Development of a Remote Safety System for Agricultural Robot

（ロボット農機遠隔安全システムの開発）

By

Chen Sixun

Dissertation

Submitted to Department of Environment Resources in the Graduate school of Agriculture

Hokkaido University, Sapporo, Japan, 060-8589 in particular fulfillment of the requirements for the degree of

Doctor of Philosophy

(Producer: Chen sixun, sixun.chen.b7@elms.hokudai.ac.jp, For any possible question please let me know by email)

Table of contents

Acknowledgment

I would like to express my deepest gratitude to Dr. Noboru Noguchi Professor for his invaluable support throughout my research and study in Japan. He has been tremendously helpful, both in my academic pursuits and in my daily life. His broad perspective and constructive suggestions have inspired me countless times. Without Professor Noguchi's patient guidance and valuable advice, I certainly could not have completed this research. I feel extremely honored to have had the opportunity to work under his guidance.

I would like to thank Dr. Kazunobu Ishii and Dr. Hiroshi Okamoto Although they were not my supervising professors, they still provided me with a wealth of valuable guidance, sharing their knowledge and expertise. They offered considerable assistance throughout the process of my research and in the writing of my thesis.

I would like to thank Dr. Ricardo Ospina. He provided a great deal of help and guidance throughout my research and experimental process, and I am grateful for his support.

I would like to thank the members of the laboratory for their support in my student life. Special thanks to Tsuyoshi Morita as a member of the remote team, he also provided me with a lot of help both in my research and daily life.

Lastly, and most importantly, I owe a huge debt of gratitude to my parents. They supported my overseas studies for many years and provided me with excellent conditions. Without them, I could not have completed my education. I am deeply thankful for their love and support.

List of Figures

List of tables

Notations

AP_p	Average accuracy of person class	%
AP_t	Average accuracy of tractor class.	%
mAP	Mean Average Precision at 50% Intersection over Union	%
box_{conf}	Bounding box confidence	
c	Speed of light	m/s
cls	The number of classes	
$class_{conf}$	Predicted class confidence	
c_x, c_y	Principal point of the camera	
C1, C2, C3, C4	Control point in PNP method	
C_x	Horizontal coordinates of the center point of the bounding box	pixel
C_y	Vertical coordinates of the center point of the bounding box	pixel
d_{i}	Squared Euclidean Distance from ground truth to i	pixel
d^2	Squared Euclidean Distance from detection point to ground truth point	pixel
$\overline{d_l}$	Average LiDAR detection braking distance	m
D_i	Distance of homogenous world point	m
D_{xi}	Distance of homogenous world point in the X-axis direction	m

f_x, f_y	Focal length in terms of pixels	pixel
FN	The number of missed objects	
FP	Number of falsely detected objects	
H	Height of the bounding box	pixel
H_0	Null hypothesis	
H_1	Alternative hypothesis	
k	Decay constant impulse function	
K	Matrix of the intrinsic camera parameters	
L_{oks}	Object keypoint similarity	
L6D	L6D signal	
n_1	Sample sizes of the pixel deviation.	
n_2	Sample sizes of the pixel deviation.	
p_i	Corresponding homogeneous image point	
p_i	Pixel located at the center of that bounding box's bottom edge	
p_c	Actual calculate point	
P	P-value with reference to the next adjacent interval.	
P_i	Homogenous world point	
R	Rotation matrix	
$[R\|T]$	Matrix of the extrinsic parameters	

s^2	Object’s segmented area	pixel
$\overline{S_d}$	Average LiDAR detection braking distance	m
S	Scale of the ground truth object	pixel
S_1	Tractor's stopping position relative to the operator's position 1	m
S_2	Tractor's stopping position relative to the operator's position 2	m
S_d	Breaking distance of the tractor	m
$\overline{t_l}$	Average LiDAR detection braking time	s
$\overline{t_d}$	Vehicle's horn average blast time (breaking time)	s
t'	T-value with reference to the next Adjacent interval	
T	Translation matrix	
TP	Number of correctly detected objects	
v_{i}	Ground truth visibility flag for keypoint i	
v'	Degrees of freedom	
V_{max}	Tractor max speed	m/s
V_P	Pedestrian ‘speed	m/s
W	Width of the bounding box	pixel
$\overline{x}_1$	Sample means of the pixel deviation.	pixel
$\overline{x}_2$	Sample means of the pixel deviation.	pixel
$\bar{x}$	Sample mean of pixel deviation.	pixel

X_C	X-coordinate in the camera coordinate system	
X_w	X-coordinate in the world coordinate system	
y_i	Bounding box center point's coordinate in the Y-direction	
y_g	Ground truth's pixel coordinate in the Y-Direction	
Y_C	Y-coordinate in the camera coordinate system	
Y_w	Y-coordinate in the world coordinate system	
Z_C	A scale factor for the image point	
Z_w	Z-coordinate in the world coordinate system	
δ	Impulse function	
ε_p	Pixel deviation	pixel
σ	Standard deviation of pixel deviation	pixel
${\sigma_1}^2$	Sample variances	
${\sigma_2}^2$	Sample variances	
σ_p	Pooled variance	

Acronyms and abbreviations

AI: Artificial intelligence,

ANN: Artificial neural network,

ARRC: Agricultural Robotics Research Community,

CANBUS：Controller Area Network bus,

CLAS: Centimeter-level augmentation service,

CNN: Convolutional neural network,

CSPNet: Cross Stage Partial Network,

DF: Degree of freedom

EV: Electric vehicle,

FGseg: Field-ground segmentation,

GBRT-based: Gradient Boosted Regression Trees,

GDS: Geomagnetic Direction Sensors,

GIS: Geographic information system,

GNSS: Global navigation satellite system,

GPS: Global positioning system,

GPU: Graphics processing unit,

IMU: Inertial measurement unit,

ISOBUS: International Organization for Standardization bus,

ISO: International Organization for Standardization,

LiDAR: Light detection and ranging,

MAFF: The Ministry of Agriculture, Forestry, and Fisheries,

mAP: Mean Average Precision,

MEC: Multi-Access Edge Computing,

MLIT： Ministry of Land, Infrastructure, Transport and Tourism,

MS COCO: Microsoft Common Objects in Context dataset,

NUC: Next unit of computing,

PASCAL VOC: Pattern analysis statistical modeling and computational learning visual object classes,

PNP: Perspective-N-Point method,

Q-Q plot: Quantile-Quantile plot,

QZSS: Quasi-Zenith Satellite System,

R-CNN: Region-based convolutional neural networks,

R-FCN: Region-based fully convolutional network,

RTK: Real-time kinematics,

SL: Self-labeled dataset

UN DESA: United nations Department of Economic and Social Affairs

SVM: Support vector machine,

TCP: Transmission control protocol,

TECU: Tractor’s Electronic Control Unit,

TOF: Time of flight,

Web RTC：Web Real-Time Communication；

YOLO: You Only Look Once.

CHAPTER 1. INTRODUCTION

1.1 Research Background

1.1.1 The situation and challenges of agriculture

On November 15, 2022, the United Nations announced that the global population had reached a significant milestone of 8 billion, a figure that has been a source of concern since the 1960s due to the potential for a population explosion. Data from the United Nations Department of Economic and Social Affairs (UN DESA), Population Division (2022), indicates that the current world population is more than triple what it was in the mid-20th century. Specifically, the population, which stood at 2.5 billion in 1950, doubled over approximately 37 years, surpassing 5 billion in 1987. Projections suggest that it will take over 70 years for the population to double again, reaching beyond 10 billion by 2059. This rapid growth exerts considerable pressure on the global food supply chain, necessitating a substantial increase in global agricultural production. The challenge of overpopulation leads to heightened demand for resources, resulting in overconsumption and an accelerated depletion of resources. This trend not only strains agricultural resources but also limits the availability of industrial resources, thereby undermining the potential for improved per capita output and sustainable development. According to the Food and Agriculture Organization of the United Nations, a more than 60% increase in agricultural production is required to sustain this burgeoning population.

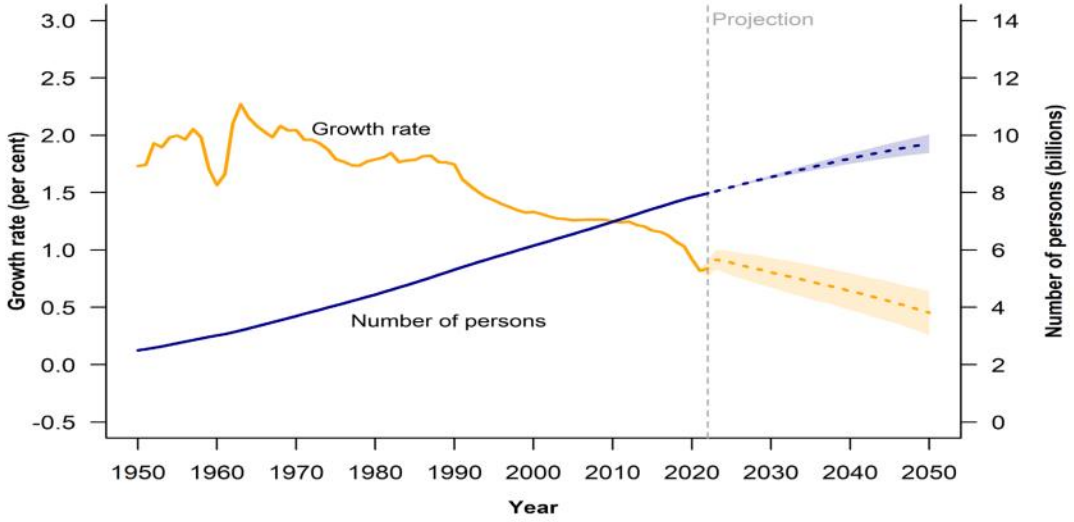


Figure 1.1-1 Global population size and annual growth rate.

As the world's population increases as shown in *Figure 1.1.-1*, the demand for food resources intensifies, yet the number of individuals engaged in agriculture continues to decline. The Ministry of Agriculture, Forestry, and Fisheries (MAFF) of Japan regularly conducts censuses of the country's agriculture and forestry sectors (2021). From 1976 to 2020, the census data show a consistent year-on-year decrease in the number of agricultural workers, falling from 4.9 million households in 1976 to 1.7 million households in 2020. The 2020 census further revealed that the number of agricultural management entities decreased by 408,000 (18.9%) compared to 2015 as shown in Table 1. Despite the ongoing reduction in the number of agricultural workers, the rate of decline has not decelerated. This trend is attributed to a concurrent decrease in new entrants to agriculture. Specifically, the number of new agricultural workers dropped from 65,030 in 2015 to 55,870 in 2020, marking the lowest figure in recent years. The decline in new agricultural workers can be linked to the perception of farming as a career with unstable income, particularly for newcomers. Such instability may discourage young people from pursuing careers in agriculture. Moreover,

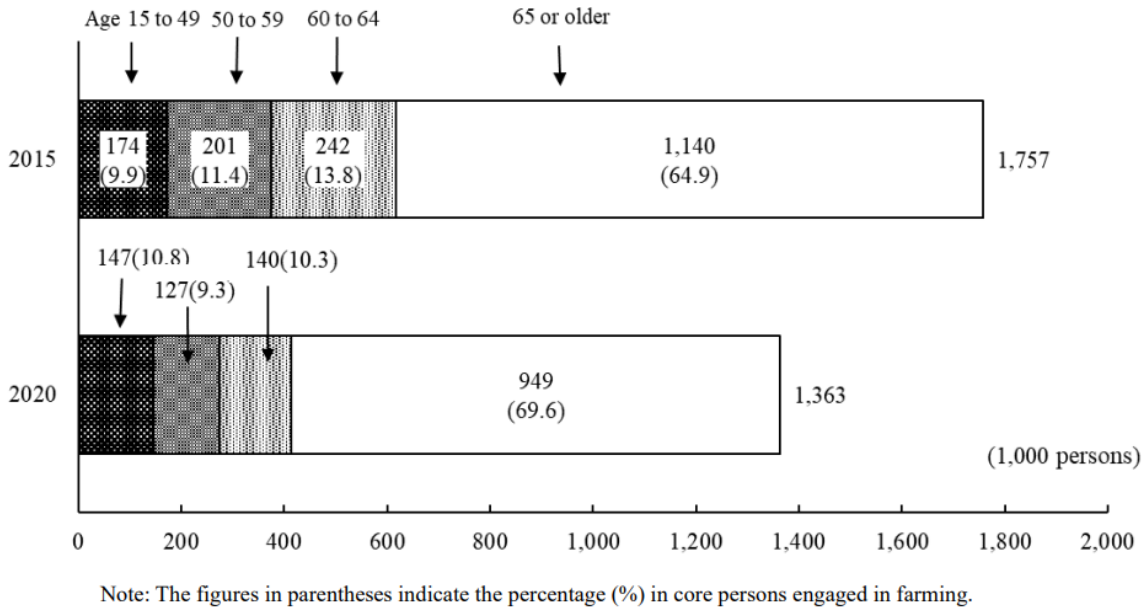
constraints on the size of operational farmland limit the overall efficiency of agricultural production and management. As farmers' consumer needs grow, the increase in agricultural income fails to match the rise in household expenses. Consequently, most farm households engage in non-agricultural activities, with their non-agricultural income often surpassing their earnings from agriculture. However, with the saturation of traditional industries, the availability of nearby part-time income opportunities for rural laborers decreases, thereby exacerbating the talent drain as individuals migrate to large cities in search of employment.

Table 1 Number of Japan’s farm households.

