

Article

Aerial Vehicle Detection Using Ground-Based LiDAR

John Kirschler and Jay Wilhelm *

Mechanical Engineering Department, Russ College of Engineering, Ohio University, Athens, OH 45701, USA; jk553916@ohio.edu

* Correspondence: wilhelj1@ohio.edu

Abstract

Ground-based LiDAR sensing offers a promising approach for delivering short-range landing feedback to aerial vehicles operating near vertiports and in GNSS-degraded environments. This work introduces a detection system capable of classifying aerial vehicles and estimating their 3D positions with sub-meter accuracy. Using a simulated Gazebo environment, multiple LiDAR sensors and five vehicle classes, ranging from hobbyist drones to air taxis, were modeled to evaluate detection performance. RGB-encoded point clouds were processed using a modified YOLOv6 neural network with Slicing-Aided Hyper Inference (SAHI) to preserve high-resolution object features. Classification accuracy and position error were analyzed using mean Average Precision (mAP) and Mean Absolute Error (MAE) across varied sensor parameters, vehicle sizes, and distances. Within 40 m, the system consistently achieved over 95% classification accuracy and average position errors below 0.5 m. Results support the viability of high-density LiDAR as a complementary method for precision landing guidance in advanced air mobility applications.

Keywords: computer vision; LiDAR; advance air mobility

1. Introduction

In the United States, there are currently 19,533 airports with 750 having ground-based Radio Detection And Ranging (RADAR) systems. Advanced air mobility (AAM) may rely on smaller regional airports without ground-based aerial monitoring that are dependent on other systems such as ADS-B [1]. Assisted landing systems will be essential as AAM continues to expand into urban areas with vertiports that lack low-altitude aerial vehicle tracking to ensure safe operations. RADAR, Radio navigation, and GNSS based landing systems may not scale for small AAM vehicles operating in urban conditions near tall buildings [2–6]. Instead, Light Detection And Ranging (LiDAR) sensors placed on vertiports using advanced object-detection data processing could allow for ground-based AAM vehicle tracking. Vehicle position during terminal operations could be wirelessly sent back and utilized for landing assistance [7]. LiDAR sensing would allow multiple vehicles to be tracked and enable operations in urban environments with non-metallic vehicle airframes. Enabling a ground LiDAR tracking and landing assistance system would require development of new methods to process point-cloud data along with an understanding of how to choose laser density with respect to distance and vehicle size. This paper takes the first step in investigating data-processing techniques for detecting and ranging to small UAS and AAM sized vehicles and evaluating performance with a range of different LiDAR sensors that could work with a future landing system.

Effective landing guidance will be critical as AAM operations expand near vertiports [8]. Traditional tracking and navigation systems such as RADAR, Automatic De-



Academic Editor: Yan (Rockee) Zhang

Received: 1 July 2025

Revised: 19 August 2025

Accepted: 20 August 2025

Published: 22 August 2025

Citation: Kirschler, J.; Wilhelm, J. Aerial Vehicle Detection Using Ground-Based LiDAR. *Aerospace* **2025**, *12*, 756. <https://doi.org/10.3390/aerospace12090756>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

pendent Surveillance—Broadcast (ADS-B), Instrument Landing Systems (ILS), Microwave Landing Systems (MLS), and Global Navigation Satellite Systems (GNSS) have been widely used at major airports for decades [4–6,9,10]. However, their performance degrades in low-altitude, congested environments. RADAR systems struggle to detect low-reflectivity vehicles, especially those with non-metallic frames, and often exhibit positional errors greater than 3 m [3,9,11]. One study examined a ubiquitous RADAR system that detected a DJI Phantom 4 drone with 99% accuracy from 0 to 1000 m, but reported a standard deviation of 0.64 m in range error. This error magnitude exceeded the length of the drone and may limit the system's suitability for dense airspace navigation [12]. ADS-B and RADAR fusion methods can improve accuracy by combining heading and positional data [2], but they depend on cooperative transmitters. MLS calculates 3D position using azimuth, elevation, and distance signals [4], while ILS uses localizers and glide slope transmitters for fixed-path descent [5]. Both systems support low-visibility landings but were designed for runway-based fixed-wing aircraft rather than vehicles using varied approach paths [13]. GNSS-based landing systems provide Category III precision [10], although signal quality is reduced in urban environments due to multipath interference and line-of-sight obstructions [14].

Computer vision offers an alternative sensing method for object detection and landing assistance. The YOLO (You Only Look Once) algorithm applies convolutional neural networks to 2D images to predict object class and bounding boxes with high inference speed and accuracy [15,16]. Slicing-Aided Hyper Inferencing (SAHI) can decrease the YOLO network training time by training the model at a lower native resolution while still allowing high resolution image inferencing without down sampling to a lower resolution [17]. This is accomplished by slicing the large image into sections equal to the native resolution, followed by recombining the sections after inferencing to the original higher resolution [17]. Fiducial marker systems such as ArUco allow onboard cameras to estimate relative position and orientation to a landing target using QR-like codes affixed to the surface [18,19]. These methods require the aerial vehicle to approach from directly above and maintain a clear line of sight, which may not align with all approach profiles.

Modern LiDAR sensors rotate 360 degrees while emitting infrared light at known azimuth and elevation angles [20]. Using the return ray, a complete scan of the environment is assembled which is represented by x, y, z coordinates relative to the sensor [20]. Additionally, the strength of the ray provides a 4th component of intensity which is based on material properties and object orientation. LiDAR-based detection algorithms typically consist of pre-processing steps followed by a neural network that extracts objects and allows for classification and position estimation based solely on point-cloud data [21–23]. Complex-YOLO is a LiDAR-based detection algorithm that utilized map encoding to allow YOLO to make object-detection predictions on 3D point clouds [22]. Using the z value, intensity, and the density of the point cloud, Red Green and Blue (RGB) values were calculated and assigned to a grid which was assembled into a 2D image for the YOLO network to inference [22,24]. When evaluated against the KITTI dataset, a dataset used for bench-marking object detection in autonomous vehicles, Complex-YOLO achieved 87% AP on the easy car category of the KITTI dataset as well as 51% AP for pedestrians and 73% AP for cyclists [25]. Clustering can be used in point-cloud object detection, where ground points are typically removed, and the remaining points are combined into groups based on the point's coordinates relative to each other [26]. Removing the ground plane can significantly reduce computational requirements as the system will not be required to search that area [22,26]. Overall, processing LiDAR sensor readouts using computer vision algorithms could achieve ground object tracking which could be leveraged to aid in aerial vehicle landing guidance.

The development of a method for detecting and ranging to specific AAM vehicles using LiDAR was explored in this work. Several point-cloud processing steps were required to translate from 3D points to a vehicle's centroid that was used for ranging. Converting 3D points generated from a LiDAR sensor to 2D RGB images can be utilized for vehicle classification [24]. Detecting the centroid of a vehicle in point-cloud data was sought to measure range instead of choosing the closest point to perspectives that can vary wildly. Once recognized, the vehicle's centroid can be estimated using point-cloud measurements to estimate range and altitude from the sensor. Classifying vehicles and matching them to 3D models is an essential step to avoid incorrect or anomalous objects from point clouds. This process can be achieved using a modified YOLO v6 network that operates on 2D RGB data [27]. Several AAM aerial vehicle sizes, from drone to multi-passenger classes, are of interest to be tracked from the ground. The distance from the vertiport, the vehicle size, and the lasers in each LiDAR sensor are predicted to affect the tracking error. The objective of this work was to evaluate the capability of the LiDAR sensors for detecting AAM vehicles from different distances and altitudes. Identification of a relationship between vehicle size, the number of lasers in the LiDAR sensor, and the distance from a vertiport was sought to guide system designers to acceptable levels of range error.

