

Multi-Sensor Fusion and YOLOv5 Model for Automated Detection of Aircraft Cabin Door

Ihnsik Weon¹, Soon-Geul Lee²

Airport Industrial Technology Institute, Incheon International Airport Corporation, Incheon, South Korea 22382¹
Dept. of Mechanical Engineering, Kyunghee University, Yongin, South Korea 17104²

Abstract—This study investigated perception technology of an autonomous driving system to enable independent connection between an aircraft and a boarding bridge. GigE video sensors and solid-state lidars were installed on the cabin side of the boarding bridge, and a technology that fuses the data from these two different sensors was developed and applied. Using the fused data, a technology for identifying the aircraft door was researched using Yolo-v5, one of the feature point extractors. Yolo-v5 is a deep learning-based feature point extractor that was able to identify the door after being trained with more than 10,000 frames of images under predetermined weather and time conditions. Additionally, a parallel alignment control function was applied between the aircraft body and the cabin of the boarding bridge to increase the reliability of the aircraft door identification technology based on the fused data. To achieve this, a certain area of interest was set within the fused data so that the distance deviation to the left and right of the cabin could be calculated. Finally, to verify the research results, tests were conducted to identify aircraft doors under various environmental conditions with more than six airlines selected. Originally, the Yolo-v5 model secured 93.5% accuracy, but through this study, the detection accuracy for limited-environment aircraft doors was increased to over 95%.

Keywords—Jet bridge; Yolo-v5; sensor fusing; segmentation; door detects; automation docking system

I. INTRODUCTION

The passenger boarding bridge (PBB) is an important airport facility that connects the airport terminal and aircraft through a hub and spoke system [1]. As, the airport's aircraft capacity increases, the need for aerobridges also increases. Therefore, it is necessary to maintain a consistent schedule for the contact time, which can vary greatly depending on the skill level of the workers. Safety is also important [2].

Accordingly, there is a need for autonomous driving technology for aerobridges in airport operations. Recently, Shinmaywa Industries and Panasonic in Japan conducted joint research and development of autonomous driving systems for aerobridges [3]. They completed the development of a system that can achieve contact with aircraft within several tens of centimeters from the aircraft body. However, the system is only applicable to C and D class aircraft, and its reliability can be greatly affected by weather and lighting conditions as it relies on image sensors for aircraft target recognition technology [4].

In addition, Airport Equipment in Australia has developed an autonomous driving system for aerobridges called

Intellidock [5], which applies deep learning-based aircraft door recognition technology. It is currently being tested at Wellington Airport. However, this technology is only applicable to aerobridge contact with short-haul C and D class aircraft, and it relies heavily on video data. Therefore, it can be greatly affected by weather and lighting conditions during the precise contact phase of the aircraft [6, 7].

Therefore, in this study, we utilized multi-sensor fusion technology to develop a technology that can calculate precise aircraft contact points with minimal weather and lighting interference. Multi-sensor fusion technology is a widely used technology in autonomous driving cars and robots that compensates for the inherent disadvantages of each sensor without any constraints, making it the most critical recognition technology for both indoor and outdoor environments.

Sensor fusion technology plays a significant role in improving the accuracy and precision of object recognition by integrating data collected from multiple sensors. However, processing such multi-dimensional data requires a significant amount of computing resources. Therefore, in this study, we used CUDA coding and designed optimal computer resources for parallel processing to address problems [8] such as low FPS (Frames per Second) and flow phenomenon. This approach optimizes the performance of sensor fusion technology, enhancing the quality of object recognition. [9]

The sensor fusion data was used to apply the Yolo-v5 technique [10] for detecting aircraft doors. To apply the Yolo-v5 technique, more than 10,000 images were extracted and used for training by specifying four-season weather, specific times of day (day and night), and different aircraft designs and colors for various airlines. To consider the different body colors and designs of each airline, representative Korean airlines and six foreign airlines were selected to create an image dataset.

The identified location of aircraft doors through this recognition process was used to control the horizontal position between the end-effector (EF) at the boarding bridge and the aircraft body for smooth aircraft door recognition. The sensor fusion data was used as input data for the horizontal control system by setting the aircraft body as a region of interest in the sensor fusion data. Fig. 1 and Fig. 2 show the automation passenger boarding bridge docking system in Australia and Japan respectively.

Finally, based on the learned data, three representative scenarios were determined according to weather conditions and

tested. The tests were conducted even under conditions where crosswinds could affect the physical sensor system.



Fig. 1. Automation passenger boarding bridge docking system: Australia airport equipment.



Fig. 2. Automation passenger boarding bridge docking system (Paxway): Japan shinmaywa industrial.

II. OPERATION SEQUENCE JETBRIDGE SYSTEM

Airport boarding bridges are an essential airport facility for passenger boarding and disembarkation, connecting aircraft and airports. Aircraft models vary depending on their flight distance and destination, and consequently, there are many types of boarding bridges to accommodate them. Boarding bridges are classified into fixed and movable types, with movable ones allowing up to three boarding bridges to be installed at one gate, depending on the number of aircraft models they need to handle [11].

Boarding bridges are named P1 to P3 in the order of their proximity to the airport terminal building, and the sequence of approach starts with the outermost P3 boarding bridge and ends with P1 as the last one [12]. Conversely, the sequence of departure follows the reverse order, with P1 as the starting point and P3 as the last one to be departed.

Furthermore, aircraft grades are divided into six categories, mainly based on the size of the aircraft's main wing span and the outer wheel width of the aircraft's main landing gear, in accordance Table I with the regulations of the International Civil Aviation Organization (ICAO). Grade A is for small planes and Grade B is for planes with fewer than 50 passenger seats, which are not used in international airports. The aircraft grades commonly used for domestic and international flights are C and above, with the B737 and A320 models being the most representative.

Therefore, at airports, boarding bridges are generally not differentiated according to passenger seat class for C and D grade aircraft, so one P1 boarding bridge is used, [14] and passengers embark and disembark mostly through the door closest to the cockpit. However, E grade aircraft, such as the B777, [15] differentiate services for premium seats such as business class and carry out boarding and disembarkation not only through the door closest to the cockpit but also through the door on the main wing side, as their fuselage length is longer. In such cases, both P1 and P2 boarding bridges are used. Finally, F grade models such as the A380 and B747-8i, which have a multi-story seating structure, use two P1 boarding bridges on the first floor and one on the second floor to board and disembark passengers (see Fig. 3).

