2024 IEEE Aerospace Conference, 2-9 May 2024, Big Sky, MT, USA
DOI:10.1109/AERO58975.2024.10521405

Eyes-Out Airborne Object Detector for Pilots Situational Awareness

Paul BENOIT
Cranfield University
College Rd, Wharley End,
Bedford MK43 0AL, UK
paul.benoit.927@cranfield.ac.uk

Yang Xing
Cranfield University
College Rd, Wharley End,
Bedford MK43 0AL, UK
yang.x@cranfield.ac.uk

Antonios Tsourdos
Cranfield University
College Rd, Wharley End,
Bedford MK43 0AL, UK
a.tsourdos@cranfield.ac.uk

Abstract— **With the exponential development of new flying objects, pilots need to pay even more attention to evaluate their environment, make decisions, and fly safely. Such situation awareness (SA) has multiple codified rules to guarantee the safety of pilots. This paper analyses the feasibility of a portable perception augmentation module (PAM) to help pilots improve their situational awareness based on two key actions on long-distance airborne objects, namely, object detection and distance and trajectory estimation. The developed object detection pipeline based on the state-of-the-art (SOTA) YOLOv8 architecture achieves high accuracy with a mAP50 of 0.835 for objects up to 3000 meters. The inference of the system is 1 second for a 360° scan of the aeroplane surroundings thanks to 4 wide FOV high-resolution cameras. The data used for the model is generated by Airsim in a completely automatized process. The potential implementation of stereo vision and the influence to the PAM are also evaluated. All of these tests are also performed on additional real-life data to evaluate generalization performances, which also show satisfactory results. Efforts in the development of the PAM were made to find the best balance between various constraints such as weight, energy consumption, and accuracy. Characteristic analysis of the PAM such as weight, energy consumption, and accuracy are proposed to seek the optimal balance between various real-world constraints. Real hardware considerations are made to estimate the hardware cost of the PAM based on the simulated results in this study. With further improvement in the trajectory estimation and model generalization, a prototype could be made, deployed, and sold to recreational pilots for safer flights. The code and data are available on: https://github.com/Alcharyx/IRP-Eye-out/**

Keywords— **Eyes-Out, Airborne Object Detection, Situational Awareness, YOLOv8, Portable Module, Near-Real-Time**

TABLE OF CONTENTS

1. INTRODUCTION 1
2. RELATED STUDIES 2
3. METHODOLOGY 3
4. SYSTEM DESIGN 3
5. RESULTS 4
6. DISCUSSION AND FUTURE WORKS 6
7. CONCLUSION 7
ACKNOWLEDGEMENT 7
REFERENCES 7
BIOGRAPHY 8

1. INTRODUCTION

While flying in an aeroplane, a pilot needs to be highly focused on many different elements: commands, communication, environment, etc. According to Endsley's SA theory [1], human SA can be classified into three levels, which are 1) environment perception, 2) comprehension, and 3) prediction. There are many reasons for the reduce or loss of SA such as fatigue, distractions, stress-inducing circumstances, excessive work demands, lapses in vigilance, inadequately communicated information, and flawed mental models, etc. The loss of SA can lead to small damages to the plane but also to a crash in the worst-case scenario. The current data reveals that light aircrafts are responsible for 93% of the death in the EU in 2021 [2]. In addition to a constant 360° surveillance, it can be challenging for a pilot to estimate the distance and speed of another aircraft. Indeed, an aircraft would appear as an immobile black dot until it starts blooming at around 500 meters [3]. Lastly, during a flight, some manoeuvres are riskier than others, especially during "approach", "landing" and "take-off" [4]. This is also partially due to the high air traffic concentration around airports. All of these elements justify the need for an efficient SA support tool for pilots specifically during the critical phases.

This study assesses the feasibility of a system designed to enhance detection for these smaller airplanes, and give more time for pilots to react, thus reducing risks of collision. This study will focus on optical solutions. The evaluation of the system is based on multiple key metrics such as range, accuracy, inference, hardware, and cost. The system can be divided into four subsystems: 360° scan, object detection, distance estimation, trajectory estimation, and Graphical User Interface (GUI). The main contributions of this paper are the following:

- An automatized tool to generate realistic airborne object detection dataset using Airsim [5] with labels and additional data parameters logs.
- Object detection Artificial Intelligence (AI) model for flying objects up to 3000 meters with a very low rate of False Positive Per Image (FPPI).
- Stereo Vision evaluation and limitations for this type of module.

© 2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This study is organized as follows: Section II reviews the current literature on airborne object detection. Section III explains the approach employed to establish the framework of the system. Section IV focuses on the specific design of the system, including the dataset generation, object detection model, stereo vision, and hardware considerations. The results are presented in section V and then discussed in section VI along with potential improvements to the system. Finally, section VII concludes this study.