While previous studies [21,22,24,25] have demonstrated LiDAR-based detection methods for autonomous driving and general object recognition, they have not explicitly addressed the needs of AAM operations or the challenges of detecting small aerial vehicles from ground-based sensors. These works often assume dense, near-field LiDAR returns and do not investigate the relationship between point-cloud density and detection accuracy across different aerial vehicle scales. Furthermore, few studies have explored how conventional object-detection frameworks, such as YOLO, can be adapted for LiDAR data encoded into RGB format. This paper addresses these gaps by evaluating the impact of sensor density, vehicle size, and distance on detection performance, and by implementing a novel adaptation of YOLOv6 with SAHI for point-cloud-based aerial vehicle detection. This paper is organized into the following sections of the description of the methods, the results, and a conclusion.

2. Methods

Evaluating the impact of LiDAR sensors for AAM vehicle object detection was achieved by investigating various LiDAR sensors, vehicle classes, and developing an object-detection algorithm. Four LiDAR sensors with different Fields Of View (FOV) and laser counts, along with five vehicle classes ranging from small hobbyist drones to prototype air taxis were selected to provide a wide range of operation conditions. The simulated testing environment represented a vertiport consisting of an asphalt landing pad free of obstructions [13]. Point clouds generated by the LiDAR sensors were processed through a modified RGB YOLO neural network [27] for object classification and position estimation. SAHI was utilized to reduce YOLO training time by overlapping regions of the point cloud [17]. The full system pipeline can be seen in Figure 1. Once the vehicle is detected, using known dimensions and predicted YOLO bound box, the centroid can be estimated. After varying sensor parameters and vehicle position, the detection performance and position error results will then be compared to evaluate the effectiveness of the development method and feasibility of use.

LiDAR sensors examined in this work represented a variety of commercially available systems with differing FOV and number of lasers. The Ouster OS1 and OS0 were selected with both sensors delivering up to 128 vertical and 2048 horizontal lasers, while the OS0 has a FOV of 90 degrees versus the OS1's 45-degree FOV [28]. The Velodyne VLP-32 consists of 32 vertical lasers [28]. In addition to the Ouster and Velodyne sensors, a theoretical LiDAR

featuring double the vertical and horizontal lasers allows for a higher density comparison which further showcased the effect of point-cloud density on system performance [29]. The LiDAR sensors of interest are shown in Table 1 along with their parameters.

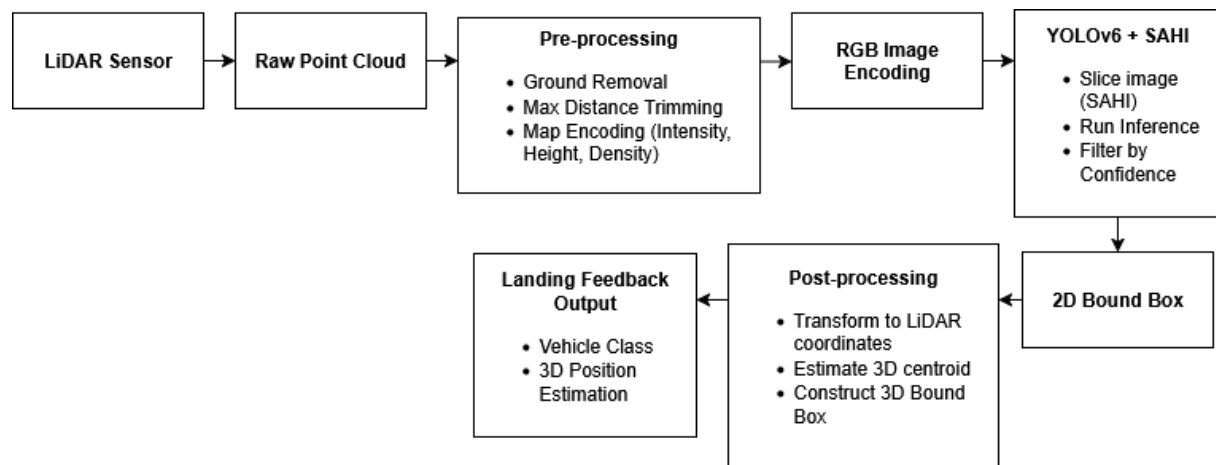










Figure 1. Ground-based LiDAR object-detection pipeline.

Table 1. LiDAR sensor parameters.

	LiDAR Sensor	Horizontal Lasers	Vertical Lasers	Field of View (Degrees)
	VLP-32	1875	32	±15
	Ouster OS0	1024	128	±45
	Ouster OS1	1024	128	±22.5
	Theoretical	2048	256	±33

AAM vehicle classes exist in various different sizes, allowing for sensor comparison based on vehicle size. The five classes examined in this work are a prototype air taxi being developed by Overair, the Beta Alia, the Joby S4, a high payload drone represented by the Aurelia X8 Max, and a hobbyist drone represented by the 3DR Iris+. Vehicle classes range in surface area from 0.249 to 167.53 square meters, platform dimensions and characteristics are shown in Table 2.

Table 2. Aerial vehicle dimensions.

	Aerial Vehicle	Rotors	Length [m]	Width [m]	Height [m]	Surface Area [m ²]	Reference
	3DR Iris+	4	0.457	0.457	0.353	0.249	[30]
	Aurelia X8 Max	8	1.651	1.651	0.750	1.274	[31]
	Joby S4	6	7	11.6	5.5	86.393	[32]
	Beta Alia	4	11	15.24	5	122.988	[33]
	Overair Prototype	4	12	12	6	167.53	[34]

Gazebo, a Robotic Operating System (ROS)-based physics simulation environment, was used to model the LiDAR sensors and vehicles [35]. The developed Gazebo world consisted of a vehicle, LiDAR sensor, and vertiport landing pad, shown in Figure 2. The LiDAR sensor was positioned at the center of the vertiport and had Gaussian noise of 0.01 m to mimic real-world sensor outputs [36]. Using Python 3.10 and ROS Noetic, a listener node was created to record the LiDAR sensor's "/points" topic which broadcasts the x, y, z components of simulated rays and save single-point clouds for further processing. ROS integration also allowed bound boxes to be relayed back to the Gazebo world for plotting overtop the aerial vehicle. An additional Python script was made to pseudo-randomly move the vehicle in the environment and change its orientation which was used to generate the training and test data for the object-detection algorithm. The vehicle was initially moved within ± 40.96 m from the LiDAR sensor in both the X and Y directions of the simulated environment, representing the maximum radial distance supported by Complex-YOLO and used to generate training data for the network [22]. Vehicle height (Z value) was determined based on the sensor's field of view. To capture fine details of small aerial vehicle features, a grid size of 0.04 m was selected, enabling the algorithm to differentiate between trained classes. This grid size, combined with the 40.96-m detection radius, resulted in an image resolution of 2048 by 2048 pixels, referred to as D2048. In the test set, the vehicle was moved between ± 81.92 m in both the X and Y directions of the simulated environment, double the maximum trained detection range and with the max laser range of the LiDAR sensors, from the sensor using SAHI to inference at the native resolution while maintaining the fineness of the grid. Vehicle height used the sensor's FOV to constrain any vehicles from occurring outside the point cloud. Using the extended maximum detection range and the same grid size, the test dataset contained images with a resolution of 4096 by 4096 pixels, called D4096. The vehicle's orientation was changed to provide more diverse data for the algorithm and simulate real-world scenarios that the system could see as vehicles approach for landing. The pitch and roll values ranged between ± 15 degrees from level flight, while the yaw value was between 0 and 360 degrees, which limited any unrealistic flight conditions from occurring in the simulation. Roll, pitch, and yaw ranges were selected from literature [37,38].