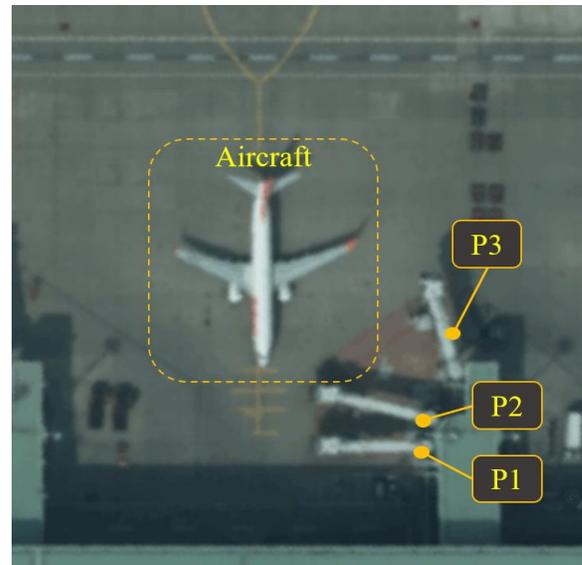


Fig. 3. Airport apron environment and passenger boarding bridge placement status.

TABLE I. AERODROME REFERENCE CODE [13]

CODE ELEMENT 1		CODE ELEMENT 2		
CODE NUMBER	AEROPLANE REFERENCE FIELD LENGTH	CODE LETTER	WING SPAN	OUTER MAIN GEAR WHEEL SPAN*
1	Less than 800m	A	Up to but not including 15m	Up to but not including 4.5m
2	800m up to but not including 1200m	B	15m up to but not including 24m	4.5m up to but not including 6m
3	1200m up to but not including 1800m	C	24m up to but not including 36m	6m up to but not including 9m
4	1800m and over	D	36m up to but not including 52m	9m up to but not including 14m
* Distance between the outside edges of the main gear wheels.		E	52m up to but not including 65m	9m up to but not including 14m
		F	65m up to but not including 80m	14m up to but not including 16m

III. SYSTEM CONFIGURATION

In order to accurately recognize and calculate the precise position of aircraft doors, basic information on the posture of boarding bridges needs to be measured accurately. According to Fig. 4, the sensors installed on the boarding bridges only included height measurement sensors, rotation angle measurement sensors, cabin angle measurement sensors, and wheel carriage angle measurement sensors. The data collected from these sensors is unreliable and does not support high-resolution data required for integration with automatic operating systems as low-resolution support sensors.

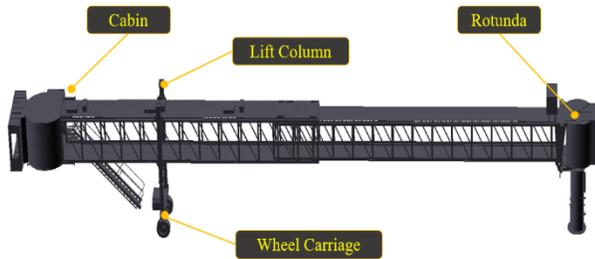


Fig. 4. Configuration of passenger boarding bridge.

A. Schematic Diagram

To conduct research on aircraft door recognition technology, multiple environmental sensing sensors were utilized, specifically two types of 3D LIDAR sensors and a GigE video sensor capable of supporting gigabit data transfer rates and 4K resolution. Additionally, a MEMS-type GPS/INS sensor was installed to estimate the relative position of the boarding bridge. As shown in Fig. 5, the 3D LIDAR sensors collected data through TCP/IP and UDP interfaces using the RJ42 type, while the GigE video sensor outputted images with a maximum resolution of 3,840 * 2,160. Because large amounts of data must be processed in real-time, the two types of 3D LIDAR sensors were connected to a single switch hub, while the GigE video sensor was directly connected using a separate frame grabber. The GPS/INS sensor used a USB 3.0 interface to ensure a maximum data refresh rate of 10Hz.

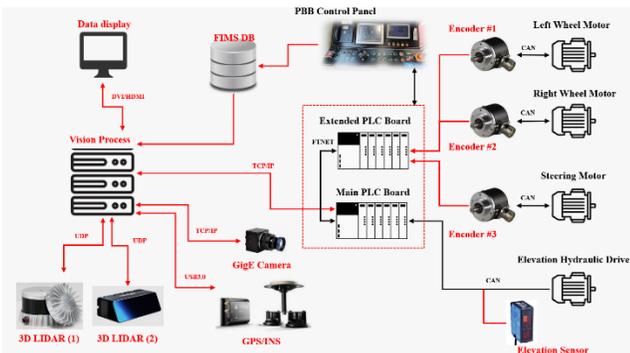


Fig. 5. Schematics of passenger boarding bridge for automations.

Basic boarding bridge posture information, including joint and end effector data information based on the operation of the boarding bridge, is collected through the PLC central controller and the CAN and FTNET interfaces. The PLC central

controller and the aircraft door recognition processor collect and refresh data every moment through TCP communication.

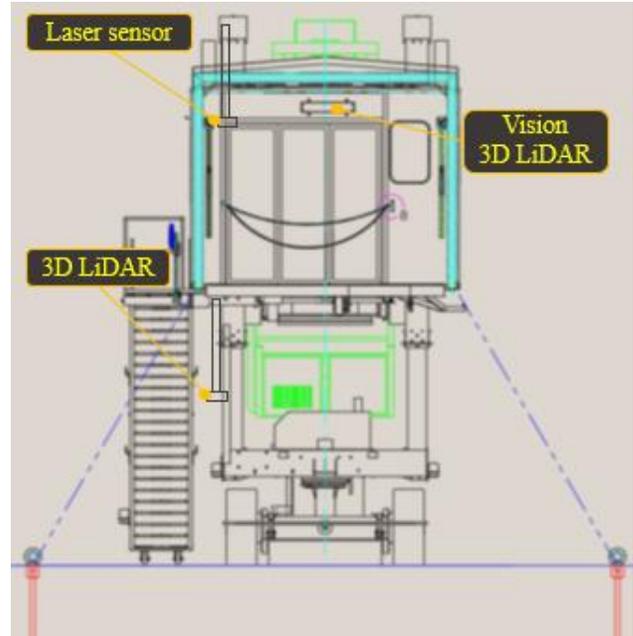


Fig. 6. Sensor installation location diagram.

B. System Layout

The structure of aircraft aprons varies from airport to airport, which also affects the layout of boarding bridges. Therefore, there is a significant difference in the accuracy and precision of recognition results depending on the arrangement of environmental sensing sensors. Additionally, as shown in Fig. 4, boarding bridges are composed of multi-joint structures with at least 4 degrees of freedom, so target point recognition for kinematics control is crucial. Due to the environmental characteristics of Incheon International Airport, strong winds frequently occur, maintaining an average wind speed of over 9 m/s. As a result, significant vibrations occur in the cabin, which is a structure vulnerable to vibrations.

Vibrations can affect image sensor and LIDAR sensor data, leading to significant impacts on object recognition results. Therefore, as shown in Fig. 6, a 3D LIDAR and image sensor were installed in a single housing, located as close as possible to the wheel carriage, the vibration source, and near the center of rotation of the EF cabin to minimize coordinate system unification and external interference.

IV. DATA ANALYSIS

To process multiple environmental sensing sensor data simultaneously, we first examined the configuration and update cycle of each sensor data. For this study, the boarding bridge is equipped with three types of sensors: Solid State 3D LiDAR, Rotary 3D LiDAR, and GigE cameras. The posture information of the boarding bridge is received through a PLC processor for coordinate system alignment of each data. The posture information of the boarding bridge can be collected as shown in Fig. 4, and it is transmitted with a refresh rate of 50ms, taking into account the data processing speed between the PLC processor for angle sensor data received from each

drive position and the integrated controller for automatic operation.