Year and prefecture	Total	Commercial farm households	Non-commercial farm households
Feb. 1, 2015	2,155,082	1,329,591	825,491
Feb. 1, 2020	1,747,079	1,027,892	719,187

Source: The 96th Statistical Yearbook of Ministry of Ministry of Agriculture, Forestry, and Fisheries, 2021

In addition, the percentage of aged 65 or older among core persons engaged in farming of individual management entities is 69.6%, rising by 4.7 points from 5 years ago as shown in *Figure 1.1-2*. The aging of the agricultural population can lead to a decline in the quality of agricultural labor and an increase in the rate of accidents in agricultural production, both of which are detrimental to the sustainable development of agriculture.



Note: The figures in parentheses indicate the percentage (%) in core persons engaged in farming.

Figure 1.1-2 Age Structure of Japan's Agricultural Population

Source: The 96th Statistical Yearbook of Ministry of Ministry of Agriculture, Forestry, and Fisheries, 2021

1.1.2 Agricultural robot tractor

To address the various challenges faced by traditional agriculture, the development of smart agriculture is seen as a necessary condition for sustainable agricultural growth. Robot tractors, capable of replacing humans in agricultural production, are an integral part of smart agriculture. Firstly, to attract more new agricultural workers, it is essential to ensure that professional farmers achieve an income commensurate with the level of economic development, necessitating an expansion of their operational farmland scale. If the pace of expanding the scale of farmland lags significantly, and the income level from agricultural activities remains consistently lower than non-agricultural ones, the expectations for farming income among the young and middle-

aged labor force will not improve, inevitably leading to a substantial loss of this workforce. From 2015 to 2020, the average farmland area managed by Japanese farm households increased from 2.5 hectares to 3.1 hectares (MAFF, 2021), which is beneficial for increasing the agricultural income of these households. However, this also increases the production burden on the farmers, making it crucial to enhance the production efficiency per household. According to a 2019 operational report on the use of robot tractors in Iwamizawa City as shown in Table 2, autonomous driving led to a 77% reduction in operational time and a 57.8% decrease in labor costs.

Table 2 Changes in Labor Time for Wheat Cultivation Brought About by the Introduction of Robot Tractors.

Total time for Wheat Cultivation (h/ha)	1.51
Robot Tractors available Time in Total Time (h/ha)	0.46
Time reduced due to the introduction of robot tractors (h/ha)	0.35
Reduction Rate of Operation Time in Total Operation Time (%)	23.18

Source: Demonstration of Smart Agriculture Model Utilizing Local 5G Technology, 2020

This undoubtedly contributes to the improvement of production efficiency and income per household. Additionally, modern lifestyles and urbanization trends have led younger generations to seek job and life opportunities in cities. With the rapid development of social media and other platforms, contemporary youth are increasingly aspiring to the improved quality of life offered by urban living, as compared to their predecessors. Agricultural labor, which often requires intensive physical work and long hours, is becoming less appealing to the younger generation.

Agricultural robots can assist farmers in completing tedious tasks such as seeding, weeding, and harvesting. These innovations can free young people from the burdens of heavy agricultural activities, allowing them to focus more on their quality of life. Finally, traditional agriculture often relies on the "intuition" and "experience" of the farmers, which is not conducive to the participation of young people who are willing but lack experience and methodology in agriculture. In contrast, smart agriculture is "data-based agriculture." It excels in transforming abstract phenomena into describable, computable data, which is more conducive to learning and research by those without experience. This can also contribute to the growth of the agricultural population.

Robot agricultural machinery typically performs operations such as seeding, spraying, weeding, and transporting. They complete these tasks following pre-defined route maps, without the need for human drivers. However, they must accurately determine their location during operations, as well as their kinematic and dynamic states, and whether their environment poses any danger. Therefore, sensors are crucial for robotic tractors, serving a role analogous to the eyes, nose, and ears for humans, enabling the robot to understand its own state and its surroundings. There are various sensors available for use in robotic tractors. For instance, visual sensors capture images to assist the robot in recognition and decision-making. Laser sensors (LiDAR) use lasers for precise object distance measuring, usually available in 2D and 3D forms. Ultrasonic sensors measure the travel time of sound waves for distance gauging. Inertial Measurement Units (IMUs) measure the robot's motion, tilt, and acceleration. Global Navigation Satellite Systems (GNSS) provide location information using satellites. Geomagnetic Direction Sensors (GDS) determine the robot's orientation using geomagnetic field data. Contact sensors prevent collisions by detecting physical

contact. These sensors collectively enable robotic tractors to operate autonomously and efficiently in various agricultural tasks. For example, researchers at Japan's Hokkaido University developed a robot tractor that can do farm work within a 5 cm error (Noguchi et al., 1997). The detection of obstacles by an agriculture robot is an aspect of safe operation, and it was shown that lasers are an economical and practical device for this purpose (Kise et al., 2005). However, a two-dimensional (2D) laser cannot detect an obstacle whose height is lower than the laser's scanning plane. Guo et al., (2002) developed a safety alert system that uses two ultrasonic sensors, and this system can detect the position of a moving object in the vicinity of agricultural machinery and generate a warning signal to ensure the operator's safety. The sensors have good stability, but they cannot recognize visual information and thus cannot perform more complex tasks, such as identifying obstacle types. Chateau et al., (2000) proposed a method for guiding agricultural vehicles using lasers. Firkat et al., (2023) proposed a 3D-LiDAR based algorithm for Field-ground segmentation (FGseg) that is more effective in handling sloped environments compared to traditional methods. Gai et al., (2021) developed a system that utilizes a depth camera for the navigation of agricultural vehicles. It is particularly useful for robots operating under canopies of tall plants such as corns and sorghums, where GPS signal is not always receivable. Shen et al., (2019) developed a navigation method based on multiple cameras and ultrasonic sensors integration technology for orchard mobile robot. The robot can move stably and precisely along the road of the semi-structured orange orchard. Each of the above-described systems has its advantages and disadvantages, and it was suggested that the use of fusion sensors as an alternative can better cope with the problems caused by the systems' disadvantages (Castanedo 2013). The fusion method of spatial information from LiDAR and machine vision was described by Sun et al., (2021). The

proposed method is able to achieve a balance between detection accuracy and detection speed.

In summary, if robot agricultural machinery can be used efficiently and on a large scale, it could help solve and alleviate the issues of insufficient labor and inefficiency in traditional agriculture.

1.1.3 Deep Neural Networks in Agriculture

The mathematical technique of backpropagation, published by Rumelhart et al., (1986) laid the foundation for the rapid development of deep learning by enabling the effective training and optimization of neural networks. This technique has become a core component of modern deep learning algorithms, exerting a profound impact on the entire field of artificial intelligence (Plaut, 1986). Particularly after 2010, with the significant enhancement in computing power, deep neural networks began to be practically applied across various domains. The ImageNet project (Deng et al., 2009), initiated by Professor Fei-Fei Li of Stanford University, assembled a database of over 14 million labeled images, providing crucial data support for the application of deep learning in fields such as image recognition. Convolutional neural networks (CNNs) have been widely used in many fields (Alzubaidi 2021). In the field of autonomous driving, CNNs have been applied to object detection, lane detection, and pedestrian recognition, enabling vehicles to perceive their environment and make decisions based on real-time data (Bojarski et al., 2016; Arnold et al., 2019). Similarly, the application of Deep Neural Networks (DNNs) in agriculture signifies a revolutionary shift, providing numerous novel solutions to enduring challenges. Utilizing DNNs in

areas such as agricultural image detection, yield prediction, and automation enables the processing and analysis of vast data sets from varied sources, including satellite imagery, sensor data, and weather forecasts. This integration enhances the precision and efficiency of agricultural practices, making DNNs a pivotal tool in the pursuit of sustainable agricultural development. For instance, Ma et al., (2023) employed Unmanned Aerial Vehicles (UAVs) and multispectral imagery in conjunction with CNNs to predict the yield of winter wheat. Ferreira et al., (2022) utilized deep neural networks combined with 3D imaging for the identification of dairy cattle. Bhat et al., (2023) developed a crop selection system using Gradient Boosted Regression Trees (GBRT-based) deep learning surrogate models. This system assists farmers in selecting the most suitable crops according to different soil types. Overall, these studies offer valuable insights into the application of advanced machine learning techniques for the development of smart agriculture. To develop a system capable of efficiently monitoring the safety of remote-operation robot tractors, we can also leverage these deep learning technologies.

1.1.4 Remote safety system

According to the Safety Assurance Guidelines for Agricultural Machinery Autonomous Navigation issued by the MAFF of Japan (2017). Human monitoring is necessary during the work of agricultural robots. Robot agricultural machinery is categorized into three levels: level 1, level 2, and level 3 as shown in *Figure 1.1.3*. Level 1 is defined as robot tractor or combine relies on GNSS and automatic steering, allowing humans to operate hands-free; Level 2 is defined as robot agricultural machinery that, although it does not require riding, needs human visual monitoring from nearby； Level 3 is defined as not requiring on-site visual monitoring, instead

needing supervision of multiple robot agricultural machines from a distant monitoring center, and includes the ability to move between fields. The objective of this research is to develop a remote safety system to ensure the safety of Level 3 robot monitoring

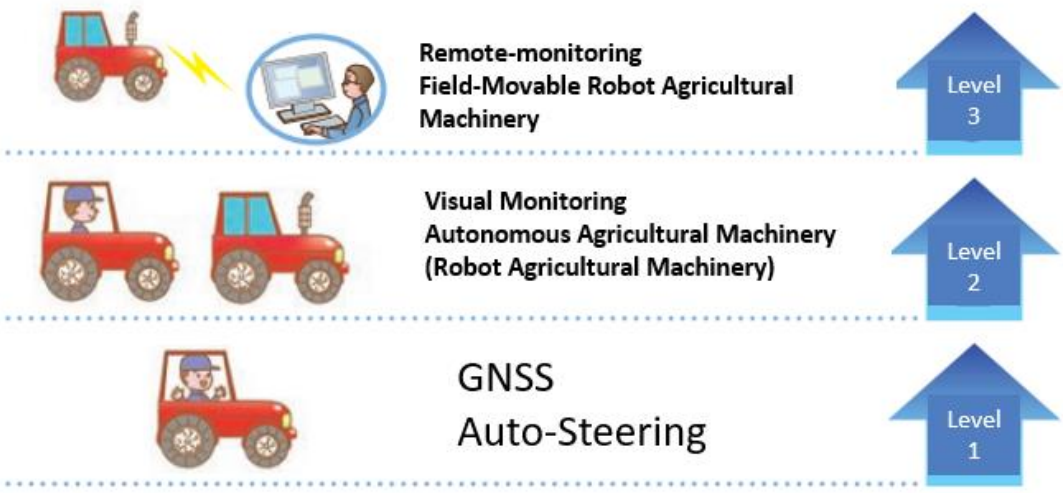


Figure 1.1-3 Three levels of robot agriculture machinery defined by MAFF.

operations.

Level 1 is currently widely used by many agricultural practitioners, and Level 2 was also commercialized in 2018 and is being sold by various companies. Level 2 brings benefits such as significant improvement in labor shortage, improved work accuracy and efficiency, and the ability for agricultural practitioners to allocate time to other tasks. However, Level 3 is expected to bring further efficiency and labor-saving improvements.

Source: What is remote-monitoring robot agriculture machinery? 2023

The utilization of robot tractors can contribute to the sustainable development of agricultural societies, with the assurance of safety being a prerequisite for their deployment. With the increasing labor shortage in various regions, the demand for large-scale uses of agricultural robots is also increasing. A multi-robot tractor system

for conducting agriculture field work was developed by Zhang et al., (2017) Their system can improve work efficiency by having multiple robots working together. A key point in the development of the multiple-robot system is that when multiple robots work together in one field, a single human is sufficient to monitor all of the robots. This method has saved labor but is not as efficient as having multiple robots working in different fields. If multiple robots work separately in different fields, there is no risk of collision between the robots, but more human monitors are needed to monitor the operations of the robots based on the current conditions (Noguchi 2000). For agricultural robots working in different fields, the deployment of a human monitor for each field is not as efficient as desired. Developing a remote-control system to monitor all of the robots active in various fields from just one monitoring room would be an effective way to save labor costs and improve the robots' efficiency. Albiero et al., (2022) also suggests a roadmap for the Agricultural Robotics Research Community (ARRC) to optimize agricultural operations by using multiple robot tractors with lower power instead of a single large machine, thus aiming to enhance logistics, operational geometry, and energy efficiency through robotics.

1.2 Research objectives

To enable the stable, cost-effective, and efficient monitoring of multiple tractors in operation, we have developed a remote safety system for level 3 robot monitoring. The system uses a monocular camera installed on each tractor to collect visual data from the front of the robot tractor. The visual data are transmitted to 'the cloud' via internet for analysis, and instructions from the system are sent back to the robot

tractors for execution. Moorehead et al., (2012) developed a system can also control autonomous tractor in remote end for orchard maintenance. Compared to this system, our newly developed remote safety system has several advantages.

(1) Compared to communication through a local area network, communication over the Internet enables remote monitoring of autonomous tractor robots located at greater physical distances. A single human operator can monitor multiple robot tractors working simultaneously at vastly different locations, thus increasing efficiency and providing labor-savings.

(2) Differentiated from Geometric Detector, Appearance Classifier or other traditional machine learning method, deep-learning obstacle-detection methods are closer to human thinking and logic and have better credibility.

(3) Our primary data processing and computations are conducted remotely. The majority of the human operator's work is carried out in a remote-control monitoring room in a controlled environment, relying solely on signals to control the robot. This improves the system's stability and makes it easier to maintain and update.