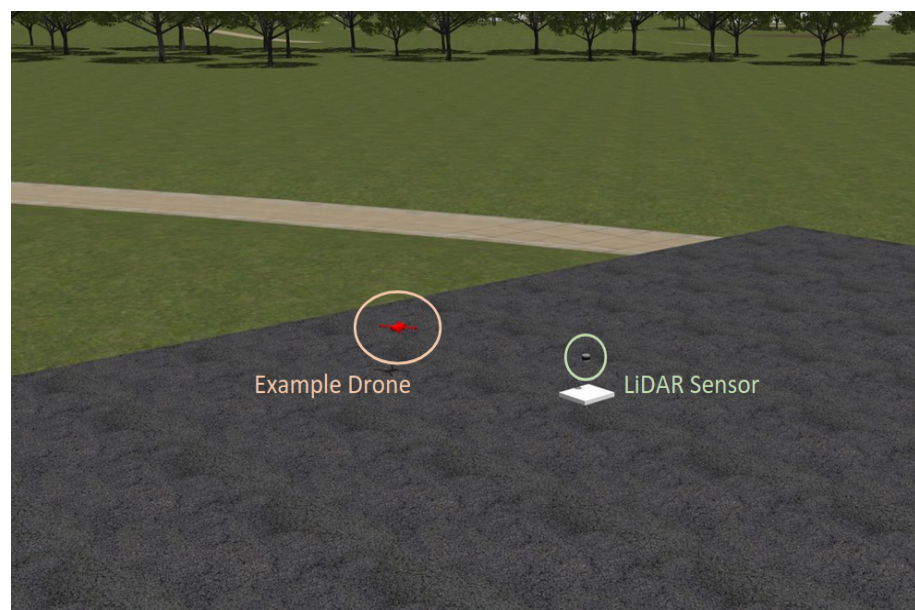


Figure 2. Gazebo environment of vertiport.

YOLO is unable to interpret the 3D space of a point cloud, inputs are expected to be 2D images made up of RGB pixels. Pre-processing allowed point clouds to be mapped to a 2D image for the CNN to inference using SAHI, while post-processing re-mapped the 2D image with predictions back to the LiDAR coordinate frame for use in landing guidance. The system received raw point clouds as inputs and outputs predictions in the form of an estimated class, a 3D object center, and a 3D bounding box in the LiDAR coordinate frame. The complete ground-based LiDAR system is shown in Figure 3. In the following paragraphs, pre-processing, the YOLO model, SAHI, and post-processing will be explained.

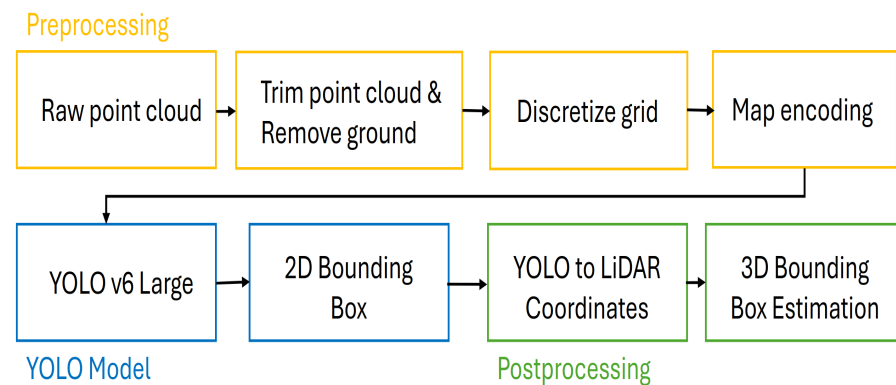


Figure 3. Complete object-detection system diagram.

Pre-processing transformed the raw point cloud into a usable format for the YOLO algorithm to interpret using the previously discussed map encoding techniques of Complex-YOLO, shown in Figure 4. Then, the YOLO model received the encoded image and made classification and position predictions using SAHI, which outputs 2D bounding boxes with an estimated class. Finally, the 2D bounding boxes were transformed back to the LiDAR coordinate frame, and the 3D bounding box is calculated.

The first pre-processing step was trimming the raw point cloud to the system's max detection range, meaning that any points outside max detection range from the sensor were discarded. The ground plane was removed by discarding all points with a Z-value ≤ -0.5 m, which corresponded to the known sensor mounting height in the simulation. This approach assumes a flat terrain aligned with the x-y plane and is similar to filtering methods used in [23] where ground segmentation precedes object detection. While this assumption is suitable for flat vertiports, future implementations may require adaptive ground segmentation for varied terrain. Second, the point cloud was discretized into a grid that will make up the encoded map. This small grid square size allowed for better resolution on the small drone features. Following discretizing the grid, the final step of map encoding occurred. Inspired by [22,24], three RGB channels were encoded using density, intensity, and height. For each grid square, the max height, max intensity, and density were calculated from the points within that grid square, shown in Figure 5. Each feature was then mapped to red, green, or blue, ranging from 0 to 255, giving the grid square a complete RGB color. After this encoding occurred, the point cloud was represented by a 2D image made up of RGB pixels, which the YOLO algorithm used for object detection and position estimation.

