally, section VI presents concluding remarks and future work.

III. Methodology

The overall system design of AirTrack is shown in Fig. 2 and consists of the following four sequential modules: (1) Frame Alignment, (2) Detection and Tracking, (3) Secondary Classification, and (4) Intruder State Update. Let the inputs be two successive grayscale image frames $I_t, I_{t-1} \in \mathbb{R}^{H \times W}$ where $H \times W$ are the dimensions of the input frames. We utilize the full image resolution during inference to maximize the chances of detection at long ranges ($\geq$ 1km). The final outputs of the system are a list of tracked objects with bounding box coordinates, track ID, 2D Kalman Filter state (containing pixel-level velocity and acceleration), estimated range, and time to closest point of approach (tCPA). The following subsections describe each of the modules in detail.

A. Frame Alignment

In order to help distiquish the foreground objects from the background, one must align successive frames in a video so that the ego-motion of the camera can be discarded. This is done with the help of a frame alignment module that predicts the optical flow between two successive image frames and the confidence of the predicted flow.

This module takes as input the current and previous input frames $I_t, I_{t-1} \in \mathbb{R}^{H \times W}$, where H and W are the input image height and width respectively. Since the sky is mostly textureless, it does not provide much addition information for computing the background flow. The input images are thus cropped from the center-bottom covering most of the high-texture details present below the horizon using a fixed-size crop of size 2048 × 1280. The backbone architecture for alignment is a ResNet-34 with two prediction heads: (1) optical flow offsets $\mathbf{F}_t \in \mathbb{R}^{2 \times H \times W}$, (2) confidence heatmap of offsets $\mathbf{C}_t \in \mathbb{R}^{H \times W}$. The prediction is made at 1/32 scale of the input, and low confidence offset predictions are rejected.

For training, 75% of the input image tuples are created by data-augmentation. The data-augmentation involves generating a random affine homography by sampling the parameters from a normal distribution. This random homography is used to warp an image frame and thus create a training sample. The remaining 25% of the input image tuples are picked as successive frames from the dataset. The target homography for the neural network is generated by first computing the Lucas-Kanade optical flow (OpenCV) and then finding the affine transform. The training objective minimized is the following:

$$\frac{1}{N} \sum \left(\mathbf{C}_t \circ \text{MSE} \left(\mathbf{F}_t, \hat{\mathbf{F}}_t \right) \right)$$

where $\hat{\mathbf{F}}_t$ and $\mathbf{F}_t$ are the predicted and ground truth optical flow respectively, and MSE denotes the mean squared error.

B. Detection

The detection architecture is made up of two cascaded detection modules. The primary detection module takes in two full resolution aligned image frames $H \times W$, and the secondary detection module takes as input a smaller crop of $h \times w$, $h, w \approx H/5, W/5$ around the top k detector outputs (based on confidence) of the primary module. The cascaded network is fully-convolutional, and the output scale is 1/8 of the input resolution. It outputs five maps: (1) center heatmap encoding the center of the object, (2) bounding box size, (3) center offset from grid to center of object, (4) track offset of object center from the previous frame, (5) distance of the object in log scale. Each of these five heads is trained with a separate loss function.

The target center heatmaps during training are rendered using a Gaussian kernel. For distant objects, this results in a single-pixel render which is not helpful for training. Therefore a minimum box size of 3×3 is used for center rendering. The center heatmap is trained using the focal loss [32] for handling large class imbalances:

$$L_h = \frac{1}{N} \sum_{xy} \left\{ \begin{array}{l} (1 - \hat{H}_{xy})^{\alpha} \log \hat{H}_{xy}, \text{if } H_{xy} = 1 \\ (1 - H_{xy})^{\beta} \hat{H}_{xy}^{\alpha} \log(1 - \hat{H}_{xy}), \text{otherwise} \end{array} \right\}$$

where x, y are the pixel locations in ground-truth and predicted heatmaps $H, \hat{H}$, and α, β are the focal loss parameters. The peaks of this predicted heatmap are the object centers, and we choose the corresponding values from the other predicted heads based on the pixel locations of the peaks.