(4) Compared to depth cameras and 3D-LiDAR, monocular cameras and 2D-LiDAR have the advantages of low cost and easy maintenance. Furthermore, the method of remote data processing eliminates the need to install high-performance computers on each individual working tractor. All these advantages will help ordinary farmers make use of these new technologies (Noguchi and Barawid 2011).

(5) The network environment is relatively poor in most rural farming areas, the remote safety system developed in this paper can operate normally with a minimum upload speed of 30kbps at the tractor end, and it also utilizes local safety sensors. If

network delays or interruptions occur due to unforeseen circumstances, safety issues will not arise.

1.3 Organization of thesis

Chapter 1 introduces the current state of world population growth, the status and challenges of agriculture in Japan, the benefits and necessity of using robotic tractors, the development and application of deep neural networks, and the main purpose of this thesis is to develop a remote safety system for robot tractor's stable, cost-effective, and efficient monitoring in operation for level 3 robot agriculture machinery monitoring.

Chapter 2 introduces Research Platform and Materials of the system.

The objective of chapter 3 is to train a deep learning model for use in a remote safety system. The aim is to detect humans or other tractors that appear in the field. First, the methods for data preparation are introduced, including data collection, label assignment, and data augmentation. Then, the process and parameters of model training are discussed, along with how to perform optimization.

The objective of this chapter 4 is to utilize the YOLOv5s detector developed in the previous chapter, based on deep learning models, to accomplish the localization of obstacles appearing in front of the robot tractor.

The objective of this chapter 5 is to refine the positioning results of the detector developed in Chapter 2 and the positioning method proposed in Chapter 3. Given the relative accuracy disadvantage of monocular vision compared to other high-precision sensors, it is important to correct the results. This study employs statistical methods to analyze the detection results and has adjusted accordingly.

Chapter 6 presents some farm experiments conducted using the remote safety system, exploring the operation of the system on various platforms, its performance in different environments, and the remote capabilities of the system.

Chapter 6 is the summary of the research.

CHAPTER 2. RESEARCH PLATFORM AND MATERIALS

2.1 Introduction

This chapter introduces the information about the tractor platforms and sensors used in this study. As shown in *Figure 2.1-2*, this model, which depicts the future of agriculture showcased in Iwamizawa City, includes the remote safety system developed in this paper (primarily the red part). The hardware of the system shown in *Figure 2.1-1* consists of two main parts: one for the robot tractor (edge part) and one for the remote control of the robot tractor (remote-control part). In this study, the edge part includes the robot, encompassing the robot vehicle, robot control computer, data relay computer (NUC), and various sensors (monocular camera, 2D-LiDAR, IMU, GNSS). The remote-control part is situated in a robot monitoring room, a certain distance from the robot itself, and comprises a remote-control computer, monitor, and an alarm. The edge part and the remote-control part engage in data exchange via a

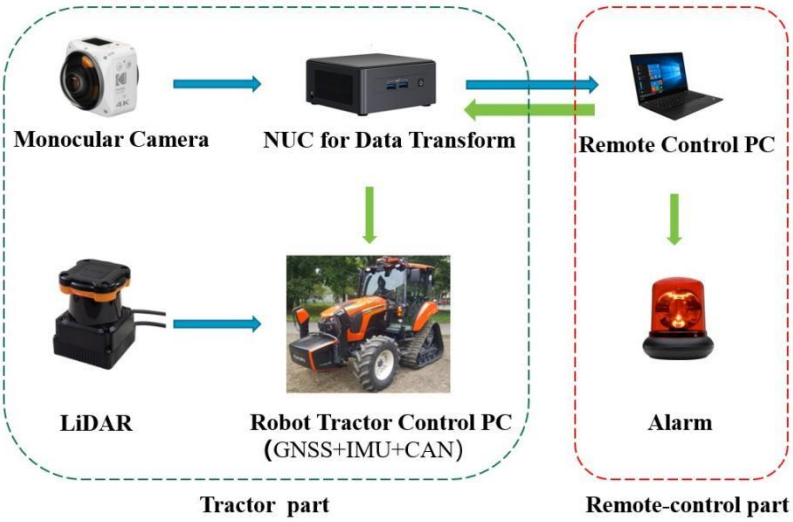


Figure 2.1-1 Overview of the remote safety system.

wireless network through internet. The Vehicle Robotics laboratory of Hokkaido University developed the robot vehicle used in this study.

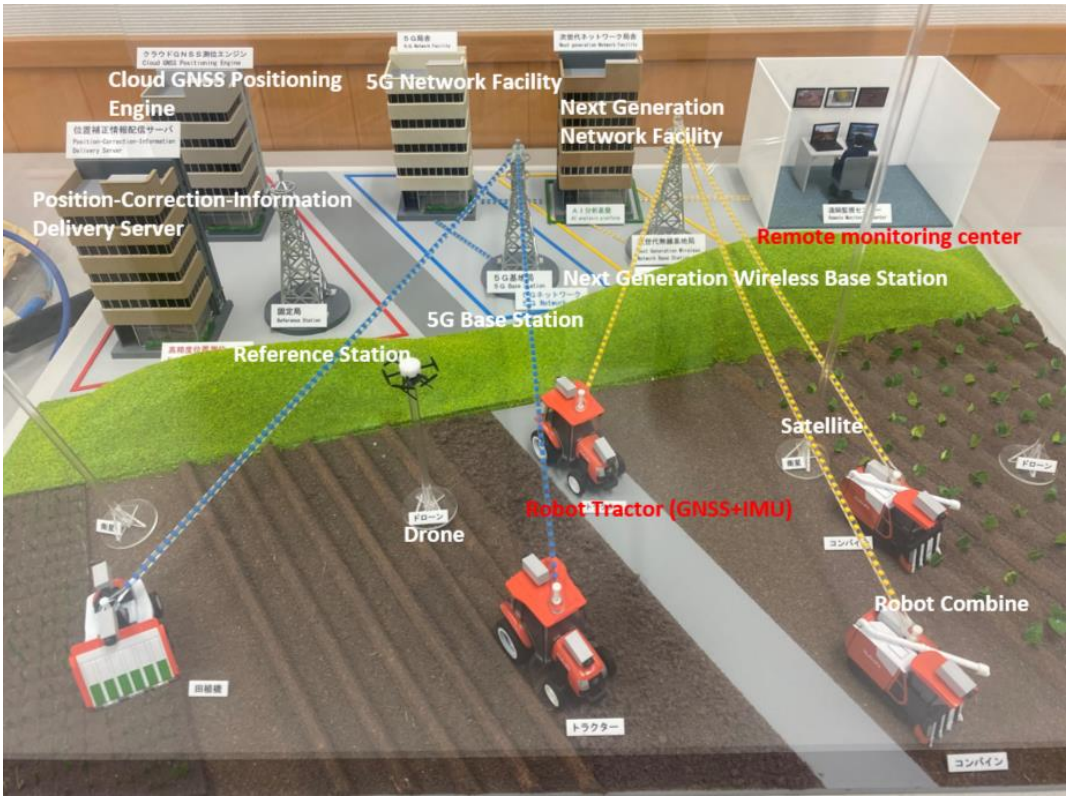


Figure 2.1-2 Future agricultural image in Japan.

Source: Iwamizawa City, 2023

Through remote monitoring from the monitoring center, various operations of multiple agricultural machines can be controlled, and information around the agricultural machines is transmitted from the robot machines. The robot agricultural machines determine their positions using location information from GNSS satellites and navigate through fields as shown in the diagram. GIS, which stands for "Geographic Information System," reflects geographic information on maps.

The new system's remote safety system workflow as shown in *Figure 2.1-3* can be described as follows. (1) If the camera captures a picture, the camera will transmit the raw image data to the NUC in the tractor via a universal serial bus (USB) cable. The NUC sends the image data to the workstation in the remote monitoring room through the internet by connected wireless Wi-Fi. The transmission control protocol (TCP) is used for data transmission and Web Real-Time Communication (RTC) is used for Video stream transmission.

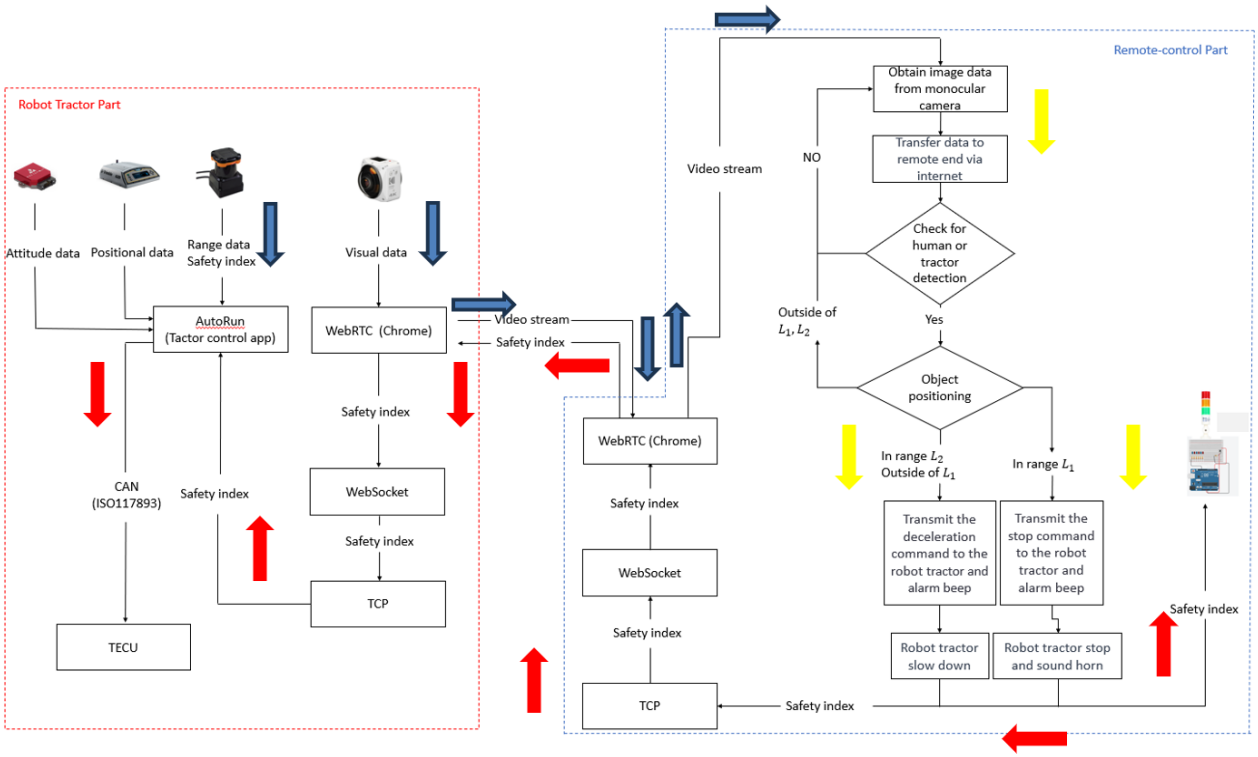


Figure 2.1-3 Flow chart of the remote safety system.

After the image data are received, the workstation will input the received raw image data into the CNN-based image analysis program, and the program will analyze the raw image by a YOLOv5s neural network as shown in *Figure 2.1-4*. If an object (e.g., a human or a tractor) is detected, the program will assign a serial number to the detected object and mark it in the image in the form of a bounding box. Next, the coordinates of the object in the image are transformed to the world coordinate system, and the distance between the object and the tractor is calculated to generate a safety index. Table 3 summarizes the safety index details. When more than one object is detected simultaneously, the sizes of all safety indexes in each frame are compared, and the largest safety index is output. Each safety index is transmitted back to the robot tractor NUC via the internet.

Table 3 Definition of the safety index.