For environmental perception sensors, we used the FLIR Blackfly S GigE camera. Solid State LiDAR installed in the cabin area is Velodyne M1600 model, and the Rotary LiDAR installed in the lower part of the boarding bridge is Ouster OS1-64. The update cycle and size of each sensor's data were confirmed. The GigE camera supports a resolution of 2,448 * 2,048, with a sensing area of 8.45 * 7.07 (mm), and it updates the image at a speed of approximately 25~27 frames per second (FPS) based on the maximum supported resolution.

Velodyne M1600 LiDAR sensor can collect data up to a maximum distance of 30m, and the size of the data generated depends on the data update speed. In this study, it supports 160 data lines with a refresh rate of 100ms and a vertical resolution of 0.2 degrees. Furthermore, it is optimized for identifying nearby objects with a maximum distance error of 4mm.

As shown in Fig. 7, the environmental conditions are standardized, and the object material conditions that determine data collection errors are uniform, resulting in consistent data collection under most conditions. However, since all three types of data need to be collected simultaneously, there was an issue of Lidar data loss occurring during processing when the network occupancy exceeded 300 Mbps.

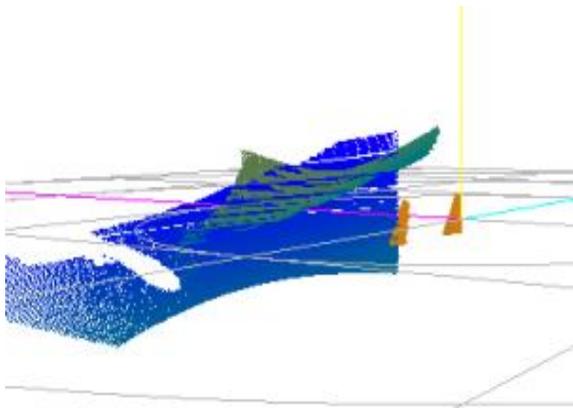


Fig. 7. Velodyne M1600 sample 3D data in real environment.

To address this fundamental problem, a separate network card was installed for the GigE camera, and during the data calibration process between Lidar and image data, a preprocessing step was performed by setting the Region of Interest (ROI) to correct only the necessary areas, thereby resolving the issue.

V. DETECTION ALGORITHM

During the process of performing the docking operation to an aircraft, precise calculations of the docking target position are essential. The target position is determined in a three-dimensional space, considering the height, left, and right positions relative to the sensor location. According to Fig. 10, the aircraft door position is explored based on the learned information, and the two-dimensional positional information is calculated. Subsequently, the fused three-dimensional data is used to explore the position information of the docking target point.



Fig. 8. GigE camera data with aircraft and apron.

A. Multi-sensor Fusing

The collected aircraft body image data, as shown in Fig. 8, is based on the UV coordinate system, while the installed LiDAR sensor is composed of three-dimensional data as depicted in Fig. 7. To unify two or more data with different dimensions into a single coordinate system, a sensor data fusion process was conducted. For data fusion, the LiDAR data (x, y, z) was corrected and applied to the UV coordinate system, as illustrated in Fig. 10. The data fusion process can be divided into three main stages:

1) *Pre-processing*: In this stage, the raw LiDAR data (x, y, z) is obtained from the sensor. Calibration and correction processes are applied to align the LiDAR data with the U*V coordinate system. The pre-processed LiDAR data is then ready for further fusion steps.

2) *Fusion of data into U*V coordinate system*: The corrected LiDAR data (x, y, z) is integrated with the U*V coordinate system data. This fusion step allows combining the two different types of data into a unified U*V coordinate system. Fig. 9 shows proposed fusion 3D object recognition framework for aircraft.

3) *Coordinate system construction*: In this final stage, a complete coordinate system is constructed, incorporating the fused data. The fused data now provides three-dimensional information in the U*V coordinate system, enabling precise calculations and target exploration [16].

By following these three steps, the sensor data fusion process successfully unifies the U*V coordinate system with the 3D LiDAR data, allowing for accurate target positioning and exploration.

$$T = \frac{\log(1-p)}{\log(1-(1-e)^s)} \quad (1)$$

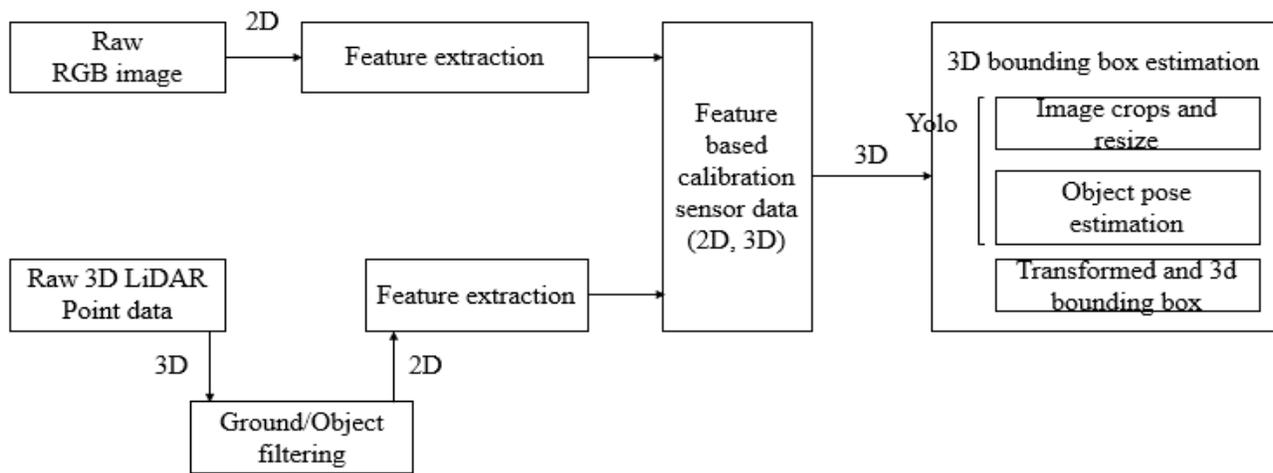


Fig. 9. Proposed fusion 3D object recognition framework for aircraft cabin door detecting algorithm.

The process of generating unified data between LiDAR and image data can be achieved through the following steps:

- 1) Ground and noise data filtering for setting the region of interest for LiDAR data [17].
- 2) Extraction of feature points from image data and setting reference points based on LiDAR data.
- 3) Matching of pixel-point feature points [18] to establish correspondence between LiDAR and image data.