The bounding box size prediction is regressed by minimizing the following objective:

$$L_{\text{size}} = \frac{1}{N} \sum_{i=1}^{N} |\hat{s}_{\mathbf{p}_i} - \mathbf{s}_i|$$

where $\hat{s}_{\mathbf{p}_i}$ is the predicted box size at pixel location $\mathbf{p}_i$ and $\mathbf{s}_i$ is the ground truth box size. L1 error is also minimized for the center offset prediction:

$$L_{\text{off}} = \frac{1}{N} \sum_{i=1}^{N} |\hat{o}_{\mathbf{p}_i} - o_{\mathbf{p}_i}|$$

where $\hat{o}_{\mathbf{p}_i}$ is the predicted offset and $o_{\mathbf{p}_i}$ is the ground truth offset. The track offset loss is again an L1 loss:

$$L_{\text{track}} = \frac{1}{N} \sum_{i=1}^{N} \left| \hat{T}_{\mathbf{p}_i^{(t)}} - \left(\mathbf{p}_i^{(t-1)} - \mathbf{p}_i^{(t)} \right) \right|$$

where $\hat{T}_{\mathbf{p}_i^{(t)}}$ is the predicted track offset and $\mathbf{p}_i^{(t-1)} - \mathbf{p}_i^{(t)}$ denotes the change in the location of the center pixel from the previous frame. Finally, the distance prediction is trained by optimizing the following L2 loss:

$$L_{\text{dist}} = \frac{1}{N} \sum_{i=1}^{N} \left| \log \hat{D}_{\mathbf{p}_i} - \log D_{\mathbf{p}_i} \right|$$

where $\hat{D}_{\mathbf{p}_i}$ and $D_{\mathbf{p}_i}$ are the predicted and ground truth distance. To help with training stability, we predict log

distance to scale the large distance values. The overall training objective that is minimized is thus:

$$L_{\text{total}} = w_1 L_h + w_2 L_{\text{size}} + w_3 L_{\text{off}} + w_4 L_{\text{track}} + w_5 L_{\text{dist}}$$

where $w_i, i = \{1, \ldots, 5\}$ are hyper-parameters.

C. Tracking

The tracking approach builds on top of the offset tracking vector approaches [31]. The key idea in this algorithm is to use the track offset vector to associate current frame detection to a list of existing tracks. The power of this algorithm lies in its simplicity. Using the predicted track offset vector, we subtract the offset from the current object center to recover the center location in the previous frame. Then we compare the previous frame center with the offset-adjusted center and check if the distance between them falls below a certain threshold κ. If it does, we propagate the existing track ID to the current detection and mark it as matched. Otherwise, we spawn a new track ID with the current detection and proceed.

D. Secondary Classifier

A ResNet-18 module is used as a secondary classifier for false-positive rejection. The input to the network is a fixed-size crop (padded and resized) around the bounding boxes detected by the object detectors. It is a binary classification network that predicts whether the input crop is an aircraft or not. The training data for this network is collected from the results of the detectors. We mine false-positive samples to train the classifier for better false-positive rejection. We use focal loss [32] to train the model since a training batch contains more false-positive samples than true positives. The secondary classifier aims to improve the overall precision of the system and to allow quick retraining using novel data rather than training the detector, which would take some time. This enables one to run the object detectors at a low confidence threshold to improve recall.

E. Intruder State Estimation

The final stage of our DAA module is intruder state estimation. This module maintains an internal state of each intruder detected using a 2D Kalman Filter with a constant acceleration motion model. For offset tracking, this filter is necessary to compute pixel-level velocity and acceleration. The angular-rate is computed as:

$$r = \theta \sqrt{\dot{x}^2 + \dot{y}^2}$$

where θ is the degree/pixels ratio of the camera, and $\dot{x}, \dot{y}$ denotes the pixel velocity of the object center. This module also computes the time to closest point of approach:

$$\text{tCPA}(i, j) = \frac{t_j - t_i}{-1 + \sqrt{\frac{a_j}{a_i}}}$$

where a_k, t_k denotes the area of the bounding box and time at frame k respectively. This tCPA formulation is based on [33] and the key idea is that as the bounding box area of the intruder increases in size (meaning that the object is getting closer), the tCPA value decreases. tCPA is inversely proportional to the rate of change of the square root of the bounding box area.