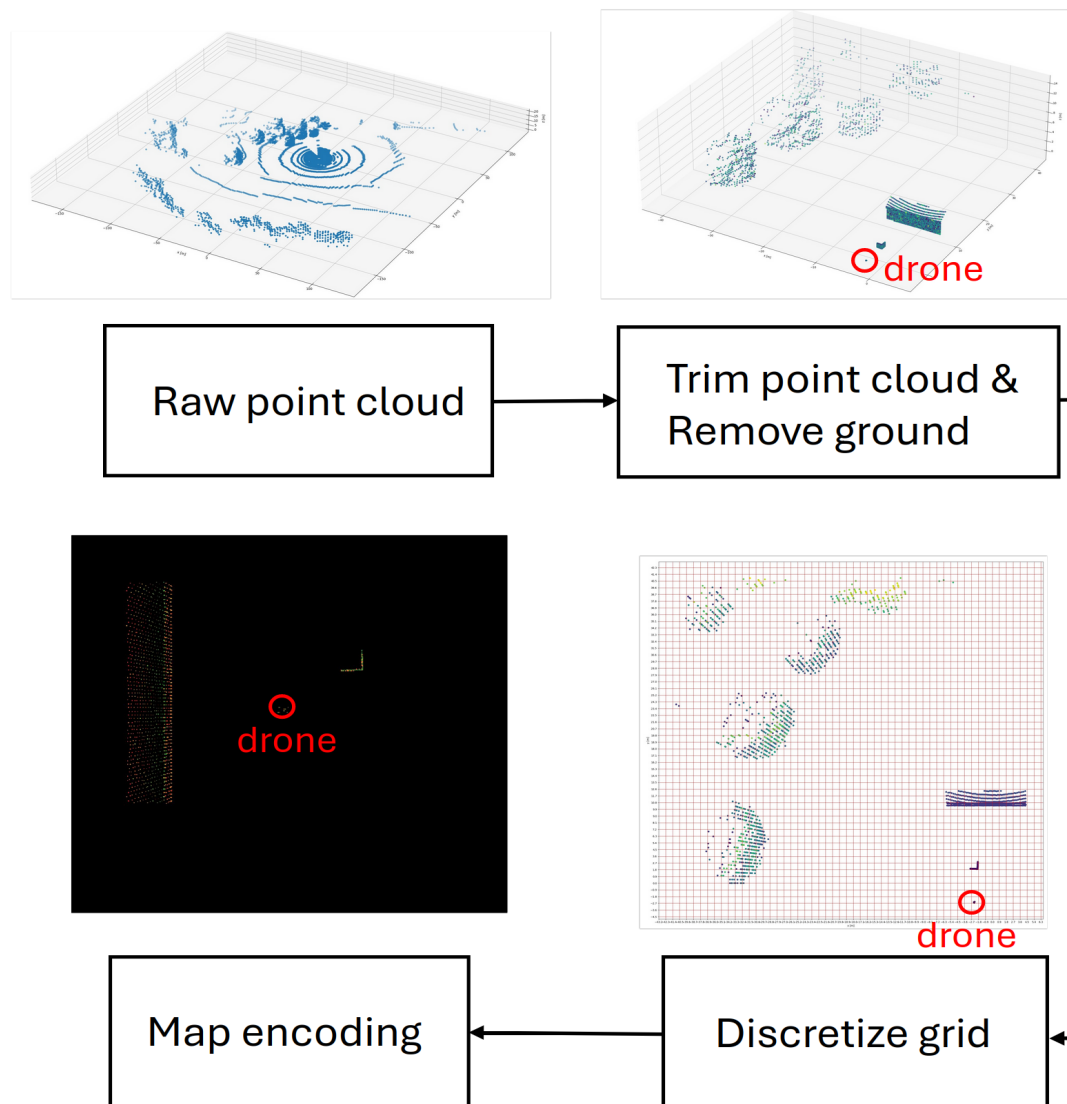


Figure 4. Preprocessing steps.

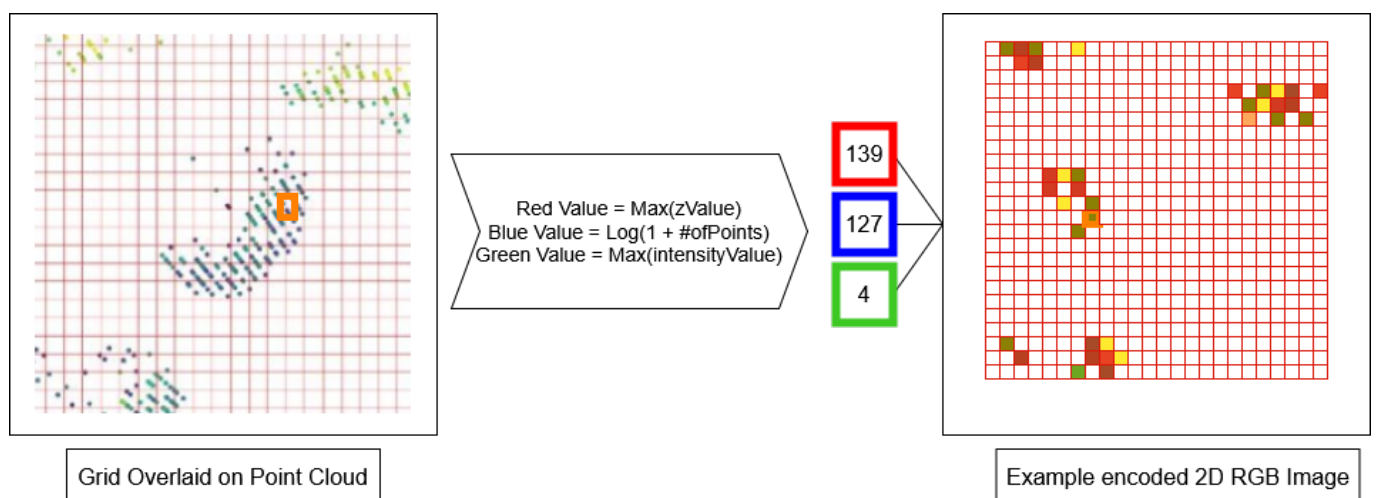


Figure 5. RGB encoding from LiDAR point-cloud features.

YOLO was chosen due to lower latency predictions with higher accuracy on the Common Objects in Context (COCO) dataset than previous YOLO versions [27]. The YOLO

algorithm uses a large dataset to learn each class's features by tuning the model's parameter weights, enabling it to classify and detect specified objects in new environments. SAHI was used in conjunction with YOLO to allow inferencing at the lower D2048 native resolution while maintaining the same small grid square size and extended max distance of D4096 [17]. SAHI takes the preprocessed RGB image from D4096 and slices it into 9 images with the resolution of D2048, shown in Figure 6. Edge case issues were mitigated by overlapping slices, preventing vehicles in between slices from being misclassified [17]. The sliced images were input into the algorithm, providing 2D bounding boxes on the original image with a class label and confidence value. All predictions with less than 0.7 confidence were discarded to limit the number of false positives [39]. The final step of the object-detection system was the post-processing of the 2D bounding box to a 3D bounding box.

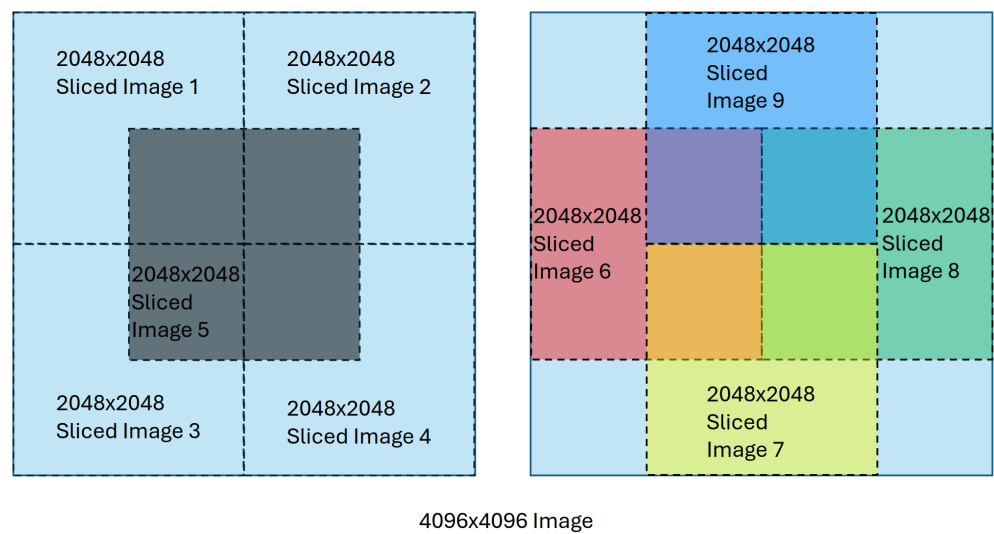


Figure 6. SAHI slicing of a 4096×4096 image into 2048×2048 slices.