Safety index	**0**	**1**	**2**	**3**
Safety condition	Safe	Safe	Attention	Stop
Tractor Condition	Normal	Normal	Slow down and beep	Stop and beep
Obstacle position	No detection	$> (L_1 + L_2)$ & $> R$	$> L_1$ & $< (L_1 + L_2)$	$< L_1$ *or* $< R$
Alarm	Standby	Standby	Beep	Beep

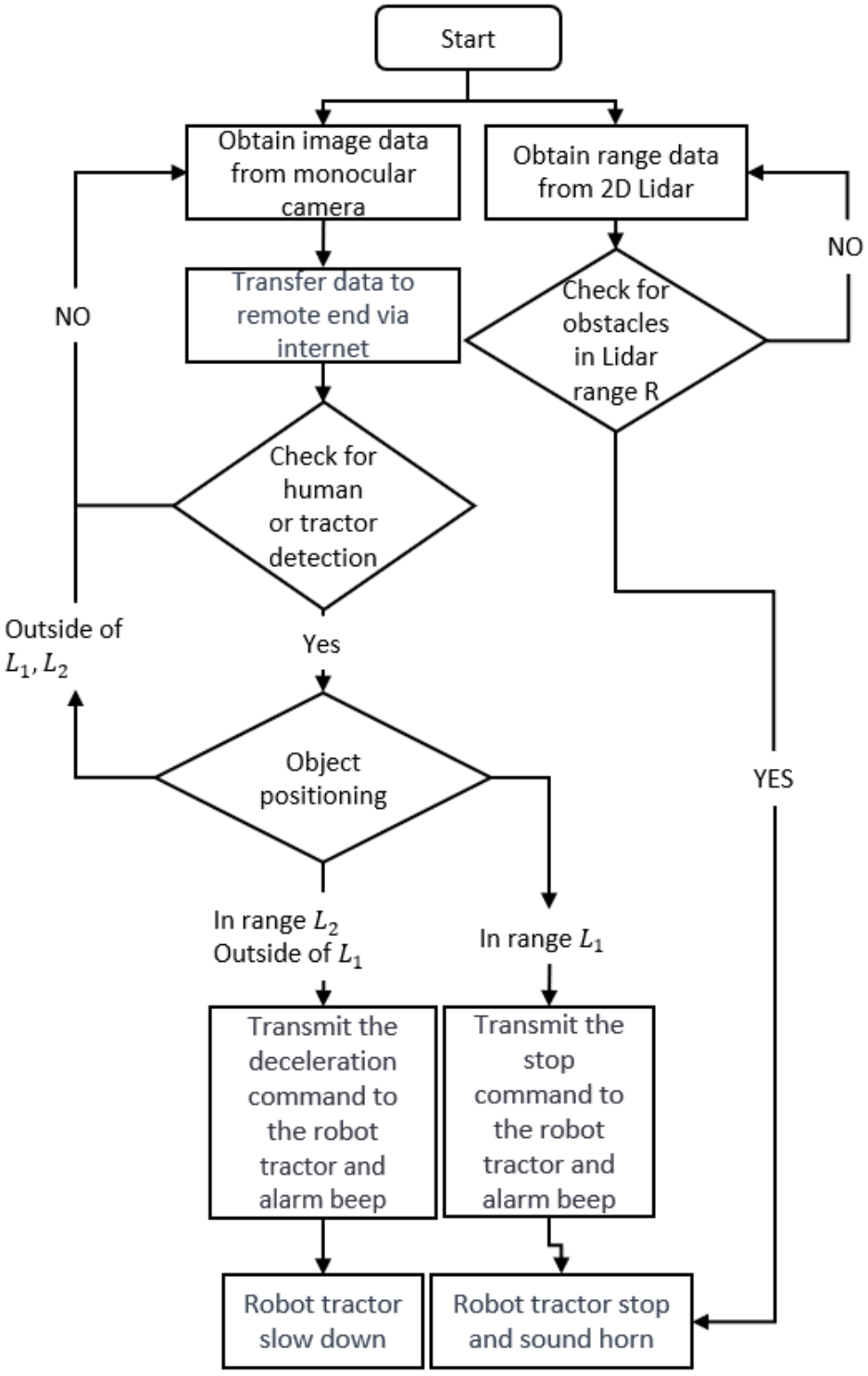


Figure 2.1-4 Flow chart of the detection part.

Finally, the robot tractor's NUC transmits the received safety index via USB cable to the robot tractor's control computer, which is directly connected to the LiDAR sensor and accepts the safety index from both the LiDAR sensor and the NUC. The control computer prioritizes the execution of the tractor's action corresponding to the safety index with the larger number, and it sends the command results to the tractor via a controller area network (CANBUS) according to the safety conditions described in Table 3 The ranges of L1, L2, R, and W are shown in *Figure 2.1-5*. Where L1 is the visual stopping zone range, L2 is the visual deceleration zone range, R is the LiDAR stopping zone range, and W is the visual zone width.

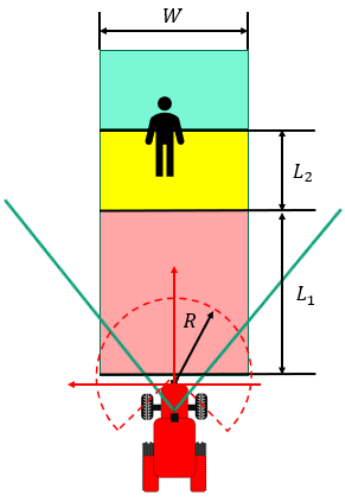


Figure 2.1-5 Segmentation of the system's detection area.

In this study, we used 8 m for L1, 4 m for L2, 5.5 m for R, and 3.2 m for W, where L1 is the visual stopping zone range, L2 is the visual deceleration zone range, R is the LiDAR stopping zone range, and W is the visual zone width.

2.2 Edge part devices

The edge part devices include a robot control computer, a NUC, a monocular camera mounted on top of the tractor, and a 2D Lidar mounted on the front of the tractor. The camera (PixPro 4KVR360, Kodak) is connected to a 'next unit of computing' (NUC) and is used as an input device for capturing images with a resolution of 1280 × 720 pixels. The NUC is a data relay station between the remote-end computer and the robot tractor controller. A compact, lightweight 2D-LiDAR sensor (UTM-30LX, Hokuyo, Osaka, Japan) directly connected to the TECU (Tractor's Electronic Control Unit) is used to detect obstacles other than people and tractors, and its use can also prevent accidents due to a network delay or missed detection. GNSS is used for the positioning and navigation of the tractor robot, while IMU is utilized to measure the acceleration and angular velocity of the tractor robot, providing information about the robot's orientation and motion. All these devices are installed on a robot tractor developed by the Vehicle Robotics Laboratory at Hokkaido University.

2.2.1 Research platform

The tractors used in this study include two half-crawler tractors and two wheel-type tractors, specifically the Kubota MR1000A crawler-type as shown in *Figure 2.2-1*, Kubota MR1000A wheel-type, Yanmar EG105 crawler-type, and Yanmar EG83

wheel-type. The main development and testing were conducted on the Kubota MR1000A crawler-type.

Figure 2.2-1 Kubota MR1000A crawler-type.

Additionally, an Electric Vehicle (EV) robot (Yamasaki and Noguchi, 2023) was employed as the experimental platform. The vehicle platform is a golf cart modified by TOYOTA TSUSHO CORPORATION for agricultural use. Its powertrain comprises a battery and electric motor sourced from a hybrid vehicle. The base vehicle boasts a loading capacity of 600 kg.

All the vehicles have been robotically modified to enable direct communication between the PC and the vehicle's CAN. The robot tractors comply with the ISO11783 standard, which aims to standardize communication between vehicles and connected electronic devices. Utilizing this feature, we can also more conveniently develop a safety monitoring system that is compatible with various platforms produced by

different companies. Moreover, leveraging the "implement control-centric" feature of the ISO BUS, we can send control commands to the TECU mounted on the tractor via a connected PC to execute stop, decelerate, and accelerate operations, as well as receive work data from the tractor (speed, gear, PTO, hitch, etc.). Kubota MR1000A crawler-type specifications as shown in Table 4.

Table 4 Kubota MR1000A crawler-type Specifications

Overall Length:	4,505 mm
Overall Width:	2,860 mm/1980 (retract sensors)
Overall Height:	2,725 mm
Weight:	3,950 kg
Volume:	3.769 L
Max Output:	73.5 kw
Max Rotation	2,600 rpm
Fuel Type:	Diesel
Fuel Tank Volume:	120 L
Overall Length:	1,310 mm
Overall Width (Each):	450 mm
Travelling Speed:	-31.8 km/h ~ 31.9km/h

2.2.2 Navigation sensors

A global navigation satellite system (GNSS) and inertial measurement unit (IMU) sensors are used to obtain information about the robot's location and attitude. The GNSS provides centimeter-level augmentation service (CLAS) to enhance the positioning accuracy.

The Global Navigation Satellite System (GNSS) is a satellite-based technology that provides accurate location, speed, and time information, widely applied in maritime, aviation, geological exploration, and daily life. GNSS technology relies on a group of satellites broadcasting signals captured by ground receivers to determine their position and time. Its main advantage is the ability to provide precise positioning under almost any weather conditions. Main GNSS systems include the United States' GPS, Russia's GLONASS, the European Union's Galileo, and China's BeiDou. This study's robot tractors initially used GPS as the primary navigation system. However, with the launch of Japan's Quasi-Zenith Satellite System (QZSS) in 2018 as shown in *Figure 2.2-2*, the tractor robots developed in our laboratory began utilizing QZSS as the main navigation system (Wang and Noguchi, 2019). The Centimeter Level Augmentation Service (CLAS) is a high-precision positioning service from Japan, forming a part of the Quasi-Zenith Satellite System (QZSS) (Cabinet Office, 2017). CLAS provides centimeter-level positioning accuracy enhancement as shown in Table 5 by broadcasting satellite signals. It is primarily used in applications requiring high precision, such as precision agriculture, land surveying, and construction engineering. Utilizing CLAS services, users in Japan and its surrounding regions can obtain more accurate positioning information.

Table 5 Comparison of Lateral Error.

	CLAS	RTK-GNSS
Maximum Error (m)	0.13	0.14
Standard Deviation (m)	0.05	0.04

Source: Listening to Professor Noguchi of Hokkaido University: Challenges and Future Prospects of Smart Agriculture in Mountainous Areas ,2023

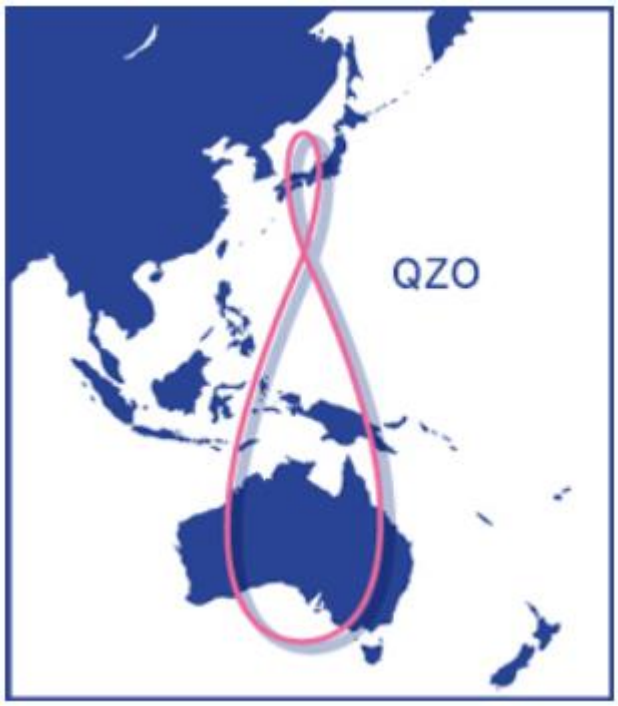


Figure 2.2-2 Coverage of QZSS and its trajectory.

QZSS (Michibiki) has been in operation since November 2018 to develop a satellite positioning service that can be used stably in all locations at all times. This system is compatible with GPS satellites and can be utilized with them in an integrated fashion. In this way, the satellite positioning service environment was advanced dramatically.

Source: What is the Quasi-Zenith Satellite System (QZSS)? 2017

The L6D signal, which transmits centimeter-level augmentation information, is not relayed via GPS, thus requiring a specialized receiver. The Alloy (Trimble) receiver as shown in *Figure 2.2-3* used in this study can receive the L6D signal provided by QZSS.

Figure 2.2-3 Alloy (Trimble) GNSS receiver.

Alloy can receive GPS L1 C/A, L2E (L2P), L2C, L5 and QZSS L1 C/A, L1C, L1 SAIF, L1S3, L2C, L5, LEX/L6D signal.

Source: Trimble Alloy Data sheet, 2023

An Inertial Measurement Unit (IMU) is a device that determines the orientation and motion state of an object by measuring its acceleration and angular velocity. It typically consists of an accelerometer and a gyroscope, which measure the object's acceleration and rotational speed to infer its position and orientation. By continuously integrating these measurements, it is possible to calculate the object's position,

velocity, and orientation information. IMUs are commonly used in applications such as navigation, attitude control, and motion analysis. An IMU contains 3-axis (pitch, roll, and yaw) gyros and 3-axis accelerometers. A VN-100 IMU as shown in *Figure 2.2-4* was used in this study and its specification is shown in Table 6.

In this study, the position error caused by the vehicle's roll and pitch angle are called error by inclination. This error can be corrected by compensating the vehicle's position.

Figure 2.2-4 VN-100 IMU.

Table 6 VN-100 IMU Specifications

Roll angle:	±180°
Pitch angle:	±90°
Yaw angle:	±180°
Angular Resolution:	< 0.05°
Input Voltage:	3.2 V ~5.5 V

2.2.3 Safety sensors

This study utilizes a monocular camera (PixPro 4KVR360, Kodak) as shown in *Figure 2.2-5* and a 2D Lidar (UTM-30LX, Hokuyo, Osaka, Japan) as shown in *Figure 2.2-6* as safety devices. While depth cameras and 3D Lidar sensors can provide more accurate and comprehensive data, they are relatively expensive. Using cost-effective sensors is advantageous to encourage farmers to adopt these new technologies. Therefore, an import focus of this research is to establish a reliable safety system using these basic sensors that meets the criteria for safety and stability and its specification is shown in Table 7, Table 8.