IV. Implementation

Since we are dealing with small objects, the choice of the backbone architectures is very crucial for good performance. The key requirements are learning high-resolution image features that can be used to identify tiny objects against the sky and ground clutter while also learning enough context to ignore false positives. This section provides the necessary implementation details.

A. Dataset

For training, we use the Airborne Object Tracking (AOT) Dataset. This dataset was released in 2021 as part of the Airborne Object Tracking Challenge [34] organized by AIcrowd in partnership with Amazon Prime Air. It is a collection of around 5000 flight sequences of 120 seconds each at 10Hz resulting in 164 hours of total flight data. There are a total of 3.3M+ labeled image frames containing airborne objects. The image resolution is 2448x2048, and the images are 8-bit grayscale. Along with bounding box and class labels, the annotations also include range information of the aircraft for a part of the dataset. The range mainly varies from 600 to 2000 meters (25-75 percentiles). The area of the objects labeled vary from 4 to 1000 sq pixels. Among the planned airborne encounters, among 55% of them would qualify as potential collision trajectories. 80% of the targets are above the horizon, 1% on the horizon, and 19% below the horizon. The dataset also captures different sky and visibility conditions; 69% of the sequences have good visibility, 26% have medium visibility, and 5% exhibit poor visibility conditions. We use a train/val split of 90/10 to train and evaluate models on this dataset.

B. Training Details

The backbone architectures used for the primary and secondary detectors are HRNet [35], and it is an ideal choice for this problem because it fuses high-level and low-level parallel convolutional feature maps using a unique fusion operation that preserves high-resolution features with enough low-level context. The full-resolution detector predicts the initial heatmap. Then 512×512 crops are taken around the top K ($K = 4$) heatmap peaks and formed into an input batch for the cropped detector. The cropped detector is typically chosen to be a heavier neural network with more parameters since it is operating on a lower-resolution image.

The frame alignment and detection neural networks are trained using the SGD optimzer [36] with cosine annealing warm restarts [37] as the learning rate scheduling strategy. 512×512 image crops are used to train all the detection networks. Since they are fully-convolutional, smaller random image-crop pairs are sufficient for training. The sampling strategy of the image batches is important for a good overall performance of the detector. For training batch samples, 50% are chosen with random crops, 25% are chosen around

Metric	Description	Domain
P ↑	Precision	$[0, 1]$
R ↑	Recall	$[0, 1]$
EDR ↑	Encounter Detection Rate	$[0, 1]$
FPPI ↓	False positives per image	≥ 0
IDSPI ↓	Track ID switches per image	≥ 0
ARE ↓	Angular rate error	≥ 0 (deg/s)

TABLE I: Key Performance Metrics

Metric	Baselines		Our System	
	YOLOv5+SORT	Siam-MOT	SC: off	SC: on
P	0.8436	0.9758	0.9532	**0.9916**
R	0.3326	0.4253	**0.4776**	0.4634
EDR	0.8867	0.938	0.9542	**0.9639**
FPPI	0.0987	0.0050	0.0111	**0.0018**
IDSPI	0.0872	0.0011	0.0010	**0.0009**
ARE	1.15	0.89	0.87	**0.85**

TABLE II: Comparison of proposed method w/ baselines.

hard false-positives, and the remaining 25% are crops taken around true aircraft locations. False-positive predictions during training with a confidence score over 0.2 are saved for mining and are sampled for training in later epochs to improve the precision of the model. The models were trained on a Tesla P100 GPU with 16GB of GPU memory.

C. AirTrack Hardware

The AirTrack algorithm is designed to run on a custom build SWaP-C hardware platform as shown in Fig. 3. The platform contains six Sony IMX264 cameras spanning a total of approximately 220° FOV horizontally and 48° vertically. The payload uses an NVIDIA Xavier AGX 32GB dev kit to handle the image processing. The payload also contains a 3DM-GQ7 GNS/INSS module with a dual-antenna setup for state estimation and ground truth. The platform is designed to run standalone with an external power source.