Post-processing of point clouds allowed the transformation from 2D image coordinates to 3D LiDAR coordinates. The output bounding box from YOLO had an x, y coordinate system that ranges from 0 to 1, where 0,0 is the top left corner of the image, and 1,1 is the bottom right. Using Equations (1) and (2), where X_Y and Y_Y are the output bounding-box coordinates, the vehicle center was transformed from the YOLO coordinate frame to the LiDAR's frame.

$$X_L = 81.92 \cdot X_Y - 40.96 \quad (1)$$

$$Y_L = -1 \cdot (81.92 \cdot Y_Y - 40.96) \quad (2)$$

After the bounding box center was found, the min and max values that make up the 3D box were found using Equations (3)–(6), where w_Y is the width and l_Y is the length of the 2D bounding box. LiDAR x, y minimum, and maximum values were used to calculate the final height component to find the 3D points that make up the predicted vehicle. Using the max z value of the points that make up the expected vehicle and the known class dimensions, a complete 3D bounding box was extracted which could be used for landing guidance. The max z value prevented objects and other points below the vehicle from being incorporated in the height calculation. An example of post processing is shown in Figure 7.

$$Xmin_L = X_L - 40.96 \cdot w_Y \quad (3)$$

$$Xmax_L = X_L + 40.96 \cdot w_Y \quad (4)$$

$$Ymin_L = Y_L - 40.96 \cdot l_Y \quad (5)$$

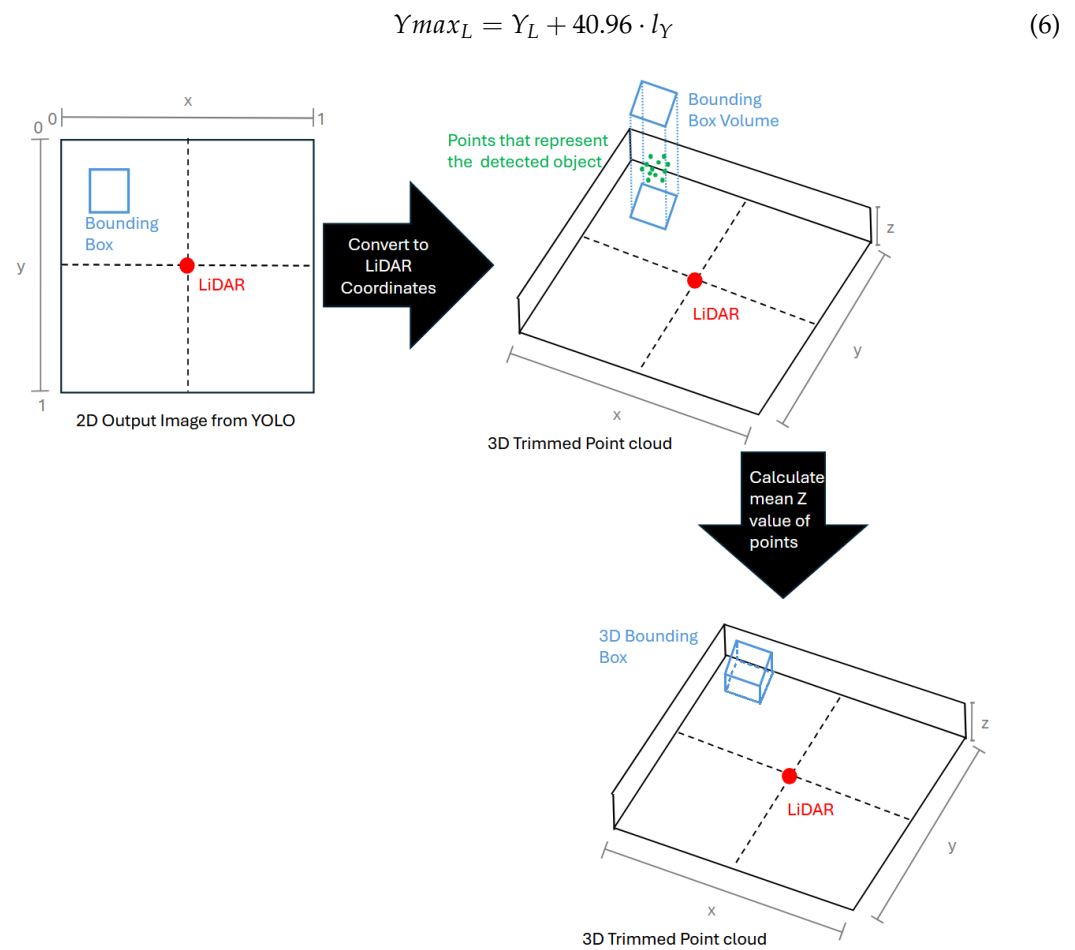


Figure 7. Post-processing of 2D bounding boxes to 3D bounding boxes.

3. Results

Utilizing the described simulation environment and point-cloud processing system in this paper, different LiDAR sensors and AAM vehicle classes were explored to estimate positional detection accuracy. First, the YOLO network needed to be trained. Next, the vehicles were placed 0 to 80 m, which makes up 4098 by 4098 pixels, away from the LiDAR sensor. Finally, the Mean Average Error (MAE) and Mean Average Precision (mAP) were calculated for 3D position estimation and classification accuracy. Additionally, 3D bounding boxes were converted to LiDAR coordinates and plotted using Gazebo to showcase a visual representation of a vehicle, as shown in Figure 8.

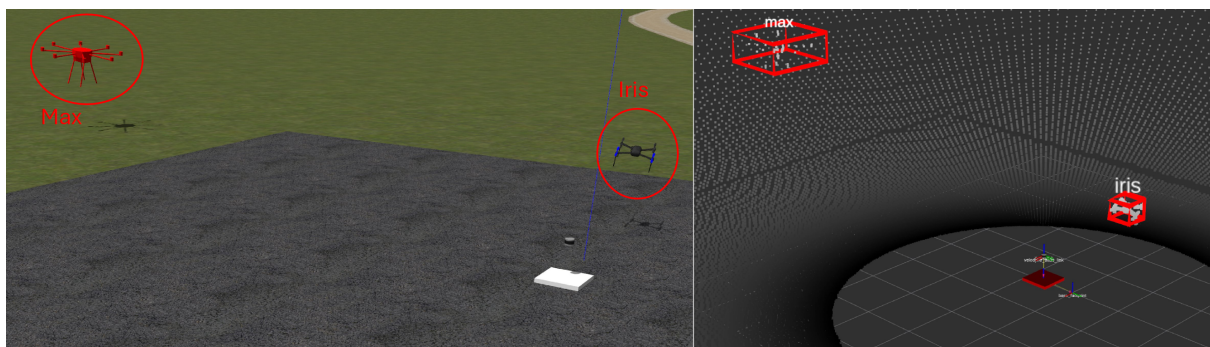


Figure 8. Gazebo environment and corresponding 3D bounding boxes in LiDAR Coordinates.

Training and validation datasets were needed for the YOLO model to estimate aerial vehicle class and position. Point clouds containing the five vehicle classes and LiDAR sensors were recorded and saved using the Gazebo simulation environment to create the training and validation datasets for model training. The YOLO model was then trained at a resolution of D2048 which allowed faster training speed than the test set's resolution of D4096. The quality of training data was an important concern for CNNs as mislabeled data, poor class representation, and non-diverse datasets could negatively affect the model's performance on the test set [40]. Labeling was performed using the known ground truth and physical dimensions of the vehicle, allowing for tightly fitted bounding boxes. All five aerial vehicle classes were equally distributed throughout the training and validation set to eliminate the possibility of over-training on one specific class. Dataset diversity was also improved using YOLO's built-in augmentations that alter training data. Translation, mix-up, and flips were the highest weighted augmentations chosen, which were shown to impact model classification accuracy positively [41]. Note that augmentation only occurred on the training dataset and did not occur during evaluation of the test set.

The LiDAR sensor posed an issue with obtaining useful training data because objects that were occluded or had low point density did not produce enough data points for the model to make accurate predictions. Point clouds that did not have at least 13 points representing the vehicle were removed making the training data sufficient for convergence [42]. The process used to evaluate training and validation point clouds can be summarized in Figure 9, where the vehicle is moved within the environment and the resulting point cloud is checked to see if at least 13 point exist at the vehicle's ground truth location.