2. Related Studies

Flying object detection is a highly researched topic. For example, the Unmanned Armed Vehicles (UAVs), helicopters, aeroplanes, and birds are all part of the airborne ecosystem. Many sensors and detection techniques are available currently and their advantages/constraints are discussed below.

A. Radar-based systems

Radar is widely used to detect objects up to several kilometres, depending on the radar, the environment, and the object to detect. A radar is barely subject to weather conditions but has a limited resolution, resulting in a high rate of False Positive (FP) and poor identification [6]. Radars are also costly, complex, and under a lot of regulations for their usage, making the development of a system more complicated [7]. The most common radar implementations are ground-based applications for security surveillance of airspace. Many companies offer these types of technologies, but high-resolution light radar modules are harder to find. Long LiDAR system could also provide point cloud mapping up to two kilometres [8] and could also provide data for distance estimation.

B. Radio Frequency-based systems

An AI model can listen to Radio Frequency (RF) spectrum and identify UAVs or aeroplanes, but this comes with many limitations. First, this type of detector is not able to identify birds and is very sensitive to noisy environments [9]. Even if the range can be up to several kilometres, the noise might make the data unexploitable at such a far distance. The RF detector can identify many types of drone radio frequency: Wi-Fi, Bluetooth, cellular, radio control, etc. but also aeroplane RF which are on other frequency bandwidths. Allahham et al. proposed an AI approach for drone detection and identification with a 1-D Convolutional Neural Network (CNN) [10] using the RF spectrum of drone activity, with high performance for detection and identification. The main issue is that it is generally challenging to precisely identify the position of an emitting drone, making this application not adapted for the system. Another option to increase the precision and the range of the detection is to fuse it with other sensors, as explained in [11]. However, this multiple-sensor solution might be heavy in terms of computing power, price and actual weight, which is not suitable for this application.

C. Single Camera-Based Systems

Many researchers focus on a single camera detector such as in [12], using data from a small UAV camera and motion cues to help the detection of other flying objects, the stabilisation of the background, and the classification of 3D descriptors from spatiotemporal image cubes. Another more recent approach uses the state-of-the-art image detector Yolov8 [13], creating a feature activation map for the classifier to identify flying objects. For the system to estimate the trajectory and speed of the classified object, the movement of both the object and the camera must be considered. It is important to take into account its movement and rotation to track and estimate the direction of the detected object. Kim and Yow [14] calibrate the drone camera, using the disparity formula and a global coordinate transformation to accurately predict the trajectory of an object. Although considered a single-camera system, this type of system would still require at least four cameras with a high enough Field of View (FOV) to cover all the angles around the aeroplane.

D. Stereo Camera Based Systems

Stereo cameras allow a spatial representation of the detected object and track it within the shared FOV of the two cameras, as explained in [15]. It is also possible to estimate distances by using the correct equation for high-precision data based on different situations [16]. With a rectified system it is possible to get a more a general equation but because the amount of precision needed being inversely proportional to the distance of the object it isn't very efficient for long range applications [17]. Thanks to distance estimation, the pilot will be able to increase their Situational Awareness and anticipate dangerous situations.

E. 360° Camera Based Systems

Object detection using a 360° camera is relatively new. There is a huge lack of data for this type of image. On top of that, the existing camera module resolution is not high enough (maximum 40 Megapixels), considering the 360° FOV which gives a poor pixel resolution compared to a four-high-resolution camera system, resulting in a loss of precision. One option could be the construction of a 360° image using multiple high-resolution cameras and image stitching techniques, to only run the AI model once for all the panorama, speed up inference, and avoid any cropped image. Nevertheless, a 360° detector is not as simple to create as a 2D detector. Because of the distortion in 360° images, the bounding boxes of objects are not rectangles anymore. This slightly changes the classic Intersection over Union (IoU) technique. As discussed in [18], there are two ways of viewing 360° images, either with an Equirectangular Projection (ERP) or with a sphere view. Different IoU techniques are applicable. Considering the scale of this project, developing a 360° solution from scratch would be very time consuming.

F. Dataset

Many datasets of flying objects and images taken from the air are available on the internet. But they often include only one

type of object such as aeroplanes, UAVs, or birds and those are relatively close (< 200 meters). Ideally, for this model, the dataset needs to be composed of multiple objects, at a far distance. The only one found that fits the requirement for this project was the Airborne Object Tracking (AOT) dataset [19]. It was used for a challenge organized by Amazon Prime Air. It is made of multiple grayscale images from flight sequences collected onboard aerial vehicles with high-resolution cameras (2448x2048). Objects from different distances and different sizes are identified and the dataset is well documented, although quite heavy (11TB) and difficult to manipulate.