Figure 2.2-5 PixPro 4KVR360, Kodak

Table 7 PixPro 4KVR360 Specifications

	Image Sensor A	Image Sensor B
Effective Image Sensor Pixels:	20.68 Megapixels	20.68 Megapixels
Total Image Sensor Pixels	21.14 Megapixels [1/2.3” BSI CMOS]	21.14 Megapixels [1/2.3” BSI CMOS]
Focal Length	1.633 mm	1.257 mm
F number	F2.4	F2.4
Focus	Fix Focus	Fix Focus
Field of View	Max. 197 Degree [For Front Mode (16:9)] Max. 155 Degree [For VR Mode (2:1)]	Max. 235 Degree
Focusing Range	30 cm~	30 cm~
Jacks	USB 2.0 (Micro 5 pin USB), HDMI (Type D) Stereo Microphone Input (Diameter of 2.5mm)	
Power	Li-ion battery (LB-080), 3.6V 1250 mAh, in-Camera Charging	

Figure 2.2-6 2D Lidar (UTM-30LX)

Table 8 Hokuyo UTM-30LX 2D laser rangefinder Specifications

Voltage:	Voltage: 12 VDC ±10%
Current:	Max: 1 A, Normal: 0.7 A
Light source:	Semiconductor laser diode (λ = 905 nm) Laser safety Class 1 (FDA)
Detection Range:	0.1 m ~ 30 m (White Square Kent Sheet 500 nm or more)
Input Voltage:	3.2 V ~5.5 V
Scan angle	270°
Accuracy	0.1 m ~ 10 m: ±30 mm, 10 m ~ 30 m: ±50 mm
Angular resolution	0.25° (360°/1,440 steps)
Scan time	25 ms/scan (40 Hz)
Connection	USB: 2 m cable with type-A connector

2.3 Remote-control Part Devices

The remote-control unit is a high-performance computer or server used to process image data sent from the tractor's end and provide control commands in return. A workstation (ThinkPad P71, Lenovo) equipped with an Intel Xeon E3-1535M V6 processor with 16 GB of RAM and an NVIDIA Quadro P4000 graphic card and a 16-GB RAM PC is used to process the images to determine whether or not there is an obstacle and to calculate the positions of obstacles. An alarm (handmade; Arduino) alerts the operator when an obstacle is detected. The surveillance of robots is conducted in remote monitoring centers, which are typically equipped with multiple screens. The central screen displays GIS, enabling real-time monitoring of the robot tractor's location, GNSS accuracy, speed, PTO status, and more. Surrounding this are four screens, each dedicated to displaying images from a specific tractor. These images are processed to detect and mark obstacles (such as people or other tractors) in real time, calculating the distance between the target and the tractor, and assessing the robot's safety status. If a safety risk is identified, the system automatically triggers an alarm and sends commands to the robot to slow down or stop. To date, four remote monitoring centers have been established at the Iwamizawa City Data Center, the Faculty of Agriculture at Hokkaido University, the Hokkaido University Smart Agriculture Education Center as shown in *Figure 2.3-1*, and the Tsurunuma Town Improvement Center. The remote-control computers are mobile, and a movable alarm light system is also in place, enabling the surveillance of robot tractors from various

locations. The monitor during an alarm as shown in *Figure 2.3-2*. The remote-control computer located behind the screen wall.

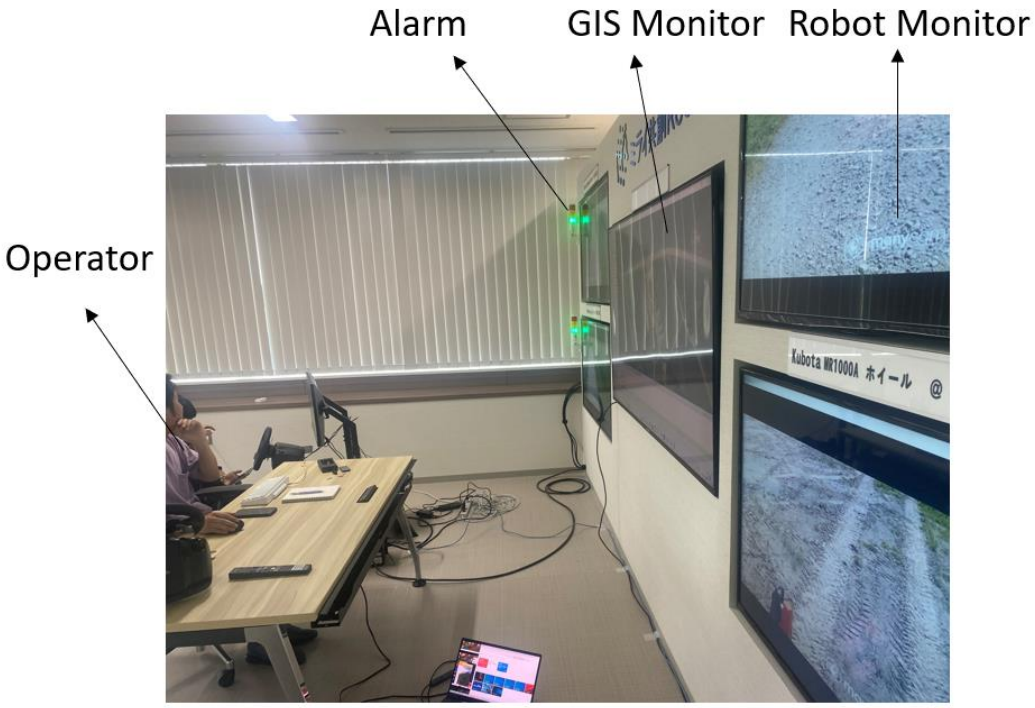


Figure 2.3-1 The configuration of the monitoring center.

The Figure shows the monitoring center located at the Smart Agriculture Education Center of Hokkaido University. A monitor is supervising robot tractors. The central large screen displays GIS, while the four screens on the left and right display the frontal image information of four different robots, with an alarm next to each screen. The remote-control computer is located behind the wall.

Figure 2.3-2 The Robot Monitor during an alarm.

The Figure depicts the monitoring center located at the Iwamizawa City Information Center. The front camera of the robot triggers an alarm and stops the robot upon detecting a human intruding directly ahead.

2.3.1 Remote control unit

The remote-control unit, consisting of a computer or server, primarily functions to process and analyze image data transmitted from tractors in real time using neural networks and to send back instructions accordingly. Therefore, it is essential to equip it with a high-performance graphics processing unit (GPU). Additionally, to ensure rapid data transmission, a stable network connection is vital. In this study, the remote-control unit utilized is a workstation (ThinkPad P71, Lenovo), the specifications of which are detailed in the Table 9.

Table 9 ThinkPad P71, Lenovo Specifications

Processor:	Intel® Xeon® E3-v6 Processors for Mobile Workstations 7th Generation Intel® Core™ Processors
Graphics:	NVIDIA Quadro P4000,8GB memory
Memory	DDR4 64 GB
Power Supply:	170 W
Battery:	8 Cell (96 WHr)

2.3.2 Display and alarm unit

The remote-control unit is connected to monitors and alarm lights as shown in Figure 2.3-3 for observing the surroundings of the tractor and for drawing the attention of the surveillance personnel. An alarm (handmade; Arduino-based) alerts the operator when an obstacle is detected. The alarm lights are of two types: a fixed version for use in the monitoring room and a portable version for use when outdoors, both controlled by the remote-control PC via an Arduino development board as shown in Table 10, when the

remote-control PC assesses a danger level based on the image information transmitted by the robot, the alarm light will flash red and the buzzer will sound.

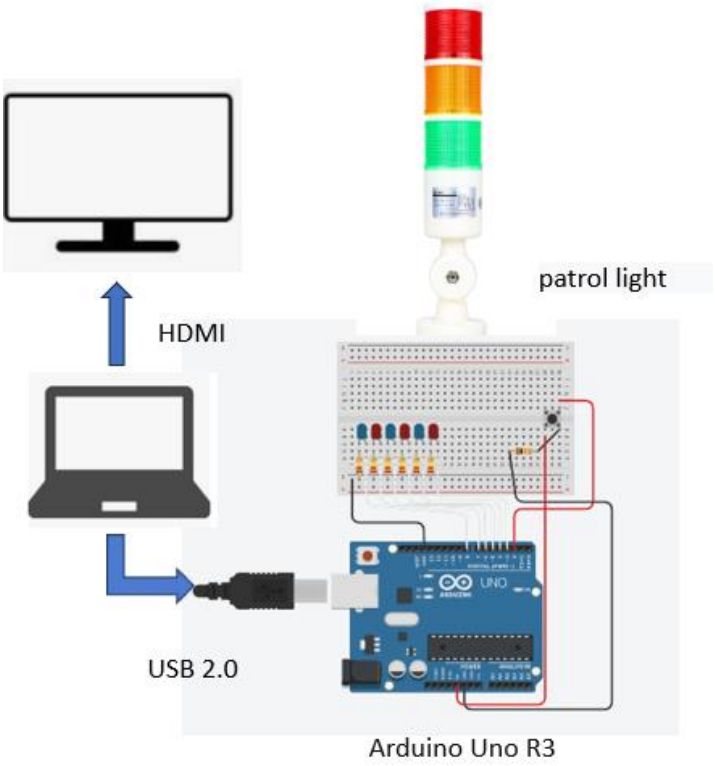


Figure 2.3-3 The configuration of the Display and Alarm Unit

Table 10 Arduino Uno R3 Specifications

Processor:	8-bit AVR® RISC-based microcontroller
Memory:	AVR CPU at up to 16 MHz/ 32KB Flash 2KB SRAM/ 1KB EEPROM
Power	2.7-5.5 volts

CHAPTER 3. TRAINING OF THE YOLO-BASED DETECTION MODULE FOR FIELD OBSTACLES

3.1 Introduction

The objective of this chapter is to train a deep learning model for use in a remote safety system. The aim is to detect humans or other tractors that appear in the field. First, the methods for data preparation are introduced, including data collection, label assignment, and data augmentation. Then, the process and parameters of model training are discussed, along with how to perform optimization. Additionally, to evaluate the performance of the model, other models are trained for comparison.

Regarding traditional approaches applied to human detection, from the classifiers point of view there are several classification algorithms used to perform pedestrian detection, most of which are applied in a supervised approach, e.g., support vector machine (SVM), artificial neural network (ANN), or boosting algorithms (Brunetti et al., 2018). As deep-learning approaches there are some mainstream object detection architectures, including 'you only look once' (YOLO) (Redmon et al., 2016), R-FCN (Dai et al., 2016), R-CNN (Girshick et al., 2014), Faster R-CNN (Ren et al., 2015), Mask R-CNN (He et al., 2017), and SSD (Liu et al., 2016), which are all CNN-based. Generally, there is no specific guideline on which model researchers or practitioners should use. The choice of model varies depending on factors such as memory requirements, accuracy, and time cost, and the choice should be determined based on the specific detection task.

The YOLO (You Only Look Once) series of architectures is typically used for real-time object detection tasks. Its core idea is to treat object detection as a single regression problem, directly mapping from image pixels to bounding box coordinates and class probabilities. Its main workflow is as shown in figure 3.1-1: YOLO first divides the input image into an SxS grid. Each grid cell is responsible for predicting the objects centered in its area. Then, each grid cell predicts a fixed number of bounding boxes. For each bounding box, the model predicts the center's x, y coordinates, width, and height, as well as a confidence score for the presence of an object. Each grid cell also predicts the class probabilities of each object in the cell (in this study, tractors and humans), ultimately leading to the detection results.

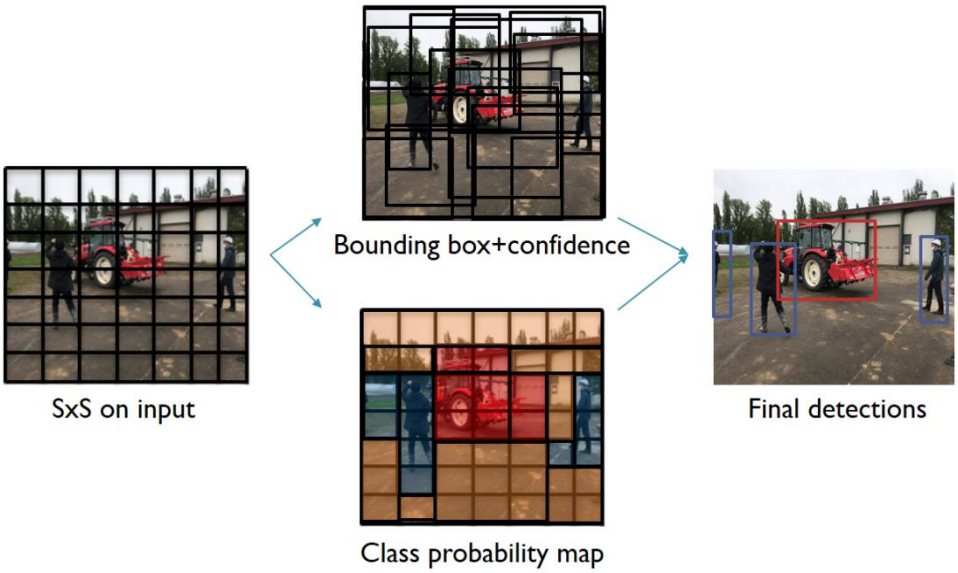


Figure 3.1-1 The workflow of YOLO-based architectures.

YOLO first divides the input image into an F grid. Then, both a confidence score class probability map and a confidence score is computed per each estimated bounding box. Finally, we can get the detection results by confidence score.

3.2 Dataset Preparation

Compared to autonomous vehicles operating in cities, robotic tractors generally have fewer detection targets, and we thus chose the two most frequent obstacles in farmland as detection targets: humans and tractors. Other possible obstacles are detected using a LiDAR sensor. Human detection is a challenge that robot tractors must overcome before being applied to real-world scenarios because human safety is most important issue. In general, the following challenges are encountered in creating datasets for humans and tractors:

(1) To ensure the generalizability of the model as shown in *Figure 3.2-1*, it is essential to cover various types of humans and tractors, environmental backgrounds, and lighting conditions.

(2) The self-labeled dataset differs significantly in shooting conditions from the COCO dataset. The COCO dataset provides uniformly high-resolution, wide-angle images with multiple objects, whereas the self-labeled dataset varies in quality due to diverse image sources as shown in Table 11. The impact of mixing these two types of data needs to be explored.

(3) Given the disparity in the number of provided human and tractor samples, it is necessary to balance the sample sizes of both to prevent model bias towards a particular type and avoid overfitting the model to the dominant class.