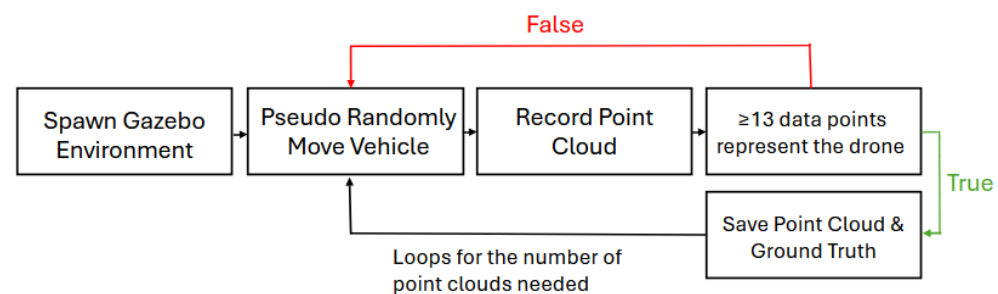


Figure 9. Training and validation point-cloud process.

Training and validation results are shown in Figure 10 that indicate convergence of the model and the completion of training. The training dataset consisted of 7500 point clouds, each class represented by 1500 point clouds. The model training was completed using an NVIDIA GeForce RTX 4080 over 120 epochs with a batch size of 2. Additional training beyond 120 epochs yielded no improvement on model performance. Model training lasted 33 h, with the best epoch occurring at number 78, resulting in 96.2% Mean Average Precision (mAP) for all classes at an Intersection over Union (IoU) threshold of 0.95, shown in Figure 10. The validation set was evaluated every 20 epochs for the first 60 epochs, after which evaluation occurred every five epochs. Training the YOLOv6 model required 33 h on an RTX 4080 and is a one-time offline operation.

Following the completion of training, the epoch with the highest mAP was used to evaluate the D2048 resolution test set resulting in the confusion matrix shown in Table 3. Vehicles were correctly classified in 37,757 of 40,000 total test point clouds, resulting in a mAP of 94.3% with the incorrect instances being classified as similar vehicle types.

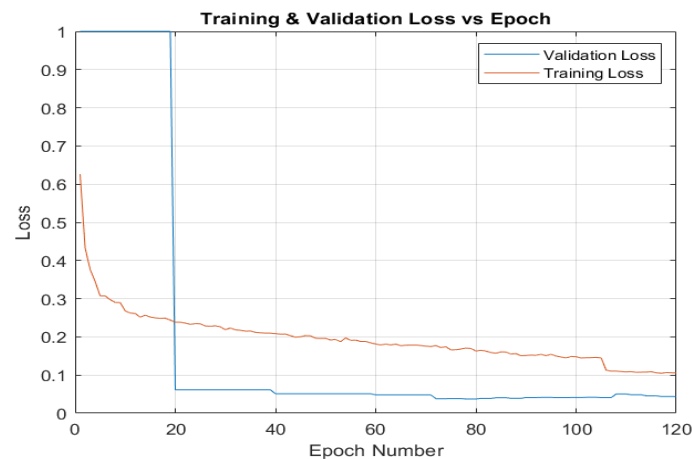


Figure 10. Training and validation loss across epochs.

Table 3. Test-dataset confusion matrix.

		Predicted					
		Iris	Max	S4	Alia	Overair	No Prediction
Actual	Iris	5957	0	0	0	0	2043
	Max	1	7967	0	0	0	32
	S4	0	0	7899	0	10	91
	Alia	0	0	0	7956	2	42
	Overair	0	0	4	0	7978	18

Four sets of point clouds existed in the test set, each representing one of the four LiDAR sensors investigated in this work. Within each LiDAR set, 2000 point clouds of each vehicle class with varying distances from the sensor, with a resolution of D4096. The test set was used to evaluate the system's accuracy on new data as the training and validation datasets have been seen by the system before which could skew performance. In addition to the point clouds, ground truth labels were included to calculate position estimation and classification errors. Evaluation of the dataset used mAP and MAE as the primary metrics. Classification was evaluated using mAP which allowed for an even balance between precision and recall. The Intersection over Union (IoU), or intersection of predicted and ground truth bounding boxes, thresholds between 0.5 to 0.95 were used to calculate mAP. The higher the IoU threshold, the more similar the predicted bounding box must be to count as a positive classification [43]. The MAE from the expected center to the ground truth center was used as the performance metric for vehicle position calculated using:

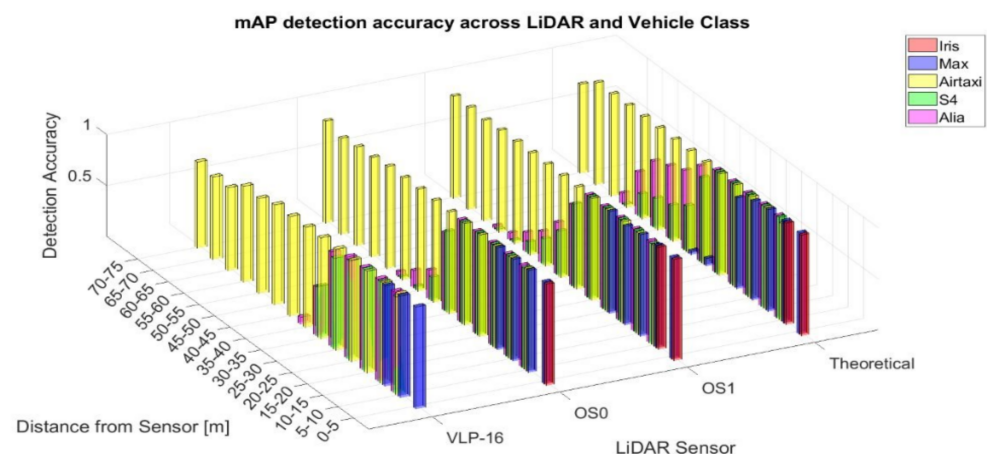
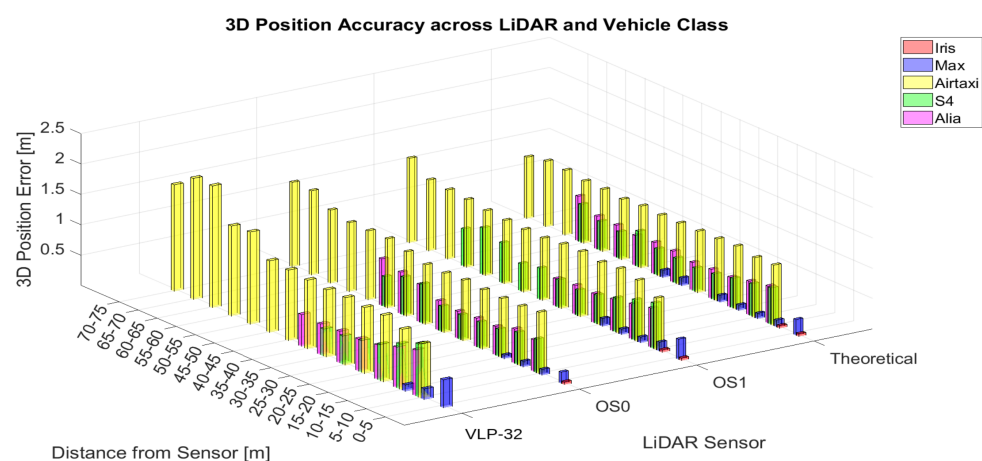
$$AE = \sqrt{(x_L - x_T)^2 + (y_L - y_T)^2 + (z_L - z_T)^2} \quad (7)$$

where the center of the ground-truth 3D bounding box is (x_T, y_T, z_T) and the model's predicted 3D bounding box center is (x_L, y_L, z_L) . For each LiDAR sensor, these metrics were compared to examine the effects each sensor has on object detection and position estimation. When tested at the model's native resolution, D2048, the system achieved 95% mAP across all vehicle classes and sensors except for the Iris drone when sampled by the VLP-32 LiDAR, shown in Table 4. This was due to the lower point-cloud density of the VLP-32 combined with the small drone size, which resulted in suboptimal point coverage.