V. Evaluation Experiments

To showcase the efficacy of AirTrack, we present results on held-out AOT dataset as well as real world flight tests with large scale aircraft. Comparisons with established baselines, ablation studies and above/below horizon detection performances are provided. We compare out approach to two baseline methods. Siam-MOT based on Siamese multi-object tracking[38] and YOLOv5[39] trained on the AOT dataset. The confidence threshold for prediction was set at 0.5. Table I outlines the key detection and tracking metrics over which we evaluate the methods.

A. Flight Tests

We perform two kinds of flight tests as shown in Fig 3. For the first kind, we install the AirTrack hardware on a general aviation Cessna 182. The aircraft is flown closer to airports and is allowed to interact with cooperative general aviation air traffic. The second set of tests involve a UAS flying towards an intruder helicopter. The UAS and the helicopter start at opposite ends of the runway at different altitudes. They converge and break right to ensure collision avoidance. Post-hoc, we manually label both of these for evaluation.

B. Qualitative Comparative Results

For qualitative evaluations we combine results from held-out AOT and real world flight tests to provide a complete analysis. For the encounter detection rate, an encounter is considered valid if the airborne object is consistently tracked for at least 3 seconds before its range falls below 2000ft. Table II outlines the overall results of the proposed system along with the two baselines. From the table, it can be deduced that YOLOv5+SORT has the worst performance. This is because YOLOv5, which is an excellent *state-of-the-art* object detector in general, is not the most ideal for detecting small objects. Our system outperforms the baselines in all the metrics mentioned earlier. Using the secondary classifier (SC) greatly improves the precision of the system while only taking a slight reduction in the recall.

C. Above/Below Horizon Results

Detecting aircraft above and below the horizon poses significantly different challenges, the latter being the more difficult due to the presence of background clutter. Table III outlines the detection metrics based on *above* or *below* the horizon cases. We can observe that the precision is steady across both domains while the recall is much lower for below-horizon detections being a more difficult task. This is a key area to improve upon in future work. We can also observe the impact of the secondary classifier improving false-positive rejection by reducing the FPPI number by almost 10x. It however does reduce the recall slightly.

D. ASTM Interpretation

We provide and interpret results from the perspective of the ASTM standards. The range error is computed as a fraction of the ground truth range. For the tests we observe that the median range error obeys the allowable error threshold of 15% up to a range of 1.5km. Fig. 4 shows a qualitative range plot for detected aircraft from the real world flight tests for a single sequence. We can observe that the range prediction quality gets worse with increasing object distance, especially with distances of over 1.25km. Fig. 5 shows the probability of track (recall) of the proposed system based on range intervals for the AOT dataset. The probability of track remains more than 95% up to a range of 700m.

Based on our quantitative results, we interpret our results with regard to the ASTM F3442/F3442M standard [5]. This standard was created for unmanned aircraft (UA) with a maximum dimension $\leq$ 25ft, operating at airspeeds below

	Above Horizon		Below Horizon	
	SC: off	SC: on	SC: off	SC: on
P	0.94	**0.99**	0.94	**0.99**
R	**0.63**	0.59	**0.38**	0.36
FPPI	0.0122	**0.0012**	0.0175	**0.0026**
IDSPI	**0.0071**	0.0073	0.0015	**0.0011**

TABLE III: Performance of the proposed system based on metrics calculated separately for above and below horizon scenarios.