3. Methodology

The objective of this system is to be fast, lightweight, and accurate. Three common flying objects were picked to be detected: Airplane, Helicopter, and Balloon. The module is composed of four camera angles (front, left, back and right) and processes every frame into a live pipeline to detect all the flying objects around the aeroplane. Using the predictions, it estimates the distance and trajectory of each object to then send notifications to the GUI of the pilot. The camera images are sent through Mobile Industry Protocol Interface (MIPI) protocol to the computer which then uses Sliced Aided Hyper-Inference (SAHI) [20] to split the image into four quadrants, run the object detection model, fuse the results, and send the final prediction to the distance estimator. All the blocks are running in parallel for maximum efficiency, which also allows the computer to prioritize certain angles during the scan when an object is detected with a high collision risk value. One of the main objectives of this study was to determine the feasibility of different techniques and hardware for this type of module. Fig 1. shows the general system architecture as follows:

1. *Get the camera images from all angles: In this step, the camera feed doesn't need to be a high frame per second video, images are enough thanks to the high detection range giving time for aeroplane to appear.*
2. *Run SAHI: SAHI starts by splitting images with an overlap (4 subimages with 20% overlap in the current architecture) to reduce the RAM needed during inference to load the images.*
3. *Run the YOLOv8 inference: This step is still part of SAHI since all subimages are fed to YOLOv8 for detection. All the result are then fused into a final prediction.*
4. *Estimate trajectory and distance (to be developed): This step adds new computation based on the prediction result, allowing the system to assign a risk level to each flying object detected. This can lead to prioritizing high risk situations and give more information to the pilot.*
5. *Notify the pilot through the GUI: Thanks to the GUI, the pilot can easily access the model detections with as much data as possible to help in the decision-making process.*

4. System Design

A. Dataset Generation

The dataset is composed of multiple high-resolution grayscales simulated images with a high horizontal Filed of View (FOV) of 102°. The grayscale transformation was designed in [21] which gives the perceived brightness and intensity information of the light emitted or reflected from a surface:

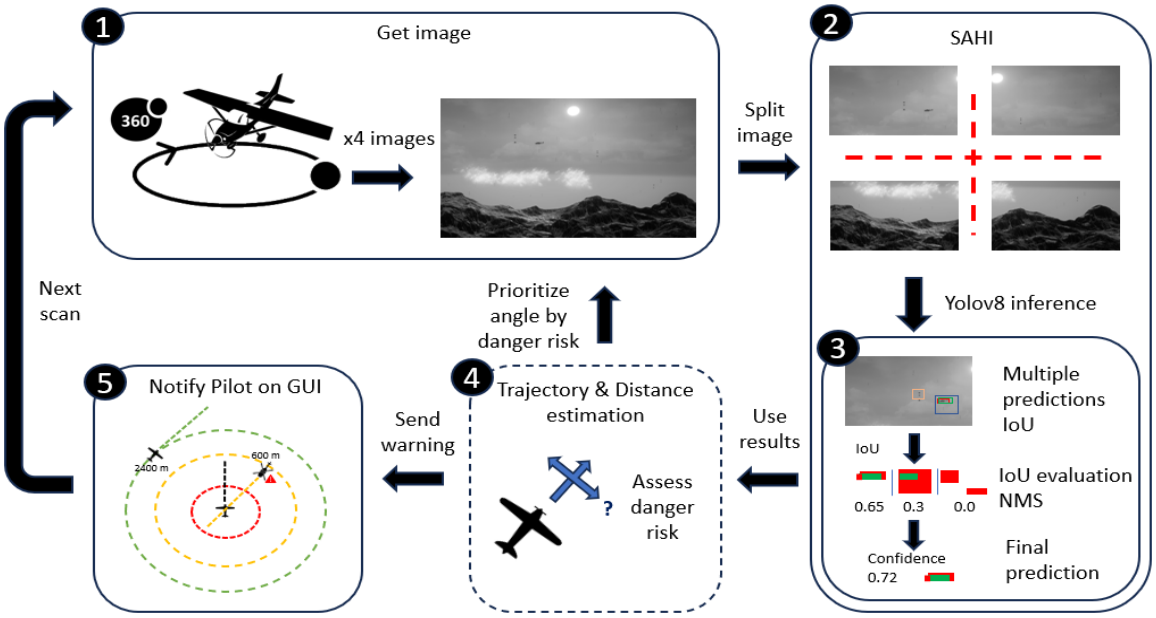


Figure 1. Overall system architecture 1) 360° scan by four high-resolution cameras. 2) Sliced Aided Hyper-Inference, processes images one by one, slices into four quadrants with overlap. 3) Yolov8 inference for object detection, making multiple guesses, using NMS to keep the best one as the final prediction. 4) Distance and trajectory estimation (not implemented yet). 5) Communication with 3D Python GUI to notify the pilot. All steps run in parallel, camera angle monitoring are prioritised depending on the danger risk.