Table 4. mAP of test dataset for each vehicle class and LiDAR.

		Aerial Vehicle Class				
		Iris	Max	S4	Alia	Overair
LiDAR Sensor	Theoretical	0.991	0.991	1.000	1.000	0.997
	OS1	0.993	0.996	0.999	1.000	0.997
	OS0	0.995	1.000	0.998	1.000	0.999
	VLP-32	N/A	0.998	0.954	0.979	0.997

The D4096 set of point clouds displayed how point-cloud density affects detection accuracy and positional error based on distance from the LiDAR sensor. Higher point-cloud density led to higher accuracy as the distance from the sensor increased, as shown in Figure 11. Larger vehicles allowed for continued detection further from the sensor due to more surface area for points to populate. An inverse relationship was seen when looking at the 3D positional error, shown in Figure 12, where higher resolution LiDARs had lower errors due to more points representing the vehicle, which allowed for more accurate positioning. Across all LiDAR sensors, larger aerial vehicles like the Overair and Beta Alia prototypes consistently exhibited the highest classification accuracy and lowest positional error. These vehicles produced denser point clouds due to their greater surface area, improving both object visibility and 3D position accuracy. For example, the Overair prototype maintained over 99% classification accuracy across all sensor types, while its average positional error remained below 0.3 m, even at extended distances.

**Figure 11.** Detection accuracy based on LiDAR and vehicle class.**Figure 12.** Three-dimensional position error based on LiDAR and vehicle class.

Point-cloud density plays a critical role in detection accuracy as vehicle distance increases. Higher-density LiDARs, such as the theoretical sensor and the Ouster OS0, maintained higher classification rates at longer distances, while lower-density sensors like the VLP-32 experienced a steep decline. Similarly, an inverse relationship between point-cloud density and positional error exists. High-density LiDARs maintained sub-meter accuracy well beyond 40 m, while low-density sensors exceeded the 0.5 m threshold even at moderate ranges, especially for smaller aerial platforms.

The developed ground-based LiDAR system was able to detect selected classes of aerial vehicles and predict their average 3D position within 0.5 m of the ground truth. When compared to ILS' category system for precision approach and landing, the LiDAR detection system meets accuracy requirements for all categories up to IIIb [5,44,45]. Additionally, point-cloud density and vehicle size were identified as the most significant factors contributing to the effectiveness of the ground-based LiDAR system in detecting vehicles and estimating their positions in 3D space. The positional accuracy achieved in this study exceeds that of ADS-B, RADAR, and MLS, as indicated in Table 5. However, the detection range is less than all other options except the vision-based ArUco image landing system [19]. The LiDAR system is constrained to the max laser distance of the sensor itself rather than reduced point coverage when the vehicle is further from the vertiport. Given these findings, ground-based LiDAR using the developed estimation system is well-positioned to assist with AAM landings in environments with limited GNSS availability, while complementary long-range guidance systems can support earlier phases of flight.

Table 5. Related work average positional error.

Method	Effective Range [m]	Average Positional Error [m]	Reference
RADAR	16,000	6.5	[11]
ADS-B	N/A	3–5	[3]
MLS	18,520	2.4–3	[46]
ILS	18,520	Varied based on Category level	[5]
Camera + ArUco	30 (vertical)	<0.05	[19]
GNSS/IMU Fusion	N/A	<0.1	[47]
LiDAR	75	0.5	Current Work

4. Conclusions

Ground-based sensing solutions for AAM vehicles may play a critical role in enabling safe autonomous operations. This work demonstrated the feasibility of using ground-based LiDAR for aerial vehicle classification and 3D position estimation during landing. The combination of YOLO v6 RGB LiDAR data using SAHI to reduce computation allows for a novel vehicle-position detection process to be used at close ranges to veriports. Through simulation of multiple sensor configurations and vehicle classes, a modified YOLO model with SAHI was able to achieve sub-meter positional accuracy (0.5 m) and over 95% classification accuracy across most scenarios. Higher-density sensors yielded the best performance, especially for large vehicles. These results show that ground-based LiDAR can serve as a complementary close-range landing system given the ability to wirelessly communicate position data to a vehicle, particularly during final approach, where precise positioning is essential. The system provides a foundation for additional studies that could explore LiDAR-based detection during adverse weather conditions, newer and higher-density LiDAR sensors, and a wider set of vehicle types.

Author Contributions: Conceptualization, J.K. and J.W.; Methodology, J.K. and J.W.; Software, J.K.; Validation, J.K.; Investigation, J.K. and J.W.; Writing – original draft, J.K.; Writing – review & editing, J.K. and J.W.; Supervision, J.W.; Project administration, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Air Traffic By The Numbers | Federal Aviation Administration. 2024. Available online: https://faa.gov/air_traffic/by_the_numbers (accessed on 13 August 2024).
2. Jeon, D.; Eun, Y.; Kim, H. Estimation fusion with radar and ADS-B for air traffic surveillance. *Int. J. Control Autom. Syst.* **2015**, *13*, 336–345. [CrossRef]
3. Renfro, B.A.; Stein, M.; Boeker, N.; Terry, A. An Analysis of Global Positioning System (GPS) Standard Positioning Service (SPS) Performance for 2017. 2018. Available online: <https://www.gps.gov/systems/gps/performance/2014-GPS-SPS-performance-analysis.pdf> (accessed on 25 January 2025).
4. Evans, T.E. Microwave Landing System. *IEEE Aerosp. Electron. Syst. Mag.* **1986**, *1*, 6–9. [CrossRef]
5. Eltahier, M.M.A.; Hamid, K. Review of instrument landing system. *IOSR J. Electron. Commun. Eng.* **2017**, *12*, 106–113. [CrossRef]
6. Federal Aviation Administration. *Urban Air Mobility (UAM) Concept of Operations 2.0*; Technical Report; Federal Aviation Administration: Washington, DC, USA, 2023. Available online: https://www.faa.gov/sites/faa.gov/files/Urban%20Air%20Mobility%20%28UAM%29%20Concept%20of%20Operations%202.0_0.pdf (accessed on 14 February 2025).
7. Ono, F.; Ochiai, H.; Miura, R. A wireless relay network based on unmanned aircraft system with rate optimization. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 7699–7708. [CrossRef]
8. NASA Autonomous Flight Software Successfully Used in Air Taxi Stand-Ins—NASA. 2024. Section: Armstrong Flight Research Center. Available online: <https://www.nasa.gov/centers-and-facilities/armstrong/nasa-autonomous-flight-software-successfully-used-in-air-taxi-stand-ins/> (accessed on 15 July 2024).
9. Musa, S.A.; Rashid, E.A.; Ibrahim, I.P.; Salah, A.A. A Review of Copter Drone Detection Using a Radar System. *Def. S&T Tech. Bull.* **2019**, *12*, 1985–6571.
10. Konno, H. Design of an Aircraft Landing System Using Dual-Frequency GNSS. Ph.D. Thesis, Stanford University, Stanford, CA, USA, 2008.
11. Naus, K.; Wąż, M.; Szymak, P.; Gucma, L.; Gucma, M. Assessment of ship position estimation accuracy based on radar navigation mark echoes identified in an Electronic Navigational Chart. *Measurement* **2021**, *169*, 108630. [CrossRef]
12. Quevedo, A.D.; Urzaiz, F.I.; Menoyo, J.G.; Lopez, A.A. Drone detection and radar-cross-section measurements by RAD-DAR. *IET Radar Sonar Navig.* **2019**, *13*, 1437–1447. [CrossRef]
13. Michael A.P. Meyers, P. Engineering Brief No. 105, Vertiport Design. 2023. Available online: https://faa.gov/airports/engineering/engineering_briefs/eb_105a_vertiports (accessed on 9 May 2024).
14. Xie, G. *Optimal On-Airport Monitoring of the Integrity of GPS-Based Landing Systems*; Stanford University: Stanford, CA, USA, 2004.
15. Hu, S.; Goldman, G.; Borel-Donohue, C. Detection of unmanned aerial vehicles using a visible camera system. *Appl. Opt.* **2017**, *56*, 214–221. [CrossRef] [PubMed]
16. Nabati, R.; Qi, H. Centerfusion: Center-based radar and camera fusion for 3d object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2021; pp. 1527–1536.
17. Akyon, F.C.; Altinuc, S.O.; Temizel, A. Slicing Aided Hyper Inference and Fine-tuning for Small Object Detection. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 966–970. [CrossRef]
18. Sani, M.F.; Karimian, G. Automatic navigation and landing of an indoor AR. drone quadrotor using ArUco marker and inertial sensors. In Proceedings of the 2017 International Conference on Computer and Drone Applications (IConDA), Kuching, Malaysia, 9–11 November 2017; pp. 102–107.
19. Khazetdinov, A.; Zakiev, A.; Tsoy, T.; Svinin, M.; Magid, E. Embedded ArUco: A novel approach for high precision UAV landing. In Proceedings of the 2021 International Siberian Conference on Control and Communications (SIBCON), Kazan, Russia, 13–15 May 2021; pp. 1–6.