(4) Tractors generally vary significantly in appearance and come with various types of implements. Detailed annotation rules need to be established, which may affect the model's ability to recognize them accurately.

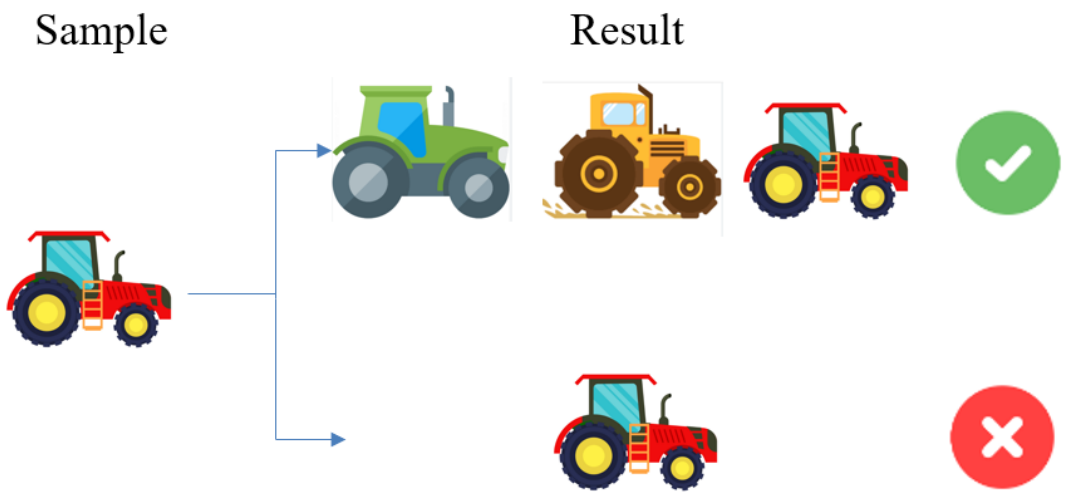


Figure 3.2-1 The generalizability of the model.

3.2.1 Data collection

The training data set can be divided into two categories: Human data from the MS COCO (Microsoft Common Objects in Context) dataset (Lin et al., 2014), and tractor and human data from the internet and video recordings of robot tractors as shown in *Figure 3.2-2*. We selected 4366 human datasets from the open-source MS COCO dataset. The self-labeled datasets were obtained from the video recordings of robot tractors owned by the VeBots Laboratory of Hokkaido University and online tractor images, which consisted of 1430 items labeled with tractors and humans. Due to the fact that a single category in the COCO dataset often contains many instances per image, whereas in our self-labeled data, a category typically includes fewer instances, we strive to maintain consistency between the two datasets by selecting images from the COCO dataset that contain fewer instances. Although the COCO dataset does not include a specific category for tractors, it does have a category for trucks. Considering the similarity in appearance between trucks and tractors, and in an effort to enhance the generalizability of our model, we selected a subset of the COCO dataset that contains both trucks and humans. The

annotation data for the truck category is utilized as a proxy for the tractor category in our study.

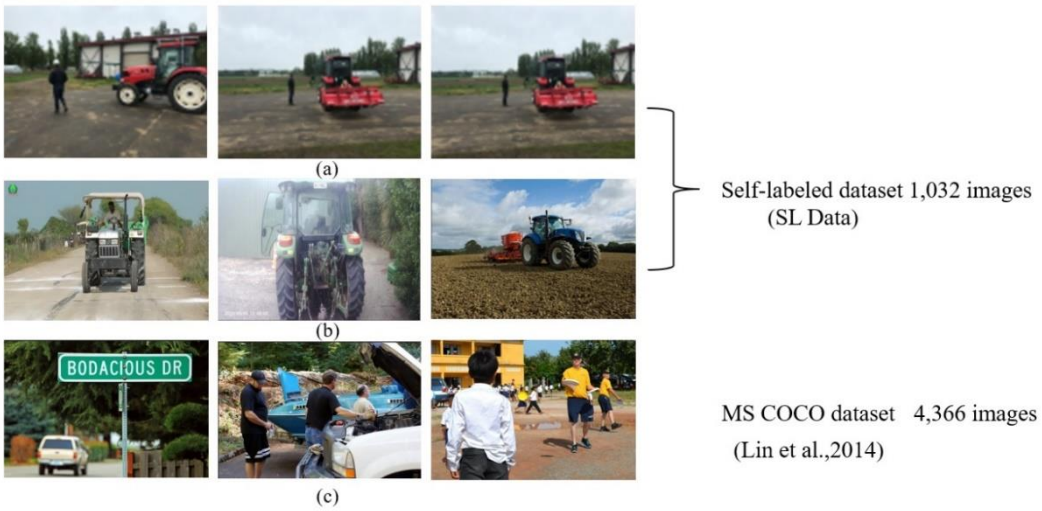


Figure 3.2-2 The SL and coco datasets.

The self-labeled datasets were obtained from the video recordings of robot tractors owned by the VeBots Laboratory of Hokkaido University and online tractor images, which consisted of 1430 items labeled with tractors and humans.

Table 11 Comparison of three Data.

Dataset	Data Size	Resolution	Background	Instances per picture
SL-Data (Self-captured)	432	higher	Uniform	Less
SL-Data (Web-scraped)	600	uneven	Diverse	Less
COCO Data	4366	highest	Diverse	More

3.2.2 Annotation

In the early stages of this study, LabelImg (Tzutalin, 2015) was used for dataset annotation. LabelImg is an open-source image annotation tool that allows users to

Figure 3.2-3 The interface of Robflow annotation tool

manually draw bounding boxes on images and label these boxes with categories, thereby creating datasets for object detection model training. In the later stages, some annotated data were supplemented using Roboflow (Roboflow, Inc, 2022) as shown in *Figure 3.2-3*. In the label of datasets, the bounding box has five parameters: (cls, x, y, w, h). The cls is the class of the Bounding box, including tractor class and person class. The x and y are the ratios of the horizontal and vertical coordinates of the center pixel of the Bounding box to the width and height of the image.

The data annotation as shown in *Figure 3.2-4* followed these rules:

(1) Ensure that tractors and humans are annotated as two separate categories. If different tractors and humans are present in the same scene, they should be annotated individually.

(2) Annotate the object if it occupies at least 10% of the total image area.

(3) If an object is obscured, but its edges are visible, it should be annotated, with the entire object encompassed by the bounding box.

(4) If the width of the implement mounted on the tractor is less than 2 times the length or width of the tractor itself, then the implement should also be annotated as part of the tractor.

(5) The bounding box annotations must closely follow the contours of the objects being marked.

(6) Perform two annotations for each image and compare their consistency. If the annotations are similar, choose either as the final annotation. If there is a significant

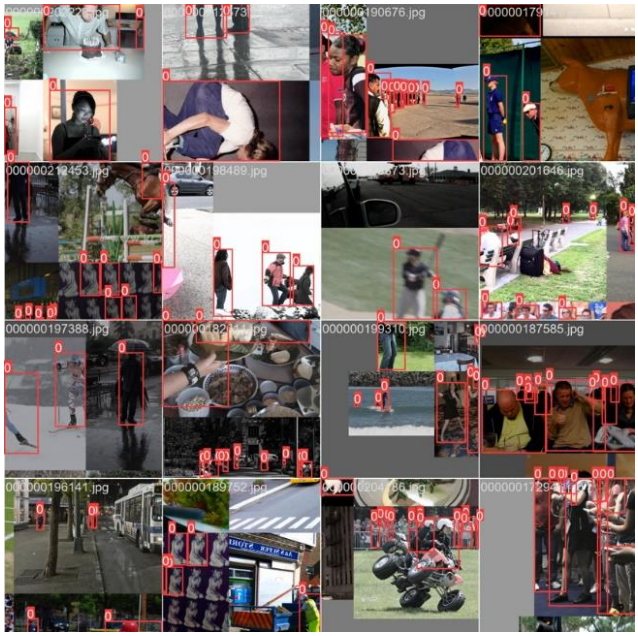


Figure 3.2-4 The labeled human dataset.

difference, re-annotate based on the accuracy of the bounding box, the coverage of the object, and the detail of the segmentation.

3.3 Data augmentation

Due to the significantly lower quantity of self-labeled data compared to the data selected from the COCO dataset, data augmentation was employed to ensure a balance between the number of tractors and humans, thereby preventing model bias towards any one category. Data augmentation is a common technique used to expand the training dataset by generating new training samples from existing data. This is achieved through a series of transformations applied to the original data. The purpose of data augmentation is to increase the diversity of the dataset, thereby helping to reduce model overfitting and enhance its generalization ability to new data (Shorten and Khoshgoftaar, 2019). We enlarged the self-labeled dataset by rotating, inverting, and scaling, etc. As shown in *Figure 3.3-1*. In addition to these traditional data augmentation methods, we used the newer method of mosaic data augmentation (Bochkovskiy et al., 2020), which reduces the reliance on a large batch size by stitching four random images into one image as shown in *Figure 3.3-1 e*; the enhanced image includes the information of all four images. The detection of small targets is usually poorer than the detection of large targets, and stitching four images into one image expands the number of small targets in the dataset, thereby improving the detection performance of the model for small targets. We supplemented the number of self-labeled datasets to 5,400 by using the data augmentation method. In addition to selecting images from the COCO dataset, 234 background images were also incorporated. Background images are those without any annotations, and their

inclusion helps to reduce False Positives (FP). In total, the dataset comprises 10,000 images, which are divided into training, validation, and test sets in a ratio of 7:2:1.

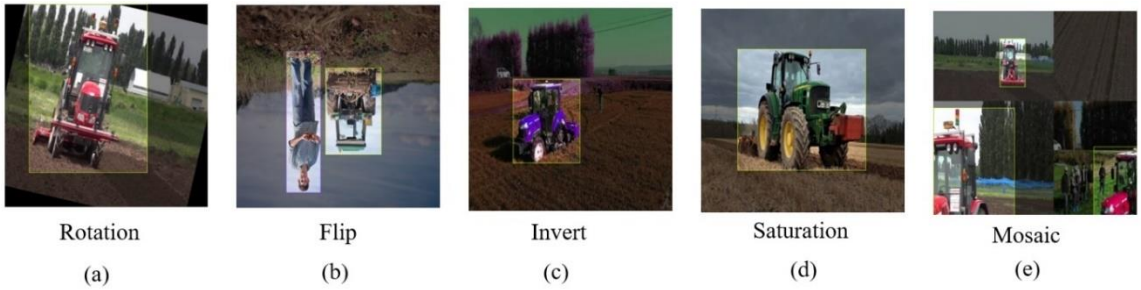


Figure 3.3-1 The data augmentation.

Mosaic Data Augmentation works by combining four different training images into a single composite image. The position, size, and scale of each image on the new canvas can be random. This method provides the model with an image that contains richer background and contextual information, which helps the model learn to recognize objects in various environments. By training the model to process these composite images, it learns to identify partially obscured objects and to recognize objects in complex, cluttered backgrounds, thereby improving its generalization ability and robustness.

Table 12 The Data augmentation parameter

Data Augmentation type	Parameter
Rotation	-15° ~ +15°
Flip	Vertical, Horizontal
Invert	Color invert
Saturation	-25% ~ +25%
Mosaic	4 Pictures

3.4 Training and Optimization

The training environment used for the detector in this study was the Windows 10 operating system, with an Nvidia Geforce RTX 3070 (8G) GPU, a Darknet framework for YOLOv3, a Pytorch framework for YOLOv5 series and variant models, and Faster R-CNN (ResNet) (He et al., 2016).

To address the issues mentioned in section 3.2, we conducted an ablation experiment on data combination. Considering the significant differences between the self-labeled data and the data selected from the COCO dataset, to explore the impact of combining these two data types, models were trained using different quantity combinations and starting from pretrained weights or with randomly initialized parameters as a test. The weights of the randomly initialized models were set to random values before training commenced. This means the model begins learning from scratch, without any prior knowledge. The training parameters were set as follows: the momentum of the learning rate at 0.937, weight decay at 0.001, size (pixels) at 640, batch size at 32, and epochs at 30. The results are as shown in Table 13 and the Precision-Recall Curve of 50/50 combination is as shown in *Figure 3.4-1*:

Table 13 Results of object detection on the test set.

Data combination	mAP (%)	Training method
COCO Data (25) + SL Data (75)	60.0	Pretrained weights
COCO Data (50) + SL Data (50)	**60.3**	Pretrained weights
COCO Data (75) + SL Data (25)	59.8	Pretrained weights
COCO Data (25) + SL Data (75)	1.3	Randomly initialized yaml
COCO Data (50) + SL Data (50)	1.3	Randomly initialized yaml
COCO Data (75) + SL Data (25)	1.1	Randomly initialized yaml

It can be observed that due to insufficient training time and a small number of epochs, models with randomly initialized parameters did not converge. But the results of the three different data combinations did not show significant differences, leading us to conclude that the disparities between the two datasets do not have a noticeable impact on the detection results.