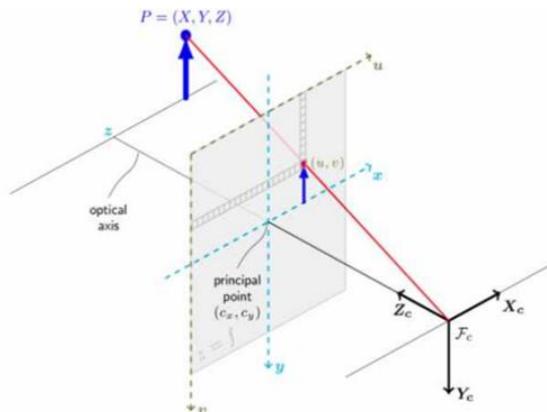


Fig. 10. Coordination system for IMAGE - 3D POINT(LIDAR).

To accomplish step 1, the ground and noise data filtering process, the 3D RANSAC algorithm was utilized to perform segmentation. As shown in Fig. 11, the data reflected from the aircraft body and the ground data were distinguished by comparing them with the RANSAC surface model, identifying inliers and outliers. To achieve a relatively high proportion of inlier data, the sampling frequency ρ in Eq. (1) was experimentally derived. ρ represents the probability of including a sample within the threshold of the standard plane model, ρ denotes the ratio of inliers to outliers in the entire dataset, and ρ is a parameter that adjusts the number of data to select the minimal set for the RANSAC process.

However, as shown in Fig. 11, the ground data adjacent to the aircraft landing gear had a higher outlier ratio than the inlier ratio. Thus, we did not apply ground filtering. To filter

even the data adjacent to the landing gear, a 3D plane model needed to be generated using actual ground point data. However, due to the characteristics of the 3D LiDAR sensor used in this study, there could be variations in accuracy between close and relatively distant data points, resulting in non-precise plane models with curvatures.

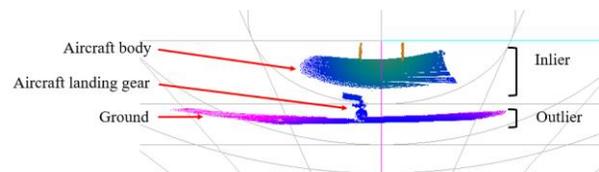


Fig. 11. Aircraft and ground 3d model data for 3d RANSAC.

To address this issue and filter data up to some parts of the aircraft landing gear, a model with curvature information was incorporated. This allowed us to filter some ground information along with the data. Fig. 12 illustrates the filtered data results after applying a preliminary correction to the image data, taking into consideration the filtered results that include some ground information.

After the ground data filtering, the next step involves matching feature points between the 3D LiDAR data and the image data to apply corrections. In this process, the landing gear and edges of the aircraft body in the 3D LiDAR data are set as feature points.

Similarly, corresponding pixel positions in the image data are stored for matching. Typically, to achieve accurate correction, one would extract feature points from the pixel panel and match them with the LiDAR data. However, in the case of the aircraft boarding bridge, the distance between the aircraft and the bridge is always constant, and the positioning of the aircraft during boarding is consistent, as shown in Fig. 12. Therefore, feature points were extracted based on a representative e-type aircraft model for data matching.

Moving on to the next step, the matched data between the LiDAR and image data were used for the final image-LiDAR matching process. However, as seen in Fig. 13, Fig. 14 and Fig. 15, errors in the matching results were observed for small parts or edge areas of the aircraft. The matching process could result

in approximately 15 cm errors in both horizontal and vertical directions. Despite these errors, the main objective of this study is not to achieve precise data matching but to identify the positions of aircraft doors and specific parts to generate 3D data. As such, including errors on the order of a few centimeters is acceptable for the study's objectives.

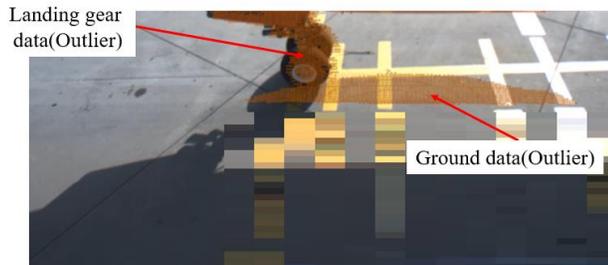


Fig. 12. Result of ground estimation with 3d RANSAC.



Fig. 13. Image-LiDAR matching deviation data results (Aircraft landing gear).

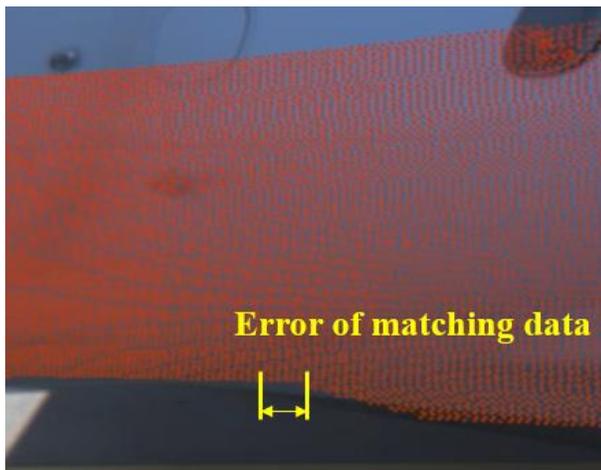


Fig. 14. Image-LiDAR matching deviation data results (Aircraft wing pan).

B. Vision based YOLO-V5 Model

The original YOLO algorithm was developed by Joseph Redmon, who is also the creator of a custom framework called Darknet. After five years of research and development leading to the third generation of YOLO (YOLOv3), Joseph Redmon announced his withdrawal from the field of computer vision. He discontinued further development of the YOLO algorithm

due to concerns that his research might be misused in military applications. However, he does not contest the continuation of research by individuals or organizations based on the initial concepts of the YOLO algorithm [19].

A Russian researcher and engineer who built the Darknet framework and implemented the three previous YOLO architectures using C, which was based on Joseph Redmon's theoretical ideas, collaborated with Chien Yao and Hon-Yuan to publish YOLOv4. As YOLO evolved, numerous object detection algorithms employing different approaches have achieved remarkable advancements. As depicted in Fig. 16, this development has led to the emergence of two primary concepts in architectural object detection: the One-stage detector and the Two-stage detector.

A common aspect among all object detection architectures is the process by which input image features are first compressed through a feature extractor (Backbone) and then forwarded to the object detector (comprising the Detection Neck and Detection Head), as shown in Fig. 16. The Detection Neck (or Neck) serves as a feature aggregator that combines and refines the features obtained from the Backbone, preparing them for the detection step performed by the Detection Head (or Head) [20].

The distinction here is that the Head is responsible for the actual detection, encompassing both localization and classification for each bounding box. The Two-stage detector executes these two tasks independently and subsequently combines their results (Sparse Detection), whereas the One-stage detector accomplishes them simultaneously (Dense Detection), as illustrated in Fig. 16. YOLO falls under the category of a One-stage detector, thus the name "You Only Look Once" [21].

As illustrated in Fig. 16, YOLOv4 conducted a series of experiments that integrated the most cutting-edge and innovative ideas in computer vision across different components of the architecture.