$$Y = 0.2125 * R + 0.7154 * G + 0.0721 * B \quad (1)$$

with R, G, and B the pixel value of each colour channel. This transformation removes colour information that could have caused overfitting due to the limitation of vehicles colours in Airsim. The simulated camera on Airsim has the following specifications: 4608x2592, Horizontal FOV 102°, and sensor size: 23.76x13.365mm, which nearly reproduces the Raspberry PI camera 3 wide specifications [22]. To generate the data, all the vehicles in the simulation are assigned a random path and are then teleported frame by frame to the coordinate where they would have been 150 meters further. Multiple parameters such as the weather, the landscape, the vehicle orientation, the time of the day, etc. are changed in each frame to increase image diversity. Range limitations can be applied to only allow flying objects between 1000 and 3000 meters for example. Images are then ordered in a log file in JSON format. Each image is split into four quadrants to generate similar images to what the model using SAHI will be receiving as input. Then the images are ordered again to avoid duplicates due to the quadrant overlapping. Finally, the train, validation, and test sets are created.

B. Object Detection

For object detection, the latest version of YOLO (8th) seemed adapted, thanks to the flexibility of the library and of the model architecture. The small model of YOLOv8 provides a good compromise between speed accuracy and weight [23]. Due to the high resolution, loading the full images in the GPU VRAM was too heavy for a portable solution (over 10 GB with a small model). The goal is to use a model weighing below 4GB so that it is compatible with an NVIDIA Jetson Nano.

With SAHI, it is possible to split the images into smaller ones, run the inference on each of them and combine the results. The use of SAHI has a cost in terms of inference. However, the module does not need to scan the sky at a very high frame rate, so it is adapted for this application. The model evaluation is based on two criteria: False Positive Per Image (FPPI) and mean Average Precision (mAP). On one hand, FPPI must be very low for the pilot to trust the module without receiving false alarms. On the other hand, the mAP must be high since it demonstrates the accuracy of the model. On top of evaluating using a new test data split, the AOT dataset is used to evaluate the model performances in a realistic environment.

C. Distance Estimation

For the distance estimation, the goal was to determine if stereo vision was adapted for the module. Stereo vision aims to compute the disparity of the whole image to create a disparity map and evaluate distances. The main issue arises for high distances: the precision must be very high because a sub-pixel error can induce 1km of error in the estimation if the object is far away. This is due to the disparity being inversely proportional to the distance as mentioned in [24].

There are multiple steps to compute disparity on stereo images. At first, the images need to be rectified (this step can be skipped if the cameras are parallel), and then the pixel disparity where the object is detected on the image should be evaluated. This is possible using algorithms such as Block Matching (BM) or Semi-Global Block Matching (SGBM) [25]. SGBM is more accurate but takes more time to compute. These algorithms have multiple parameters that need to be fine-tuned, especially the "block size" which is the size of the matching window used to find corresponding points, and the "number of disparity" which is the search range for disparity values. Once the disparity is computed, values of the pixel area where the object is detected must be used in the following formula:

$$Distance = \frac{N * B}{2 * disp * \tan\left(\frac{FOV}{2}\right)} + f \quad (2)$$

with N the total number of pixel width on the image, B the baseline distance between the two cameras, disp the disparity value in pixels between the right and left images, FOV the value of the horizontal FOV in radians, and f the focal length of the camera in meters [16] [17].

Mono-camera solutions also exist. Although they are expected to have worse performances than the stereo vision ones, they do not require additional hardware and would be a very efficient solution. A simple CNN could be trained by creating a dataset on Airsim, thanks to the "planar vision" camera.

D. Hardware Considerations

Even if this project is completely simulated, hardware considerations were made to ensure its feasibility in terms of computational power, price, and hardware specifications. The expected computer for the module would be a NVIDIA Jetson Nano which has a 4GB GPU and MIPI CSI-2 connector for up to 4 cameras. The camera configuration is based on the Raspberry PI camera module 3 wide with high-resolution and high FOV. To allow the four cameras to be connected to the Jetson Nano, additional hardware is needed, such as a hardware attached on top (HAT) like the Arducam Jetson camera multiplexer [26]. Taking only this hardware into account, this would already cost around £500 for the module. Additional costs might be added for the wiring and sensors such as an accelerometer.

5. RESULTS

A. Dataset Generation

The best way to evaluate a dataset is through the object detection model performances. The process to improve the dataset generation was very iterative: training a model, looking at predictions, and ordering them based on parameters (weather, distance, time of the day) to evaluate their influence. The results were finally used to improve the dataset and thus the model. Some parameters such as the weather appeared to have a negligible influence but still contribute to the diversity of the dataset.