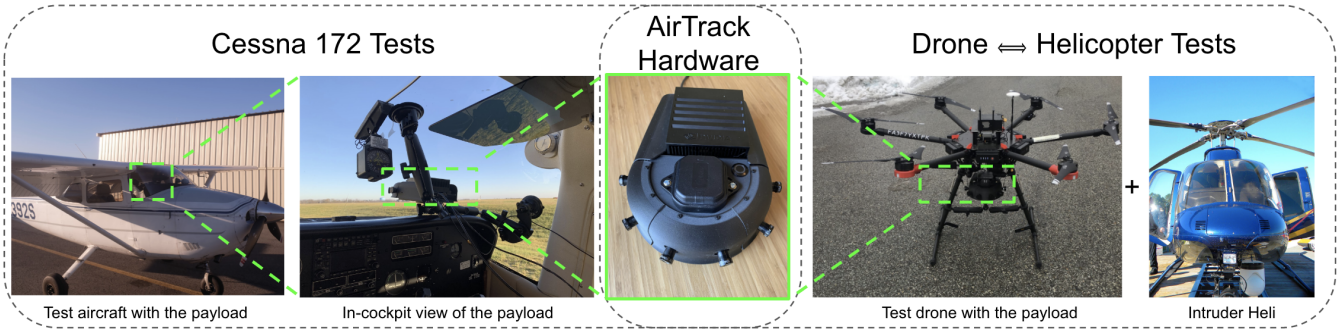

Fig. 3: Overview of the different field tests. Flight tests were performed with onboard SWaP hardware on both a general aviation Cessna 172 performing standard general aviation flight behaviors as well as on a UAS performing dedicated near-collision experiments with a Bell 407 helicopter intruder flying towards the UAS in controlled settings, with both showcasing satisfaction of ASTM F3442/F3442M standards for DAA.

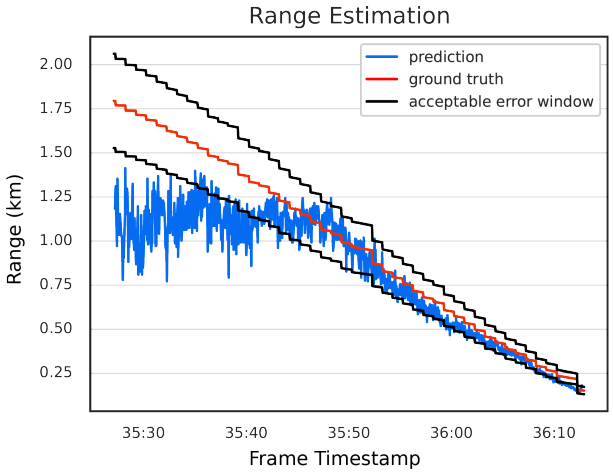

Fig. 4: Range estimation plots compared to ground-truth. We note that the range estimation deteriorates after 1.2km typically. Black line denotes the 15% error margin for interpretation with regards to ASTM F3442/F3442M standard

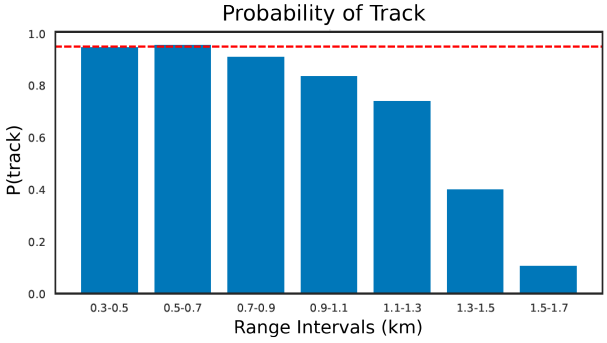

Fig. 5: P(track) vs range. The probability of track (recall) reduces with increasing distance. Dashed red line shows the 95% level. P(track) $\geq$ 95% upto 700m.

100kts, and of any configuration or category. This standard found only range estimation and angular-rate error to be the only two key performance indicators for DAA and specifies a minimum intruder tracking probability of 95%. Although it does not specify how this probability was computed, we interpret it as the tracker recall value (see Fig. 5). Based on the recall value, our system has been shown to have

Ownship Specifications			Min. Required Range
Cruise Speed	**Turn Rate**	**Vertical Speed**	
30 kts	63.05 deg/s	250-500 ft/min	1222m
60 kts	10.51deg/s	250-500 ft/min	963m
60 kts	**31.53 deg/s**	**250-500 ft/min**	**703m** ($\sim$ 700m)
90 kts	7.01 deg/s	250-500 ft/min	1018m
90 kts	**21.02 deg/s**	**250-500 ft/min**	**666m** ($\leq$ 700m)