In the field of computer vision, object detection is a critical task involving the identification of objects within images or video frames, along with providing information about their positions and classes. Various architectures have been developed to tackle this task with the main categories you mentioned being one-stage and two-stage detectors [22].

1) *One-stage detector*: One-stage detectors, as the name implies, perform object detection in a single step. They directly predict the bounding box locations and class labels for each object instance. YOLO (You Only Look Once) is a prominent example of a one-stage detector. YOLO divides the input image into a grid and predicts the bounding boxes, object scores, and class probabilities for objects within each grid cell. YOLOv4 is an enhanced version of the YOLO architecture, which has improved detection performance through various innovative ideas and enhancements.

2) *Two-stage detector*: In contrast, two-stage detectors perform object detection in two steps. In the first step (region proposal stage), they generate a set of potential Regions of Interest (ROIs) likely to contain objects. These ROIs are then

refined and classified in the second step. The most well-known example of a two-stage detector is Faster R-CNN (Region Convolutional Neural Network). In Faster R-CNN, a Region Proposal Network (RPN) is used to suggest potential ROIs, which are then refined and classified using subsequent layers.

3) *Backbone*: This is a convolutional neural network (CNN) responsible for extracting features from the input image. The backbone network processes the image at various scales and levels of abstraction.

4) *Detection Neck*: The detection neck aggregates and combines the features extracted by the backbone. It enhances features for use by the detection head.

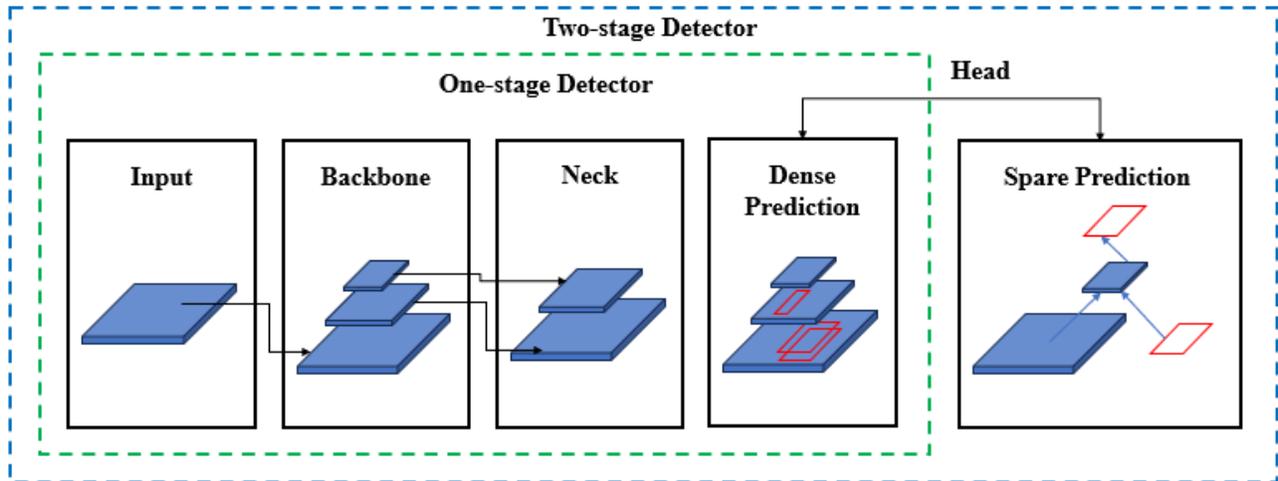


Fig. 15. Two concepts of architectural object detection [23].

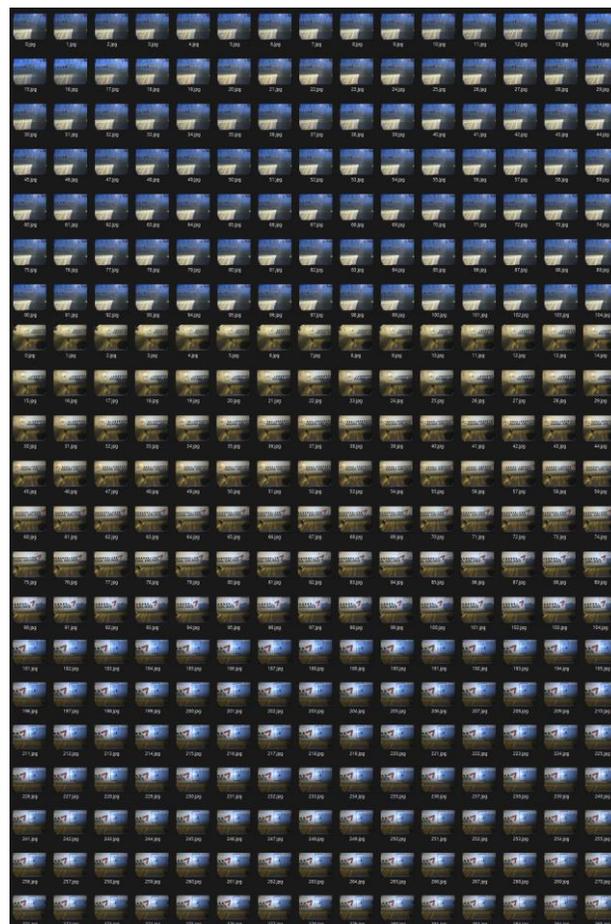


Fig. 16. Aircraft door training image dataset (real environment).

5) *Detection Head*: This is the final component of the architecture responsible for generating predictions. It takes features from the detection neck and generates predictions for object bounding box locations, objects scores (indicating the likelihood of object presence), and class probabilities.

The YOLOv4 architecture is built upon the YOLO concept and integrates various innovations to enhance detection accuracy and speed. The field of object detection is rapidly advancing, and there is ongoing development of new architectures and techniques to expand the potential scope in terms of accuracy and efficiency. Furthermore, there is continuous work on improving compatibility, scalability, and extensibility of development tools, as evidenced by the introduction of YOLOv5.

Even a month after the release of YOLOv4, the start of research for YOLOv4 and YOLOv5 was quite close [24].

To avoid confusion, Glenn decided to name his version of YOLOv5. Therefore, fundamentally, both researchers applied cutting-edge innovations in the field of computer vision at that time.

This led to the architectures of YOLOv4 and YOLOv5 becoming very similar, and this similarity resulted in many people expressing dissatisfaction with the name YOLOv5, as it did not seem to encompass significant improvements compared to the preceding version, YOLOv4. Additionally, Glenn did not publish a paper for YOLOv5, which further raised suspicions about YOLOv5 [25].

However, YOLOv5 had advantages in terms of engineering. YOLOv5 was written in the Python programming language instead of using C as in previous versions. This made installation and integration on IoT devices easier. Furthermore, the PyTorch community is larger than the Darknet community, indicating that PyTorch will receive more contributions and have greater growth potential in the future. Due to being written in two different languages on two different frameworks, accurately comparing the performance between YOLOv4 and YOLOv5 is difficult. Nevertheless, over time, YOLOv5 demonstrated higher performance than YOLOv4 in certain scenarios and gained some confidence within the computer vision community, alongside YOLOv4 [26].