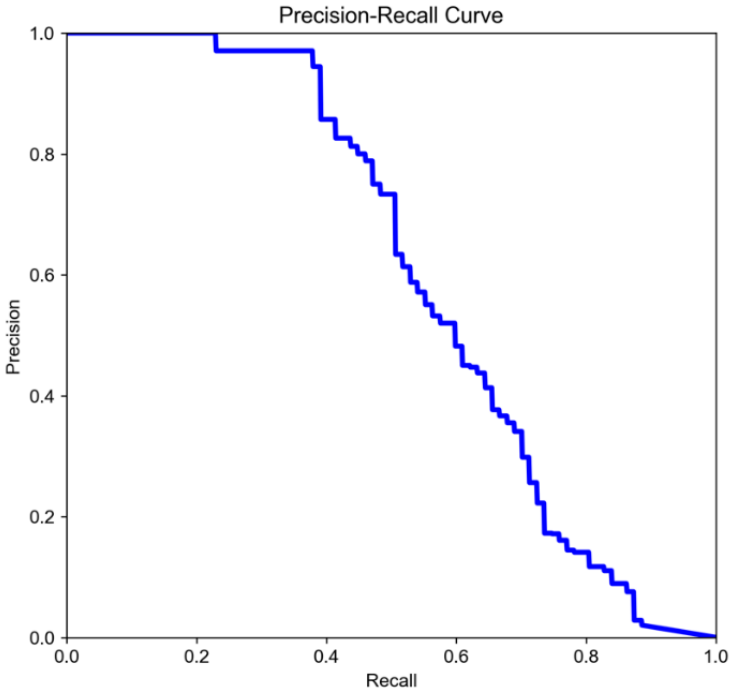


Figure 3.4-1 The Precision-Recall Curve of COCO Data (50) + SL Data (50) combination.

For the detector, we trained three models in order to compare their performance and select the best model. The system initially used YOLOv3 in the Darknet framework as the detection model (Redmon and Farhadi 2018) and trained a Faster R-CNN as a comparison, and it was then updated to YOLOv5s (Jocher et al., 2022) in the PyTorch framework as the detection model. It is worth noting that there are different types of object detection algorithms, such as one-stage algorithms like YOLO and two-stage algorithms like Faster R-CNN. The Faster R-CNN algorithm first generates a set of object proposals and then classifies them, whereas YOLO directly divides the image into grids and predicts the category for each grid, thereby significantly improving the speed of detection. This feature makes YOLO very suitable to be used in practical projects.

The latest version of YOLOv5 features models of five different scales. For our comparison, we chose the three intermediate scales: YOLOv5s, YOLOv5m, and YOLOv5l. These models are based on the same underlying architecture, with the main difference being their density. YOLOv5s has the least density, while YOLOv5l has the greatest. Comparing different versions of the same underlying architecture needs to be directly related to achievable accuracy. That is to say, if a sufficient amount of data is provided for training, denser networks typically yield better results in terms of evaluation metrics, but at the cost of increased inference time.

We also used different backbones to compare the impact of various network layers and configurations on achieved results. In this experiment, the processing speed of images is a crucial metric. Therefore, two popular lightweight backbones were chosen to replace the CSP backbone of the YOLOv5 model for testing. These are MobileNetV3 (Howard et al., 2017) and ShuffleNetV2 (Zhang et al., 2018).

3.5 Results and Discussion

3.5.1 Performance Evaluation of Detectors

Since the primary goal of the remote safety system is to detect obstacles approaching the tractor, we focused on the detection performance of the model for large and medium-sized targets, as well as the accuracy and detection speed of the model. We used the current mainstream mean average precision (mAP) as the evaluation metric of the detection model in this study. The mAP metric is the average value of the AP (average precision) for all categories (person and tractor).

These metrics are calculated as follows:

$$mAP = \frac{1}{c}\sum_{i=0}^{c} AP_i \tag{3-1}$$

$$AP = \int_0^1 Precision\ (Recall)\ dR \tag{3-2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3-3}$$

$$Recall = \frac{TP}{TP+FN} \tag{3-4}$$

where TP represent the number of correctly detected objects (true positives), FP represent the number of falsely detected objects (false positives), and FN represent the number of missed objects (false negatives). $Precision\ (Recall)$ means the precision-recall curve. c is the number of classes, which is two in this study.

The training parameters were set as follows: the momentum of the learning rate at 0.937, weight decay at 0.0005, size (pixels) at 640, batch size at 32, and epochs at 300. Results of different modules are as shown in Table 14.

Table 14 Results of object detection on the 3 different model.

Detector	Backbone	mAP (%)	AP_p (%)	AP_t (%)	*FPS*
YOLOv3	Darknet53	87.4	85.4	89.3	18
YOLOv5s	CSPDarknet	87.3	84.7	89.9	**32**
Faster R-CNN	Resnet50	**89.7**	**88.3**	**91.0**	8

[footnote] AP_p: average accuracy of person class. AP_t: average accuracy of tractor class.

We can observe that Faster R-CNN exhibits the best performance with a mAP of 89.7%, while YOLOv5s and YOLOv3 show similar results, scoring 87.3% and 87.4% respectively. However, in comparison to the other two models, YOLOv5s has a significant advantage in terms of running speed, achieving 32 FPS. YOLOv3 is slower at 18 FPS, and Faster R-CNN only reaches 8 FPS, which does not meet the requirements for real-time operation. The detection results as shown in *Figure 3.5-1*, Precision-Recall Curve of YOLOv5s model as shown in *Figure 3.5-2.*

Figure 3.5-1 The remote safty system detection results.

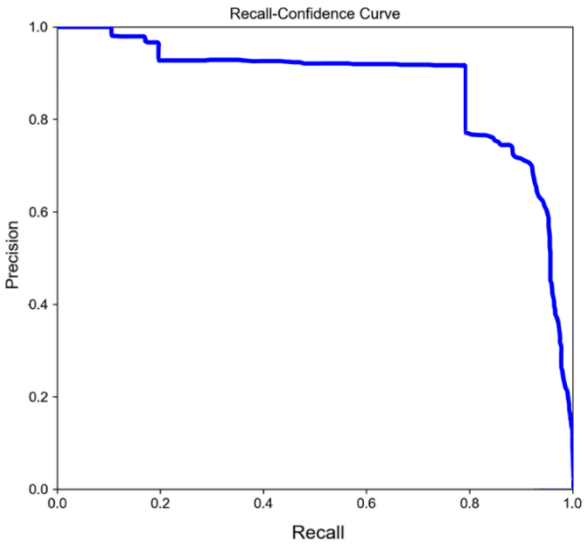


Figure 3.5-2 The Precision-Recall Curve of YOLOv5s model

The test results of YOLO models at different scales as shown in Table 15:

Table 15 Results of object detection on the different YOLOv5 scale.

Detector	Backbone	mAP (%)	AP_p (%)	AP_t (%)	*FPS*
YOLOv5s	CSPDarknet	87.3	84.7	89.9	**32**
YOLOv5m	CSPDarknet	87.5	84.7	90.2	20
YOLOv5l	CSPDarknet	**88.2**	**85.3**	**91.1**	12

[footnote] AP_p: average accuracy of person class. AP_t: average accuracy of tractor class.

We can know that YOLOv5s has the fastest running speed but relatively lower accuracy, while YOLOv5l is the slowest but has the highest accuracy. This aligns with our expectations: models with greater density achieve better detection results, but correspondingly, the increased density also leads to a decrease in processing speed.

The backbone network of YOLOv5s uses several combined modules of Conv, C3, and SPP for feature extraction; Because its network structure is more complex, and the number of parameters is large, redundant information inevitably wastes computational resources. Due to the significant heating of computers caused by running deep learning models for extended periods, it is very important to use lightweight models to ensure safety. We attempted to replace the backbone network of YOLOv5s with two popular lightweight networks mobilenetV3 and ShuffleNetV2 observed the results, as shown in the Table 16.

Table 16 Results of object detection on the different backbone.

Detector	Backbone	mAP (%)	AP_p (%)	AP_t (%)	FPS
YOLOv5s	CSPDarknet	**87.3**	**84.7**	**89.9**	32
MobileNet-YOLO	MobileNetV3Small	83.3	81.5	85.1	60
ShuffleNet-YOLO	ShuffleNetV2	82.9	81.9	83.9	**72**

[footnote] AP_p: average accuracy of person class. AP_t: average accuracy of tractor class.

From the data in the table, we can see that MobileNetV3 and ShuffleNetV2 have certain advantages in terms of computing power, memory consumption, and latency, but compared to the original network, there is a significant decrease in detection accuracy. It is still necessary to explore the effect of the decrease in accuracy on actual detection tasks.

3.5.2 Discussion

From the data combination ablation study, we found that the different quantity combinations of the two types do not have a significant impact on the results. One possible reason is that the YOLOv5 model itself may have strong robustness and generalization capabilities, allowing it to maintain stable performance across a variety of data combinations. Another possible reason is that both datasets contain sufficient information to solve the task, so different data combinations might not result in significant performance

differences. In summary, we believe that mixing these two types of data for model training is feasible.

In the comparison of the three models, Faster R-CNN performed the best in terms of accuracy, but its low inference speed makes it unsuitable for real-time processing. The performance gap between YOLOv3 and YOLOv5s is relatively small, but YOLOv5s is much faster in terms of inference speed. One reason is that YOLOv5s is a lightweight version of YOLO, having a smaller density compared to the standard YOLOv3. Another reason is that YOLOv3, an earlier version we used, is deployed on the Darknet framework, whereas YOLOv5s utilizes the PyTorch framework with many optimized integrated libraries, making it faster in matrix operations. Compared to the YOLOv3 model, YOLOv5 uses the Cross Stage Partial Network (CSPNet) as its backbone network, which enhances the feature extraction capabilities based on the original model. YOLOv5 also optimizes the bounding box loss, classification loss, and object confidence loss, contributing to the improved model performance.

In the comparison of the three scales of YOLOv5, the primary difference between the various versions of YOLOv5 lies in the size and complexity of the model. Larger models, such as YOLOv5l, typically have more convolutional layers, which means the network can learn more complex features, but this also increases the computational burden. These models of different scales offer a trade-off between speed and accuracy. Generally, smaller models like YOLOv5s run faster, but their accuracy might be lower. Larger models, such as YOLOv5l, perform better in terms of accuracy, but are slower. Models of different scales are suitable for different application scenarios. YOLOv5s is the fastest but least accurate version among the YOLOv5 models. In the present study, we aimed to maximize

the model's runtime speed while satisfying minimum accuracy requirements, and we thus chose YOLOv5s as our primary test model.

In the comparison of the three different backbone YOLOv5, we found that YOLOv5 models with two types of lightweight backbones have faster inference speeds compared to the original YOLO v5, but there is a significant decrease in accuracy. This indicates that the effectiveness of the network is influenced not only by the dataset but also by certain types of layers. Whether these two lightweight backbones meet the requirements of practical detection needs further discussion.

We selected 500 images for testing, which included target tractors or humans, all within the tractor's stopping or deceleration zone. Using the YOLOv5s model, we achieved a detection rate of 0.98. With the MobileNet-YOLO model, the detection rate was 0.90, and the same was true for the ShuffleNet-YOLO model. We believe that the two lightweight models lead to a relatively high drop in accuracy, which does not meet the objectives for safe use and still requires improvement. In practice, the speed of the analysis is an important indicator, and small-scale targets generally cause missed detection. Although a mAP value of 87.3% may not be an exceptionally high number, we observe from the results in Chapter 6 that this model achieved a 100% accuracy rate in multiple field experiments. This is because the test dataset contains many small-scale targets, and this model demonstrates a very high detection rate for targets of varying sizes within the robot's forward region of interest (ROI) during real-world usage.

In conclusion, a comparison of the three models revealed that even though the Faster RCNN model has higher accuracy, its lower frame rate may cause issues in practical applications. The accuracy of YOLOv5s and YOLOv3 is slightly lower, but while meeting the real-world usage requirements YOLOv5s has a faster speed, making it perform better in real-world scenarios. A comparison of three scales that YOLOv5s has a higher inference speed than other scales and an accuracy that meets requirements. The significant decrease in accuracy of the two lightweight backbones compared to the original model could pose problems in practical use, which is unacceptable in safety-first projects. Therefore, when considering factors such as detection performance, accuracy, and speed, we believe that the YOLOv5s model holds an advantage over the others and is capable of serving as the detection head for robots in real-world applications as shown in *Figure 3.5-3*OBSTACLE POSITIONING METHOD OF THE REMOTE SAFTY SYSTEM

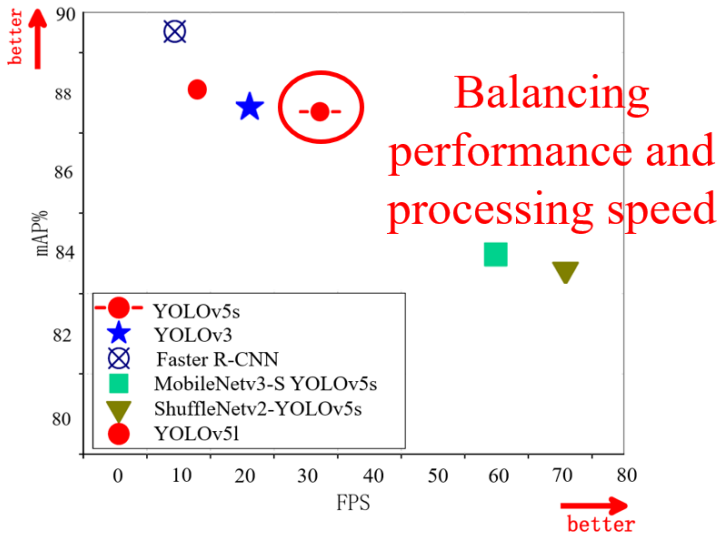


Figure 3.5-3 Comparison of performance and speed among several different models.