TABLE IV: Required min. detection range with P(track) $\geq$ 95% based on maximum angular-rate error of 0.9deg/s for different UAS configurations. These values are taken from the ASTM F3442/F3442M standard.

a probability of track of more than 95% up to a range of 700m. We also know our system angular-rate error is within 0.9deg/s. Thus based on these two error metrics, we can use the standard to check which aircraft classes can be used for low risk-ratio operations using our DAA surveillance system. Table IV outlines the minimum track range requirement for different autonomous UAS specs based on a maximum angular-rate error of 0.9deg/s. Based on Table IV we can observe that our DAA surveillance system can satisfy low-risk UA operations for two specification categories:

- 60kts, 31.53deg/s, 250-500ft/min
- 90kts, 21.02deg/s, 250-500ft/min

The performance speed of the vision setup using NVIDIA Xavier AGX dev kit with and without TensorRT are 66.8 FPS and 5.3 FPS, respectively.

VI. Conclusion

We present, AirTrack, a state-of-the-art vision-based detection and tracking module for long-range aircraft detect-and-avoid applications. AirTrack uses cascaded detection modules along with a secondary classifier to improve performance. Comparative results on the Amazon AOT dataset as well as extensive real world flight tests showcase the efficacy of the proposed approach. We also interpret the results for the newly established ASTM standards and prove that AirTrack satisfies the standards for at least two specification categories.

ACKNOWLEDGMENT

The authors would like to thank Jan Hansen-Palmus, Joao Dantas, Ian Higgins, and, David Kohanbash for helping out with experiments, field tests, repair work, and discussions.

REFERENCES

[1] FAA Sponsored Sense and Avoid Workshop, "Sense and avoid (saa) for unmanned aircraft systems (uas)," 2009.

[2] C. Munoz, A. Narkawicz, J. Chamberlain, M. C. Consiglio, and J. M. Upchurch, "A family of well-clear boundary models for the integration of uas in the nas," in *14th AIAA Aviation Technology, Integration, and Operations Conference*, p. 2412, 2014.

[3] C. Muñoz, A. Narkawicz, G. Hagen, J. Upchurch, A. Dutle, M. Consiglio, and J. Chamberlain, "Daidalus: detect and avoid alerting logic for unmanned systems," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, pp. 5A1–1, IEEE, 2015.

[4] A. Mcfadyen and L. M. Alvarez, "A survey of autonomous vision-based see and avoid for unmanned aircraft systems," *Progress in Aerospace Sciences*, vol. 80, pp. 1–17, 2016.

[5] "Standard Specification for Detect and Avoid System Performance Requirements," Standard ASTM F3442/F3442M-20, ASTM International, 2020.

[6] J. Lai, J. J. Ford, L. Mejias, and P. O'Shea, "Characterization of sky-region morphological-temporal airborne collision detection," *Journal of Field Robotics*, vol. 30, no. 2, pp. 171–193, 2013.

[7] D. Dey, C. Geyer, S. Singh, and M. Digioia, "Passive, long-range detection of aircraft: towards a field deployable sense and avoid system," in *Field and Service Robotics*, pp. 113–123, Springer, 2010.

[8] G. Fasano, D. Accardo, A. E. Tirri, A. Moccia, and E. De Lellis, "Morphological filtering and target tracking for vision-based uas sense and avoid," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 430–440, IEEE, 2014.

[9] R. Carnie, R. Walker, and P. Corke, "Image processing algorithms for uav" sense and avoid"," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 2848–2853, IEEE, 2006.

[10] J. Lai, L. Mejias, and J. J. Ford, "Airborne vision-based collision-detection system," *Journal of Field Robotics*, vol. 28, no. 2, pp. 137–157, 2011.

[11] L. Mejias, S. McNamara, J. Lai, and J. Ford, "Vision-based detection and tracking of aerial targets for uav collision avoidance," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 87–92, IEEE, 2010.

[12] J. W. McCandless, "Detection of aircraft in video sequences using a predictive optical flow algorithm," *Optical Engineering*, vol. 38, no. 3, pp. 523–530, 1999.