C. Training Aircraft Image Dataset

YOLOv5, the following process is necessary for extracting training data:

1) *Data preparation*: Start by preparing the training dataset for object detection. You'll need training images along with the bounding box coordinates and class labels for each object.

2) *Data annotation*: Annotate the training images with bounding box and class information. This information will be used by the model to recognize and classify objects.

3) *Data transformation*: Convert the training data into a format that YOLOv5 model can understand. This format should adhere to the YOLO format.

4) *Model configuration*: Configure the YOLOv5 model. You can define the model architecture using PyTorch and load pre-trained weights.

5) *Training setup*: Set up the necessary hyperparameters and training options. This includes parameters like learning rate, batch size, and number of epochs.

6) *Training process*: Train the model. YOLOv5 is typically initialized with pre-trained weights on the COCO dataset, allowing you to fine-tune it for your specific data.

7) *Evaluation and testing*: After training, evaluate the model's performance using a validation dataset. Use the evaluation results to improve and fine-tune the model.

8) *Inference*: Use the trained model to perform inference on new images and detect objects.

In order to apply the YOLOv5 prediction model, it was necessary to undergo the process of training image data. To ensure applicability in various environments and aircraft scenarios, images of aircraft were collected within an airport setting, accounting for weather and time variations. The image data extraction was carried out for four types of aircraft from Korean national airlines and six types of aircraft from foreign national airlines. The collection points were situated in the upper cabin area, specifically the boarding gate area.

Moreover, to ensure the diversity of the image dataset, image extraction scenarios were defined not only for weather conditions such as rainy days but also for different conditions, as outlined in Table II, including daytime and nighttime scenarios. This approach aimed to capture a comprehensive range of conditions and variables to enhance the robustness and versatility of the model across different scenarios.

TABLE II. THE SCENARIOS FOR COLLECTING TRAINING DATA

No.	Time	Flight	Weather
1	08:00 ~ 11:00	Korean Aircraft C-type #1	Clear, Rainy, Foggy Day
		Korean Aircraft E-type #1	
		Korean Aircraft C-type #2	
		Korean Aircraft E-type #2	
		Foreign Aircraft E-type #1	
		Foreign Aircraft E-type #2	
2	13:00 ~16:00	Korean Aircraft E-type #1	Clear, Rainy
		Korean Aircraft E-type #2	Clear, Rainy
3	18:00 ~23:00	Korean Aircraft E-type #1	Clear Day
		Korean Aircraft E-type #2	Foggy Day
		Foreign Aircraft E-type #1	Clear Day

I wanted to collect images of more diverse aircraft appearances according to weather and time of day, but there were limits to collecting diverse data because the security requirements for each airline were different when collecting close-up images and 3D exterior data.

I have collected images of aircraft from various airlines across different time zones, similar to Fig. 16. The collected images amount to over 100,000. However, I filtered them to

include only meaningful image data, resulting in a training dataset composed of over 10,000 images.

I used the YOLOv5 model as the base for image training, with a transformation input size of 320x320. I conducted training with two different configurations: one with 100 to 300 epochs, and another with batch sizes of 10 to 30. I defined a decay step of 100 with a rate of 0.95. For the training system, I used an i7-13700K CPU model and an NVIDIA RTX3080 GPU model. The detailed specifications are as shown in Table III.

TABLE III. SPECIFICATION OF TRAINING SYSTEM

MODEL	SPECIFICATION
CPU	Intel i7-13700K 16 Cores, 24 threads
RAM	64GB
OS	Windows 10 / Pytorch
GPU	NVIDIA RTX3080 CUDA Core : 8704 10GB GDDR6X Memory Bus : 320bit

I adjusted the training parameters by classifying them into two categories and modifying the necessary epoch and batch settings while keeping the dataset size and types consistent, as shown in Fig. 17. In Fig. 17, I reduced the epoch size and increased the batch size to accelerate training. In Fig. 18, I performed training with parameters reduced to approximately half compare to the previous ones. As a result, fluctuations in the values, as shown in the graph in Figure, are observed. These variations are due to differences in the training positions and data types between day and night image data within the base dataset used for training.

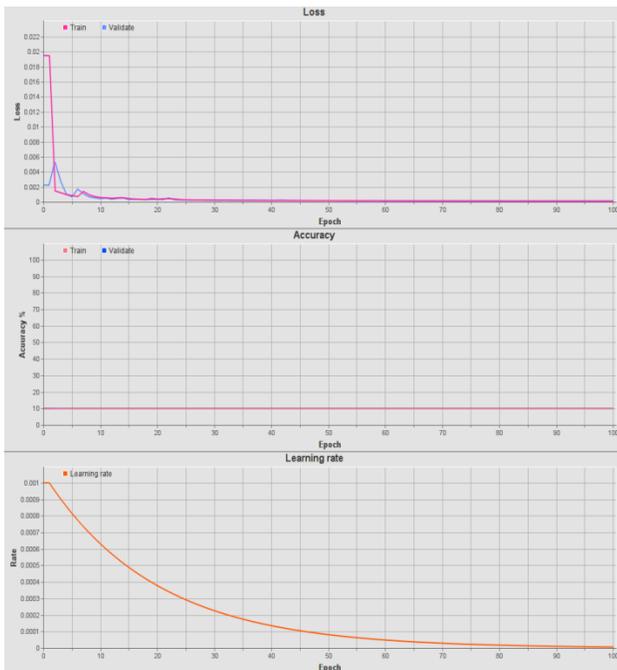


Fig. 17. Training result of image dataset (Epochs 100, Batch size 10, Learning rate 0.001).

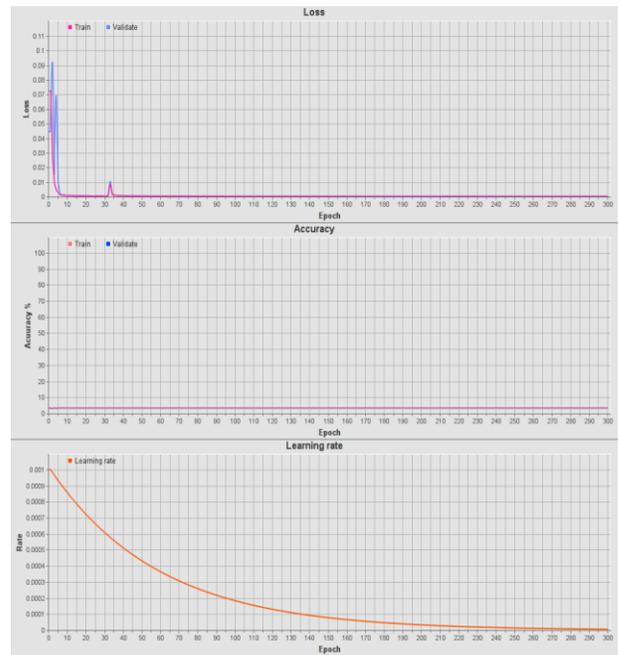


Fig. 18. Training result of image dataset (Epochs 300, Batch size 30, Learning rate 0.001).