20. Wang, X.; Pan, H.; Guo, K.; Yang, X.; Luo, S. The evolution of LiDAR and its application in high precision measurement. *IOP Conf. Ser.* **2020**, *502*, 012008. [CrossRef]
21. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499. [CrossRef]
22. Simon, M.; Milz, S.; Amende, K.; Gross, H.M. Complex-YOLO: Real-time 3D Object Detection on Point Clouds. *arXiv* **2018**, arXiv:1803.06199.
23. Chen, X.; Li, S.; Mersch, B.; Wiesmann, L.; Gall, J.; Behley, J.; Stachniss, C. Moving Object Segmentation in 3D LiDAR Data: A Learning-Based Approach Exploiting Sequential Data. *IEEE Robot. Autom. Lett.* **2021**, *6*, 6529–6536. [CrossRef]
24. Qi, C.R.; Liu, W.; Wu, C.; Su, H.; Guibas, L.J. Frustum pointnets for 3d object detection from rgb-d data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 918–927.
25. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012.
26. El Yabroudi, M.; Awedat, K.; Chabaan, R.C.; Abudayyeh, O.; Abdel-Qader, I. Adaptive DBSCAN LiDAR point cloud clustering for autonomous driving applications. In Proceedings of the 2022 IEEE International Conference on Electro Information Technology (eIT), Mankato, MN, USA, 19–21 May 2022; pp. 221–224.
27. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976. [CrossRef]
28. Lidar Sensors for High-Resolution, Long Range Use in Autonomous Vehicles, Robotics, Mapping. Low-Cost & Reliable for Any Use Case. 2024. Available online: <https://ouster.com/> (accessed on 12 September 2024).
29. Griesbacher, C.; Fruhwirth-Reisinger, C. An Investigation of Beam Density on LiDAR Object Detection Performance. *arXiv* **2025**, arXiv:2503.15087.
30. 3DR Iris+—Autonomous Multicopter. 2024. Available online: <https://www.adafruit.com/product/2199>, (accessed on 20 October 2024).
31. Aurelia X6 Max. 2024. Available online: <https://aurelia-aerospace.com/pages/aurelia-x8-series> (accessed on 20 October 2024).
32. Joby Aviation | Joby. 2024. Available online: <https://www.jobyaviation.com/> (accessed on 20 October 2024).
33. Aircraft | VTOL and CTOL. 2024. Available online: <https://www.beta.team/aircraft/> (accessed on 20 October 2024).
34. Overair. 2024. Available online: <https://www.aviationtoday.com/2022/06/15/overair-just-received-145-million-funding-evtol-development/> (accessed on 20 October 2024).
35. The Robot Operating System (ROS). 2024. Available online: <https://www.ros.org/> (accessed on 12 September 2024).
36. Ouster, Inc. *OS1 Sensor Datasheet (Rev 7, v3.1)*; Technical Report; Ouster, Inc.: San Francisco, CA, USA, 2023. Available online: <https://data.ouster.io/downloads/datasheets/datasheet-rev7-v3p1-os1.pdf> (accessed on 7 March 2025).
37. Montalvo, C.; Costello, M. Meta aircraft flight dynamics. *J. Aircr.* **2015**, *52*, 107–115. [CrossRef]
38. Kawamura, E.; Kannan, K.; Lombaerts, T.; Ippolito, C.A. Vision-based precision approach and landing for advanced air mobility. In Proceedings of the AIAA SciTech 2022 Forum, San Diego, CA, USA, 3–7 January 2022; p. 0497.
39. Ali Mazen, F.M.; Shaker, Y. Small Object Detection in Complex Images: Evaluation of Faster R-CNN and Slicing Aided Hyper Inference. *Int. J. Adv. Comput. Sci. Appl.* **2025**, *16*, 951–960. [CrossRef]
40. Jain, A.; Patel, H.; Nagalapatti, L.; Gupta, N.; Mehta, S.; Guttula, S.; Mujumdar, S.; Afzal, S.; Sharma Mittal, R.; Munigala, V. Overview and importance of data quality for machine learning tasks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Online, 23–27 August 2020; pp. 3561–3562.
41. Chung, Q.M.; Le, T.D.; Dang, T.V.; Vo, N.D.; Nguyen, T.V.; Nguyen, K. Data augmentation analysis in vehicle detection from aerial videos. In Proceedings of the 2020 RIVF International Conference on Computing and Communication Technologies (RIVF), HoChiMinh City, Vietnam, 5–7 April 2020; pp. 1–3.
42. Darrah, M.; Richardson, M.; DeRoos, B.; Wathen, M. Optimal LiDAR Data Resolution Analysis for Object Classification. *Sensors* **2022**, *22*, 5152. [CrossRef] [PubMed]
43. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
44. RTCA Special Committee 159. *Minimum Operational Performance Standards for GPS Local Area Augmentation System Airborne Equipment*; Technical Report DO-253D; RTCA, Inc.: Washington, DC, USA, 2017.
45. Federal Aviation Administration. *Criteria for Approval of Category III Weather Minima for Takeoff, Landing, and Rollout Operations*; Technical Report AC 120-28D; U.S. Department of Transportation, Federal Aviation Administration: Washington, DC, USA, 1999. Available online: https://www.faa.gov/documentlibrary/media/advisory_circular/ac120-28d.pdf (accessed on 15 May 2025).

-
46. Cicolani, L.S. *Position Determination Accuracy from the Microwave Landing System*; Technical Report; NASA: Washington, DC, USA, 1973.
 47. Rieke, M.; Foerster, T.; Geipel, J.; Prinz, T. High-precision positioning and real-time data processing of UAV-systems. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *38*, 119–124. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.