Regarding the generation, it takes three hours to fully create the dataset with around 3000 images and 7500 flying objects. More could have been generated, but it would have made the

training longer and increased the risks of overfitting. The dataset shows a lot of diversity thanks to all the different parameters and is perfectly balanced between all classes. The labelling is precise and there are very few errors. One issue is the unbalanced distance distribution: there are more "close" objects than "far" ones, which can create a bias during the training. Also, in the original dataset, the objects are mostly positioned in the middle of the image, across its entire width. However, since all the images are split into quadrants, this makes the object positioned at the top or bottom of the image across its entire width as shown in Fig. 3 C. This is partially solved by using mosaic augment during the training.

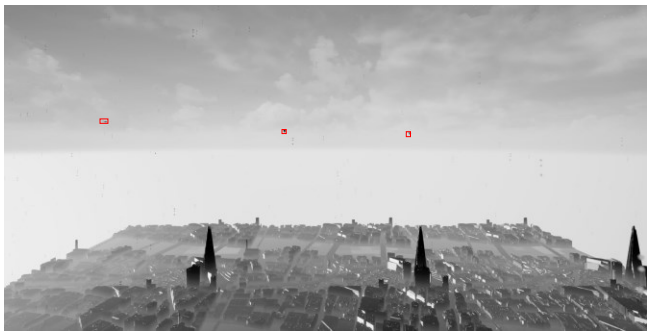


Figure 2. Dataset Image example with the bounding box before split (one helicopter and two balloons, raining slightly).

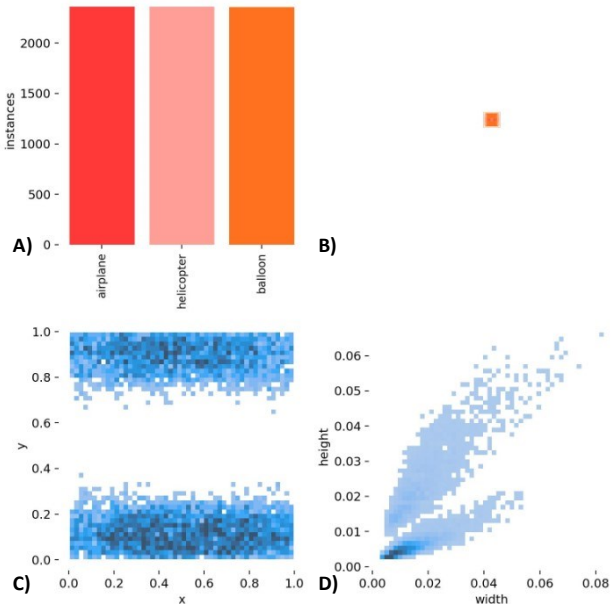


Figure 3. Dataset details A) Class balance. B) Label shape repartition. C) Label position repartition. D) Label size repartition.

B. Object Detection

Multiple models were generated using different base architectures for YOLOv8 (nano, small, medium), with multiple hyperparameters combination, e.g., the usage of SAHI and different datasets. The method used for optimisation is straightforward: fine-tune hyperparameter values one by one to get the best results and then combine all the best parameters with the best dataset. In the end, the best model YAML configuration with the use of SAHI got the following performances on the test set.

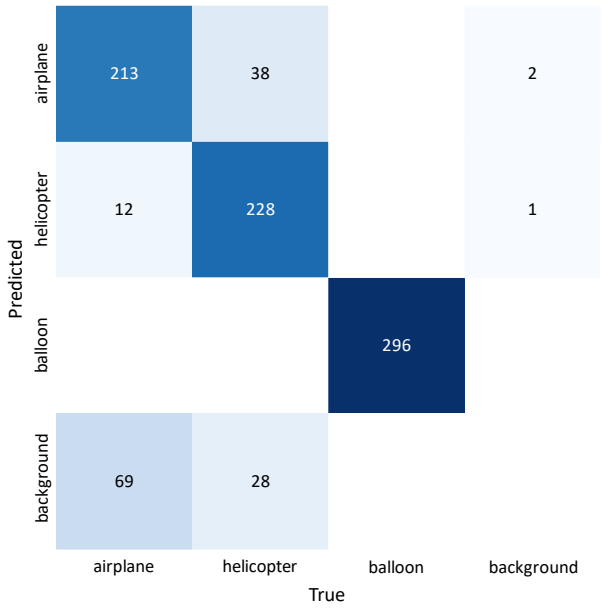


Figure 4. Confusion Matrix of Best model using SAHI.