In reality, the top line of the aircraft door exhibits clear distinguishing features during daytime, but during nighttime, it can experience loss of details due to the lighting from the boarding gate. In such cases, during YOLOv5 training, there's a possibility of spikes in values occurring because of images with entirely different types of distinguishing features, rather than converging towards the target values. This can result in fluctuations in values during training instead of a smooth convergence.

VI. EXPERIMENT AND RESULTS

To conduct testing of the aircraft door detection technology, it operates along the path between the boarding bridge and the aircraft, as shown in Fig. 19. Boarding bridge operations, as depicted in Fig. 19, are divided into three steps, and the status of the images at each operational position was verified in Fig. 19.

We performed YOLOv5 training using all the images from the operational steps as shown in Fig. 20. Subsequently, I obtained results for aircraft door detection from the depth images fused with 3D LiDAR data, as shown in Fig. 21.

In the process, similar to Fig. 20, we compared the fused depth data with actual inter-body distances to verify the consistency of the data generated through fusion. As indicated in the previously mentioned sensor installation layout, we placed laser distance measurement sensors not only the 3D LiDAR sensors but also at the bottom of the cabin of the boarding bridge. Through laser distance measurement sensors, we could measure the distance between the body and the boarding bridge for specific areas. To confirm this accurately, as shown in Fig. 21, we marked yellow reference points at the same positions as the laser distance measurement points within the images, collecting three-dimensional positional information within those pixels.

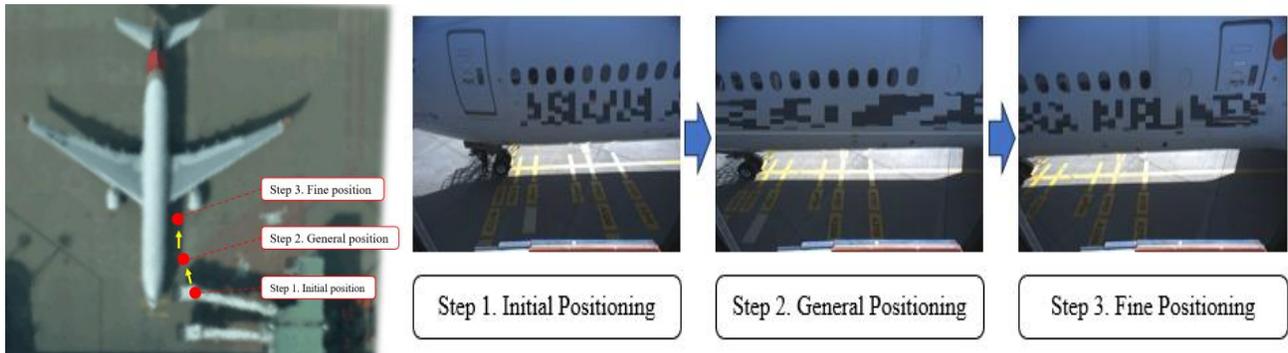


Fig. 19. Operational steps of automation Jet bridge and images data for each operational position.

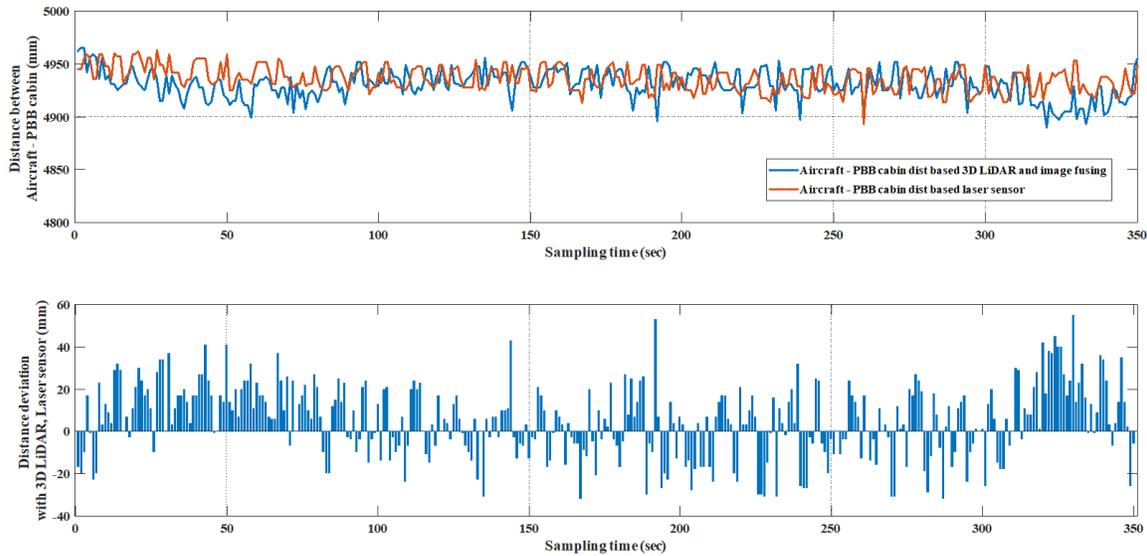


Fig. 20. Integrated (3D LiDAR – Vision) data, laser distance sensor data, and distance and deviation data between the aircraft and the passenger boarding bridge (PBB).



Fig. 21. Initial positioning steps of automation Jet bridge image data.

The collected data was represented as measured distance values, as shown in Fig. 20, and for data within a maximum range of five meters, it exhibited an error of within 5.8 cm due

to movement. Such a level of error can be attributed to factors such as the material, color, and light reflection on the aircraft body surface, as well as errors associated with the measurement location. In particular, considering both the inherent sensor error and cumulative values, the actual error was estimated to be within 3 cm.

Furthermore, aircraft doors detected through the fused data, as depicted in Fig. 22, were detectable for all orientations. During the daytime, the detection and recognition rate exceeded 95%, with false negatives primarily caused by backlighting from sunlight. Additionally, as seen in Fig. 23, the detection and recognition rates during nighttime were not significantly different. However, as shown in Fig. 24, the presence of nighttime lights installed on the top of the boarding bridge cabin and shadows resulted in light reflection on the aircraft door lines, causing the characteristics of the aircraft door lines to deteriorate, leading to situations where the aircraft doors were not recognized.

Nevertheless, this problem was resolved in the autonomous boarding bridge system operating scenario, as there was no alignment situation between the aircraft body and the boarding bridge during the closest proximity between them.



Fig. 22. Door detection in fine positioning steps at day time.



Fig. 23. Door detection in fine positioning steps at night time.



Fig. 24. Aircraft door error recognition results in fine positioning steps.