[13] V. Reilly, H. Idrees, and M. Shah, "Detection and tracking of large number of targets in wide area surveillance," in *European conference on computer vision*, pp. 186–199, Springer, 2010.

[14] F. Schubert and K. Mikolajczyk, "Robust registration and filtering for moving object detection in aerial videos," in *2014 22nd International Conference on Pattern Recognition*, pp. 2808–2813, IEEE, 2014.

[15] A. Rozantsev, V. Lepetit, and P. Fua, "Flying objects detection from a single moving camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4128–4136, 2015.

[16] S. Petridis, C. Geyer, and S. Singh, "Learning to detect aircraft at low resolutions," in *International Conference on Computer Vision Systems*, pp. 474–483, Springer, 2008.

[17] M. F. Fernandez, "Detecting and tracking low-observable targets using ir," in *Signal and Data Processing of Small Targets 1990*, vol. 1305, p. 193, International Society for Optics and Photonics, 1990.

[18] A. Nussberger, H. Grabner, and L. Van Gool, "Aerial object tracking from an airborne platform," in *2014 international conference on unmanned aircraft systems (ICUAS)*, pp. 1284–1293, IEEE, 2014.

[19] J. Lai, J. J. Ford, P. O'Shea, and R. Walker, "Hidden markov model filter banks for dim target detection from image sequences," in *2008 Digital Image Computing: Techniques and Applications*, pp. 312–319, IEEE, 2008.

[20] T. L. Molloy, J. J. Ford, and L. Mejias, "Detection of aircraft below the horizon for vision-based detect and avoid in unmanned aircraft systems," *Journal of Field Robotics*, vol. 34, no. 7, pp. 1378–1391, 2017.

[21] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan, "Aircraft detection by deep convolutional neural networks," *IPSJ Transactions on Computer Vision and Applications*, vol. 7, pp. 10–17, 2014.

[22] S. Hwang, J. Lee, H. Shin, S. Cho, and D. H. Shim, "Aircraft detection using deep convolutional neural network in small unmanned aircraft systems," in *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, p. 2137, 2018.

[23] V. Stojnić, V. Risojević, M. Muštra, V. Jovanović, J. Filipi, N. Kezić, and Z. Babić, "A method for detection of small moving objects in uav videos," *Remote Sensing*, vol. 13, no. 4, p. 653, 2021.

[24] J. James, J. J. Ford, and T. L. Molloy, "Learning to detect aircraft for long-range vision-based sense-and-avoid systems," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4383–4390, 2018.

[25] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6569–6578, 2019.

[26] P. Hu and D. Ramanan, "Finding tiny faces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 951–959, 2017.

[27] D. Liang, Z. Wei, D. Zhang, Q. Geng, L. Zhang, H. Sun, H. Zhou, M. Wei, and P. Gao, "Learning calibrated-guidance for object detection in aerial images," *arXiv preprint arXiv:2103.11399*, 2021.

[28] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 840–849, 2019.

[29] J. James, J. J. Ford, and T. L. Molloy, "Below horizon aircraft detection using deep learning for vision-based sense and avoid," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 965–970, IEEE, 2019.

[30] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468, IEEE, 2016.

[31] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *European Conference on Computer Vision*, pp. 474–490, Springer, 2020.

[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

[33] T. Glozman, A. Narkawicz, I. Kamon, F. Callari, and A. Navot, *A Vision-based Solution to Estimating Time to Closest Point of Approach for Sense and Avoid*. 2021.

[34] AICrowd, "Airborne object tracking challenge."

[35] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3349–3364, 2020.

[36] A. Defazio and S. Jelassi, "Adaptivity without compromise: a momentumized, adaptive, dual averaged gradient method for stochastic optimization," *arXiv preprint arXiv:2101.11075*, 2021.

[37] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017.

[38] B. Shuai, A. Berneshawi, X. Li, D. Modolo, and J. Tighe, "Siammot: Siamese multi-object tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12372–12382, 2021.

[39] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomammana, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Yu, changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai, "ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements," Oct. 2020.