TABLE I. Model Validation Metrics

Class	*Box*	*Recall*	*mAP50*	*mAP50-95*	*FPPI*
All	0.911	0.778	0.835	0.554	1e-4
Airplane	0.834	0.613	0.668	0.388	1e-3
Helicopter	0.909	0.737	0.823	0.508	2e-4
Balloon	0.991	0.958	0.994	0.776	0

The results are very good overall. The main issue is the big gap in performances between Airplanes and Balloons. This makes sense, because balloons are way bigger than the two other classes, but it might create another bias in the model. To improve that, it is possible to increase the balance of Airplanes and reduce Balloons to force the model to detect them to get a good accuracy.

If the identification of the objects is not the main focus, then it is also possible to create the model for a single class and get improved accuracy but a higher FPPI:

TABLE II. Single Class Model Validation Metrics

Class	*Box*	*Recall*	*mAP50*	*mAP50-95*	*FPPI*
Single Class	0.937	0.881	0.993	0.600	3e-4

Fig 5. shows the performance of the model based on the distance of the predicted object. There are three outcomes for a prediction:

- True Positive (TP): The prediction is above the confidence threshold and above the IoU threshold for one of the ground truth bounding boxes.
- False Positive (FP): The prediction is above the confidence threshold and below the IoU threshold for all ground truth bounding boxes.

- False Negative (FN): No prediction for a ground truth bounding box.

There is a clear difference in accuracy between the 1000 to 1500 meters range and the 2500 to 3000 meters range. The unequal distribution of distances mentioned earlier can also be seen.

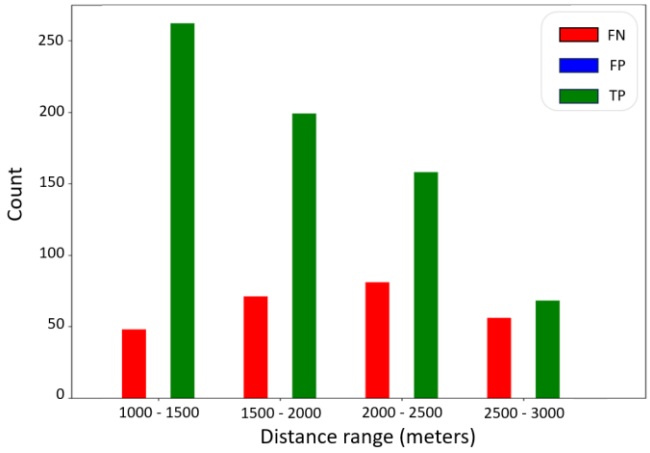


Figure 5. Prediction result based on distance.

C. Distance Estimation

Using stereo vision on the ground truth bounding box coordinates gave very accurate results. Even at high distances such as 3000 meters, the error rate was below 1%. Tests using the predicted bounding boxes were not as good and very random. Results using BM and SGBM were good for the detection of movement, but the disparity values were not accurate with a lot of holes. To get better results, the disparity was computed on multiple Region Of Interest (ROI) of the image, determined by the area where the object was detected as seen in [27]. This reduced computational time and slightly improved accuracy.

Figure 6. Disparity map using ROI at 300 meters.

The disparity map was filled with holes and a hole inpainting technique was used to fill areas where the algorithm could not determine the disparity, as shown in Fig. 7. Although visually better, the values were still not coherent over 300 meters, because of the level of precision required. More resilient inpainting techniques [28] combined with super-resolution were looked into to improve the results, but they made the distance estimation heavier in term of computational power.

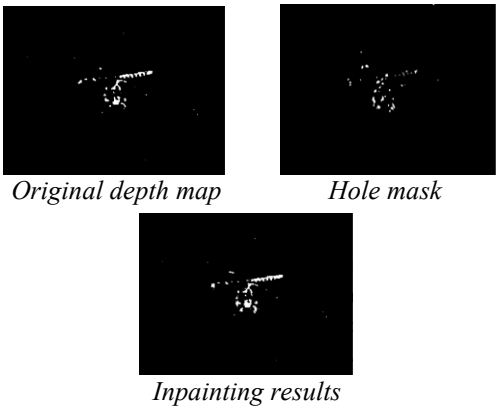


Figure 7. Inpainting process.

6. Discussion and Future Works

The dataset and the model are highly correlated in terms of results. Although they seem to give high-quality predictions with the test dataset, the model struggles a bit more with the real-life data from the AOT dataset. The AOT images do not have the same resolution as the training images and are composed of a higher variety in terms of vehicle and landscape. These results highlight a generalisation problem from the model, which is due to overfitting. Even if they are very realistic, Airsim images are still far from real ones. There are plenty of ways to improve the performances, but one key idea would be to use a second validation set composed of AOT images during the training, allowing constant feature learning. The diversity of vehicle 3D models and colours could also be increased in the dataset. The performances for the distance also showed lower results at high distances, which is expected since the objects are smaller, but it could be due to a model bias as mentioned earlier.