The average recognition rate during a single operation of autonomous boarding bridge action in both nighttime and daytime environmental conditions, for a duration of 2 minutes, was 95.3%. This rate exceeded the mAP value achieved by YOLOv5, but it was calculated for specific aircraft door recognition in a constrained environment, suggesting a higher recognition rate could be achieved.

However, there was a difference in real-time processing time between nighttime and daytime. Nighttime processing averaged around 10-11fps, whereas daytime processing averaged around 13-15fps, resulting in approximately a 3-4fps difference. Considering the maximum boarding bridge movement speed of 0.6m/s, it was sufficient for the autonomous driving system to operate effectively.

VII. CONCLUSION

This paper presents the development of technology for recognizing and identifying aircraft door locations to facilitate the development of an autonomous boarding bridge system. Due to the relatively slow movement characteristics of boarding bridges, the primary objective of this study was to achieve higher recognition rates and reduce errors rather than focusing on real-time processing speed. To accomplish this, we conducted the process of fusing image sensor and 3D LiDAR sensor data into a single dataset, successfully achieving a matching accuracy of over 98% and securing highly accurate data with errors of within 3 cm. Based on this data, we conducted research on aircraft door recognition using the YOLOv5 model.

In order to do this, we made efforts to acquire various aircraft models and airline image datasets, collecting over 100,000 images, of which more than 10,000 were used for actual training. Furthermore, we conducted a comparative analysis of recognition rates according to different time periods, ensuring a recognition rate of over 95% during both day and night hours. However, it should be noted that this paper was based on training data from images of aircraft obtained from six airlines, and for the application of autonomous boarding bridge systems to all aircraft and airlines, it is necessary to construct diverse image datasets for training purposes.

In future research, we intend to apply detection and recognition technology capable of handling a wider range of environments and scenarios.

ACKNOWLEDGMENT

This work was supported by Korea Evaluation Institute of Industrial Technology (KEIT) grant funded by the Korea government (MOTIE) (No.20023455, Development of Cooperate Mapping, Environment Recognition and Autonomous Driving Technology for Multi Mobile Robots Operating in Large-scale Indoor Workspace)

REFERENCES

- [1] Rajapaksha, Aruna, and Nisha Jayasuriya. "Smart airport: A review on future of the airport operation." *Global Journal of Management and Business Research* vol.20, no. 3, 2020, pp. 25-34.

- [2] Ihnsik Weon et al., "Development of autonomous aircraft docking system for passenger boarding bridge based on path planning and perception algorithm," in ICROS2022, 2022, pp. 277-278.
- [3] Park, B., I. Weon, and H. Kim. "Intelligent Docking System of Airport Passenger Boarding Bridge based on Perception Algorithm." KSPE 2023 Spring Conference, 2023, pp. 313-313.
- [4] Zhang, Panpan, et al. "Development on Equipment Operation Training and Intelligent Maintenance System based on Virtual Reality (VR) Technology for the Passenger Boarding Bridge." Highlights in Science, Engineering and Technology vol. 35, 2023, pp. 90-97.
- [5] Joo, Jeongha. "Framing a Guideline for Balancing Task Delegation of Human-Robot Collaboration in Automation processes: A Case Study on Automatic Passenger Boarding Bridge in Amsterdam-based Airport Autonomous Airside Operation.", 2023.
- [6] Schultz, Michael, Judith Rosenow, and Xavier Olive. "Data-driven airport management enabled by operational milestones derived from ADS-B messages." Journal of Air Transport Management vol. 99, 2022, pp. 102164.
- [7] Rajapaksha, Aruna, and Nisha Jayasuriya. "Smart airport: A review on future of the airport operation." Global Journal of Management and Business Research, vol. 20, no. 3, 2020, pp. 25-34.
- [8] Shanbhag, Vighnesh, and Jennifer Jacob. "Vehicle Detection and Traffic Control Using Sensor Technology." 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA). IEEE, 2023.
- [9] Ruiz, Ariel Y. Ramos, Luis J. Figueroa Rivera, and Balasubramanian Chandrasekaran. "A sensor fusion based robotic system architecture using human interaction for motion control." 2019 IEEE 9th annual computing and communication workshop and conference (CCWC). IEEE, 2019.
- [10] Jocher, Glenn, et al. "ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation." Zenodo, 2022.
- [11] Van Landeghem, Hendrik, and Annelies Beuselinck. "Reducing passenger boarding time in airplanes: A simulation based approach." European Journal of Operational Research vol. 142, no. 2, 2002, pp. 294-308.
- [12] ACI World Operational Safety Sub-Committee, ACI Airside Safety Handbook, 4th edition, 2010
- [13] ICAO Aerodrome Reference Code, ICAO Annex 14
- [14] Jaehn, Florian, and Simone Neumann. "Airplane boarding." European Journal of Operational Research, vol. 244, no. 2, 2015, pp. 339-359.
- [15] Nugroho, AriantoAjie, et al. "An Analysis and Design of Simulation Modelling for Airplane Passenger Boarding Strategy during Normal and New Normal Periods." Turkish Journal of Computer and Mathematics Education (TURCOMAT) vol. 12, no.13, 2021, pp. 7135-7153.
- [16] Fischler, Martin A., and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." Communications of the ACM vol. 24, no. 6, 1981, pp. 381-395.
- [17] Weon, Ihn-Sik, Soon-Geul Lee, and Jae-Kwan Ryu. "Object Recognition based interpolation with 3d lidar and vision for autonomous driving of an intelligent vehicle." IEEE Access vol. 8, 2020, pp. 65599-65608.
- [18] Weon, Ihn Sik, and Soon Geul Lee. "Environment recognition based on multi-sensor fusion for autonomous driving vehicles." Journal of Institute of Control, Robotics and Systems vol. 25, no. 2, 2019, pp. 125-131.
- [19] Jiang, Peiyuan, et al. "A Review of Yolo algorithm developments." Procedia Computer Science vol. 199, 2022, pp. 1066-1073.
- [20] Li, Wei, et al. "Headnet: An end-to-end adaptive relational network for head detection." IEEE Transactions on Circuits and Systems for Video Technology vol. 30, no. 2, 2019, pp. 482-494.
- [21] Huang, Rachel, Jonathan Pedoeem, and Cuixian Chen. "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers." 2018 IEEE international conference on big data (big data). IEEE, 2018.
- [22] Soviany, Petru, and Radu Tudor Ionescu. "Optimizing the trade-off between single-stage and two-stage object detectors using image difficulty prediction." arXiv preprint arXiv:1803.08707 (2018).
- [23] Khan, Farukh Aslam, et al. "A novel two-stage deep learning model for efficient network intrusion detection." IEEE Access, vol. 7, 2019, pp. 30373-30385.
- [24] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934, 2020.
- [25] Hu, Xuelong, et al. "Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network." Computers and Electronics in Agriculture, vol. 185, 2021, pp. 106135.
- [26] Wu, Tian-Hao, Tong-Wen Wang, and Ya-Qi Liu. "Real-time vehicle and distance detection based on improved yolo v5 network." 2021 3rd World Symposium on Artificial Intelligence (WSAI). IEEE, 2021.