Stereo vision could be improved by increasing the baseline length up to a physical limit because the module is supposed to be small and portable. It is also possible to virtually increase the baseline by rotating the camera angle, at the cost of reducing the common FOV. All these solutions do not seem sufficient to estimate accurately at far distances.

Distance estimation is only the first step before trajectory estimation. Another way to estimate distance could be to use the pixel size of an identified object on the image, compared to its actual average size in real life. This might not result in the best precision. However, if there is a 300-meter error at 2 kilometres, this would be good enough for the application. As mentioned earlier, an accelerometer might be needed to evaluate the movement of the aeroplane and consider it to estimate trajectory. Time series might also be useful for some techniques, such as optical flow explained in Mora Granillo and Beltrán paper [29], which could add a mask on top of the image for easier detection and estimation.

Custom hardware could be interesting for this module, considering that the HAT for the four cameras connection

enables streaming of camera images. The system inference is not fast enough to use images at a high frame rate, and a smart switch could be more adapted to send the camera images one by one without hindering performances.

7. CONCLUSION

In this paper, the system architecture was designed for an Aircraft Situational Awareness Eyes-Out module, the data generation was automatised for a long-distance airborne object detection model, the model was trained, and different optical techniques were analysed to estimate distances. The model accuracy is high and the FPPI low, which allows the pilot to rely on the system without questioning the predictions. As for distance estimation, stereo vision does not seem suitable for the application. Other techniques seem promising and should be studied in more detail.

There is still room for improvement before building a more complete pipeline. Nonetheless, this project has established the feasibility of an Aircraft Situational Awareness Eyes-Out module, the potential costs of such a system for a private pilot, and technical barriers that need to be overcome by further research. All of the code for the data generation and model evaluation are available on GitHub.

ACKNOWLEDGEMENT

We would like to thank Haydn Thompson from THHINK Ltd our industrial partner during this project for his support and creative feedback along this project. We would like to express our sincere gratitude to Cranfield University who gave us access to Crescent2 supercomputer facilities during this research.

REFERENCES

[1] Endsley, Mica R. "Toward a theory of situation awareness in dynamic systems." *Human factors* 37.1 (1995): 32-64.

[2] “Air Safety Statistics in the EU.” n.d. Accessed August 5, 2023. https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Air_safety_statistics_in_the_EU.

[3] “When EVERY second COUNTS”, UK Airprox Magazine, 2017 https://www.airproxboard.org.uk/media/ix0h2diw/lookout-2017-article.pdf

[4] “Accidents by flight phase.” n.d. Accessed August 5, 2023. https://accidentstats.airbus.com/statistics/accident-by-flight-phase.

[5] AirSim: Open-Source Simulator for Autonomous Vehicles Built on Unreal Engine / Unity, from Microsoft AI & Research. n.d. Github. Accessed August 5, 2023. https://github.com/microsoft/AirSim.

[6] Alencar, Francisco AR, et al. "Fast metric tracking by detection system: Radar blob and camera fusion." *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*. IEEE, 2015.

[7] RA 2307 – Rules of the Air Issue 9 UK Ministry of Defence and Military Aviation Authority. 2021 [Online]. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/987940/RA2307_Issue_9.pdf

[8] Dalbah, Yahia, Jean Lahoud, and Hisham Cholakkal. "RadarFormer: Lightweight and accurate real-time radar object detection model." *Scandinavian Conference on Image Analysis*. Cham: Springer Nature Switzerland, 2023.

[9] Ingala, Dominique GK, and Serendra Reddy. "Baseline Measurement of the Environmental RF Noise Floor in the ISM Bands." *2022 Conference on Information Communications Technology and Society (ICTAS)*. IEEE, 2022.

[10] Allahham, Mhd Saria, Tamer Khattab, and Amr Mohamed. "Deep learning for RF-based drone detection and identification: A multi-channel 1-D convolutional neural networks approach." *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. IEEE, 2020.

[11] Svanström, Fredrik, Fernando Alonso-Fernandez, and Cristofer Englund. "Drone Detection and Tracking in Real-Time by Fusion of Different Sensing Modalities." *Drones* 6.11 (2022): 317.

[12] Rozantsev, Artem, Vincent Lepetit, and Pascal Fua. "Flying objects detection from a single moving camera." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[13] Reis, Dillon, et al. "Real-Time Flying Object Detection with YOLOv8." *arXiv preprint arXiv:2305.09972* (2023).

[14] Yow, Kin-Choong, and Insu Kim. "General Moving Object Localization from a Single Flying Camera." *Applied Sciences* 10.19 (2020): 6945.

[15] Avalos, Maria Nieves Brunet. *Stereo vision-based system for detection, track and capture of intruder flying drones*. Diss. Purdue University, 2020.

[16] Tjandranegara, Edwin. "Distance estimation algorithm for stereo pair images." *ECE Technical Reports* (2005): 64.

[17] Xiyan, et al. "Distance measurement system based on binocular stereo vision." *IOP Conference Series: Earth and Environmental Science*. Vol. 252. No. 5. IOP Publishing, 2019.

[18] Cao, Miao, Satoshi Ikehata, and Kiyoharu Aizawa. "Field-of-view IoU for object detection in 360° images." *IEEE Transactions on Image Processing* (2023).

[19] “AIcrowd.” n.d. Airborne Object Tracking Challenge. Accessed August 5, 2023. https://www.aicrowd.com/challenges/airborne-object-tracking-challenge.

[20] Akyon, F. C., et al. "SAHI: A lightweight vision library for performing large scale object detection and instance segmentation." (2021).

[21] Poynton, Charles. 1997. “Frequently Asked Questions about Color.” https://poynton.ca/PDFs/ColorFAQ.pdf.

[22] Raspberry Pi Documentation, Camera. 2023. https://www.raspberrypi.com/documentation/accessories/camera.html

[23] Ultralytics. YOLOv8 in PyTorch. Github, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[24] Kaehler, A., and G. Bradski. n.d. “Learning OpenCV 3 O’Reilly Media Sebastopol.” California, Dec.

[25] Hirschmuller, Heiko. "Stereo processing by semiglobal matching and mutual information." *IEEE Transactions on pattern analysis and machine intelligence* 30.2 (2007): 328-341.

[26] Arducam Quadroscopic Camera Bundle Kit Technical documentation, https://www.uctronics.com/arducam-1mp-4-quadrascopic-camera-bundle-kit-for-raspberry-pi-nvidia-jetson-nano-xavier-nx.html

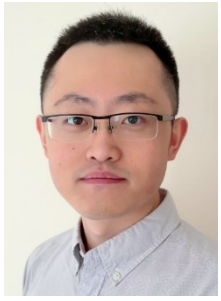
[27] Hamzah, R. A., K. A. A. Aziz, and A. S. M. Shokri. "A pixel to pixel correspondence and region of interest in stereo vision application." *2012 IEEE Symposium on Computers & Informatics (ISCI)*. IEEE, 2012.

[28] Satapathy, Sukla, and Rajiv Ranjan Sahay. "Robust depth map inpainting using superpixels and non-local Gauss–Markov random field prior." *Signal Processing: Image Communication* 98 (2021): 116378.

[29] Granillo, Omar Daniel Mora, and Zizilia Zamudio Beltran. "Real-time Drone (UAV) trajectory generation and tracking by Optical Flow." *2018 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*. IEEE, 2018.

BIOGRAPHY

Paul Benoit *is a Computer Science and Microelectronics Engineer from École des Mines de Saint-Étienne. He joined the Applied AI MSc program at Cranfield University in 2022 to pursue a double degree and specialize in AI. His primary interests lie in end-to-end pipeline development using artificial intelligence, deep learning, and computer vision. He aspires to begin his professional career as an AI Embedded Software Engineer.*



Yang Xing *(Member, IEEE) became a Lecturer in Applied AI at Cranfield University in 2021 where he received his Ph.D. degree in July 2018. His research interests focus on artificial intelligence, deep learning, computer vision, human-autonomy collaboration, and autonomous vehicles, where he has contributed 2 books and over 80 papers (including 2 ESI highly cited papers) on high-quality peer-reviewed journals and conferences. Dr. Yang Xing serves as an Associate Editor in IEEE Trans. on Intelligent Vehicles. He won the Best Workshop/Special Session Paper on IEEE IV 2018.*

Antonios Tsourdos *(Member, IEEE) received a Ph.D. degree in nonlinear robust autopilot design and analysis from Cranfield University, Cranfield, U.K., in 1999. He joined Cranfield University in 1999 and currently serves as the Director of Research - Aerospace, Transport and Manufacturing and Head of Centre for Autonomous and Cyber-Physical Systems. He conducts basic and applied research in the fields of guidance, control, and navigation for single and multiple unmanned autonomous vehicles as well as research on cyber-physical systems.*

https://dspace.org/

Eyes-out airborne object detector for pilots situational awareness

Benoit, Paul; Xing, Yang; Tsourdos, Antonios

2024-05-13
Institute of Electrical and Electronics Engineers (IEEE)
https://doi.org/10.1109/AERO58975.2024.10521405
Benoit P, Xing Y, Tsourdos A. (2024) Eyes-out airborne object detector for pilots situational awareness. In: 2024 IEEE Aerospace Conference, 2-9 March 2024, Big Sky, USA

Downloaded from a DSpace repository. https://dspace.org/