CHAPTER 4. OBSTACLE POSITIONING METHOD OF THE REMOTE SAFTY SYSTEM

4.1 Introduction

The objective of this chapter is to utilize the YOLOv5s detector developed in the previous chapter, based on deep learning models, to accomplish the localization of obstacles appearing in front of the robot tractor. The primary reason for using a monocular vision system in this study is its cost-effectiveness. A monocular vision system typically requires only one camera, significantly reducing hardware costs. This is beneficial for large-scale deployment and for farmers adopting these new technologies. Compared to multi-camera or stereoscopic vision systems, a monocular system has a simpler structure. This not only means that installation and configuration are easier, but also that maintenance and troubleshooting are more straightforward.

Additionally, a monocular vision system usually processes less data than multi-camera systems, hence requiring fewer computational resources. The approach adopted in this study involves sending image data to a remote server for processing and then returning control commands. In the typically poor network environments of farmlands, minimizing the volume of data transmission to ensure smooth operation is also crucial.

Finally, the adaptability of monocular vision is quite strong. It can adjust to a variety of lighting and environmental conditions, which is particularly important for outdoor applications or environments that are subject to frequent changes.

Monocular positioning faces several major challenges:

(1) In a monocular vision system, due to the lack of depth information, it is difficult to determine the actual size and distance of an object. The size of an object in an image may vary depending on its distance.

(2) Although monocular systems are generally simpler and less expensive than multi-camera systems, extracting three-dimensional information from a single image may require complex algorithms and substantial computational resources.

(3) During a long-distance positioning, a monocular system may accumulate errors, leading to inaccuracies in positioning.

(4) In dynamic and constantly changing environments, such as outdoors or in crowds, maintaining stable and accurate positioning is a challenge.

How to solve these problems and enhance the positioning accuracy of a monocular camera is the focus of our research.

4.2 System setup and Camera calibration

We used a monocular camera to collect images, as shown in *Figure 4.2-1*. The camera was mounted on the pan-tilt unit at the top of the robot tractor at 2.6 m above the ground, with a 30° angle. The images captured by the camera in real-time, with a resolution of 1280 × 720 pixels, are transmitted to the NUC inside the tractor. Subsequently, the images will be transmitted over the internet to a remote high-performance processor for further processing.

Figure 4.2-1 Camera position and angle.

The camera was mounted on the pan-tilt unit at the top of the robot tractor at 2.6 m above the ground, with a 30° angle.

In this study, a monocular camera is used to measure the distance to obstacles in front of the robot tractor. To establish a camera imaging geometry model and correct lens distortion, it is necessary to calibrate the camera.

The perspective projection model for cameras can be expressed as follows:

$$Z_C \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K[R|T] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{4-1}$$

where $P_i = [X_w \ Y_w \ Z_w \ 1]$T is the homogenous world point, $p_i = [x \ y \ 1]$T is the corresponding homogeneous image point, K is the matrix of the intrinsic camera

parameters, Z_C is a scale factor for the image point, and $[R|T]$ is the matrix of the extrinsic parameters. The matrix of intrinsic and extrinsic camera parameters is:

$$K = \begin{bmatrix} f_x & & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix} \tag{4-2}$$

where f_x and f_y represent the focal length in terms of pixels, and c_x and c_y represent the principal point of the camera.

$$[R|T] = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4-3}$$

where R is the rotation matrix and T is the translation matrix.

According to Zhang's method (Zhang 2000), we can calibrate the camera by using a checkerboard to obtain the intrinsic K, extrinsic $[R|T]$, and distortion coefficients. A chessboard, composed of alternating black and white squares, is used as a calibration tool for camera calibration (mapping objects from the real world to 2D images). Since a two-dimensional object lacks some information compared to a three-dimensional object, we capture images by changing the orientation of the chessboard multiple times to obtain coordinate information. As shown in the *Figure 4.2-2* and *Figure 4.2-3*, this is the same chessboard photographed by the camera from different angles.

Figure 4.2-2 Multiple shots of a chessboard grid at different positions for camera calibration.

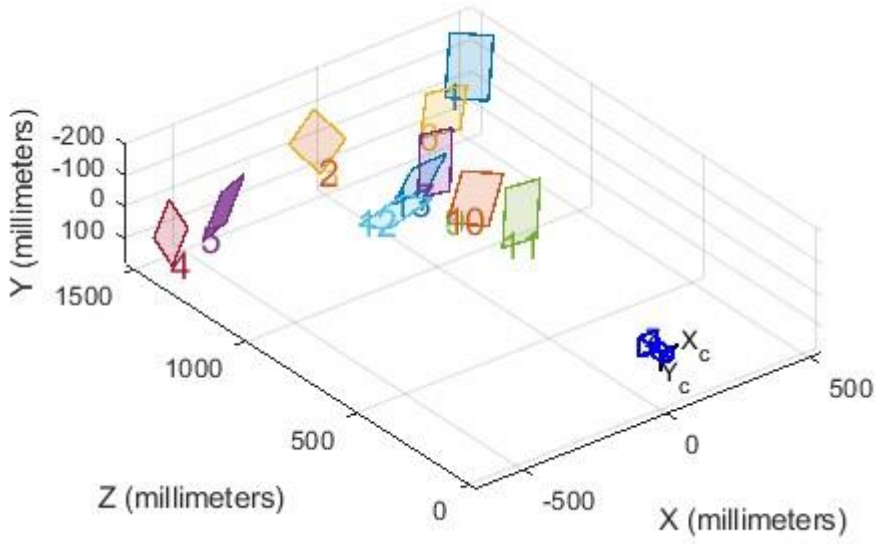


Figure 4.2-3 The chessboard grid's position relative to the camera in space.

4.3 Camera Positioning

The conversion equation for the pixel coordinates to the tractor's world coordinates is:

$$Z_C \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} [R|T] \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} \tag{4-4}$$

Equation (4-4) can be extended as:

$$\begin{cases} Z_C * x = X_w * (f_x * R_{11} + c_x * R_{31}) + X_w * (f_x * R_{12} + \\ c_x * R_{32}) + Z_w * (f_x * R_{13} + c_x * R_{33}) + f_x * T_1 + c_x * T_3 \\ Z_C * y = X_w * \left(f_y * R_{21} + c_y * R_{31}\right) + Y_w * (f_y * R_{22} + \\ c_y * R_{32}) + Z_w * (f_y * R_{23} + c_y * R_{33}) + f_y * T_2 + c_y * T_3 \\ Z_C = X_w * R_{31} + Y_w * R_{32} + Z_w * R_{33} + T_3 \end{cases} \tag{4-5}$$

According to the perspective-n-point (PNP) method (Fischler and Bolles 1981), we only need to know the world coordinates and the pixel coordinates of a feature point, the intrinsic camera parameters, and the camera distortion coefficients to find the world coordinates of the camera. We set four control points (C1, C2, C3, and C4) on the ground as shown in *Figure 4.3-1*. The control points' coordinates and the distance between the camera and each control points are obtained from the real-time kinematics (RTK) global positioning system (GPS). The pixel coordinates are obtained by imaging the control point in the tractor camera, and the world coordinates of the camera relative to the plane XY can be restored.

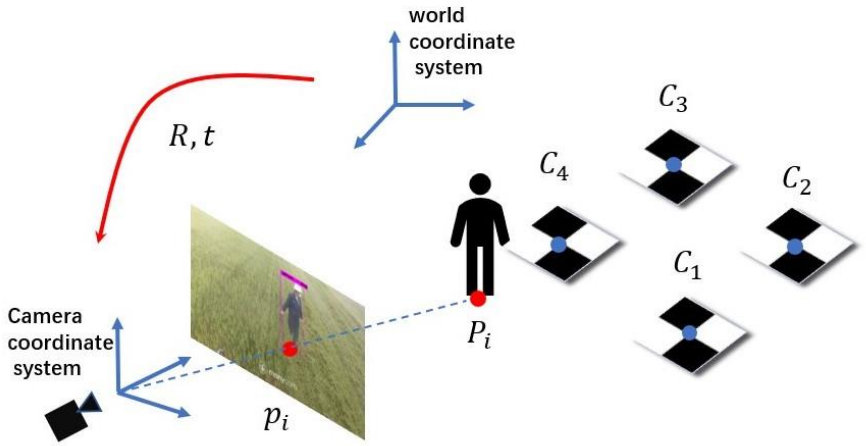


Figure 4.3-1 The perspective-n-point (PNP) method.

The center point of the camera projecting to XY, where these four control points are located, is the origin of the world coordinate system; the tractor's forward direction is the X-axis positive direction, the Z-axis directs to the center of the camera lens, and the Y-axis can be derived by the right-hand rule.

Since the points that we want to measure must be located on the plane of contact with the ground, we can assume that Z_w = 0 and substitute it into Eq. (4-5), which can be simplified to:

$$\begin{cases} Z_C * x = X_w * (f_x * R_{11} + c_x * R_{31}) + X_w * (f_x * R_{12} + \\ \qquad c_x * R_{32}) + f_x * T_1 + c_x * T_3 \\ Z_C * y = X_w * \left(f_y * R_{21} + c_y * R_{31}\right) + Y_w * (f_y * R_{22} + \\ \qquad c_y * R_{32}) + +f_y * T_2 + c_y * T_3 \\ \qquad Z_C = X_w * R_{31} + Y_w * R_{32} + T_3 \end{cases} \tag{4-6}$$

The ternary equation with the three unknowns X_w, Y_w, and Z_C is obtained and can be solved. The obtained solution is the world coordinates P_i = [X_i Y_i 0 1] T of the pixel point pi and the distance Z_{ci} between the camera and p_i. According to the world coordinates and the pre-measured distance between the control point for the tractor, we can calculate the distance D_i of P_i for the tractor. It should be noted that in this study we used the component D_{xi} of D_i in the X-axis direction as the actual distance. Through this method, we can map all points on the image onto a plane as shown in *Figure 4.3-2*, thereby calculating their distances.

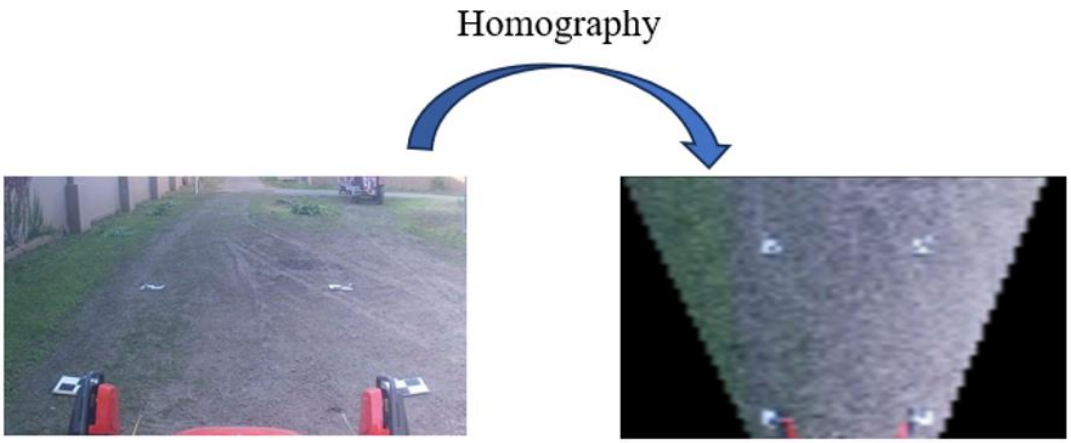


Figure 4.3-2 Homograph of the monocular image.

4.4 2D-LiDAR positioning

In addition to visual positioning, our system also utilizes 2D-LiDAR positioning as a local safety device. Compared to cameras, LiDAR can perform more accurate distance measurements, is less susceptible to factors like vibration and lighting conditions, and does not rely on data learning. It can detect the distance to any obstacle that reflects laser beams. LiDAR employs the time of flight (TOF) measurement method. A laser is emitted

from the transmitter of the rangefinder, which illuminates the target and reflects to the receiver. The distance to the target d is calculated by measuring the time t that it takes for the laser to travel to the target and back, using the speed of light c.

The calculation formula is as follows:

$$d = c * \frac{t}{2} \tag{4-7}$$

The Hokuyo UTM-30LX LiDAR that we used has a maximum scanning range of 270°, a maximum distance of 30 m, and a scanning time of 25 ms.

4.5 Determination of Safety Zones

In this study, the base tractor we used is the Kubota MR1000A crawler-type tractor with a total length of 4,505 mm, a width of 1,980 mm, and a total height of 2,725 mm. The commonly used operating width for rotary operations is 2,600 mm or 2,800 mm. For safety purposes, when operated remotely, the maximum specified speed can be set to 10 km/h, and when a specified rotation count of 2,600 mm is selected, the actual maximum vehicle speed is 8.8 km/h. In the deceleration zone, if a target is detected and the speed exceeds 3.6 km/h, the robot will decelerate to 3.6 km/h. This section requires confirmation of the ranges for the visual stopping zone, the visual deceleration zone, and the LiDAR stopping zone.

Referring to the performance evaluation criteria for reducing pedestrian collision injuries set by Japan's Ministry of Land, Infrastructure, Transport and Tourism (MLIT), it is necessary to avoid collisions with pedestrians crossing in front when they are moving at a speed of 5 km/h. The following four points were therefore considered when

specifying the range for each zone in this study:

(1) A visual stopping zone serves as the first layer of the safety zone, and when the robot is traveling at its maximum speed (8.8 km/h), braking based solely on visual detection should ensure the avoidance of collisions with obstacles within the visual stopping zone.

(2) A LiDAR stopping zone serves as the second layer of the safety zone, and when the robot is traveling at its maximum speed (8.8 km/h), braking based solely on LiDAR detection should ensure the avoidance of collisions with pedestrians crossing at a speed of 5 km/h in the vicinity of the robot.

(3) To ensure the operational efficiency of the robot tractor, braking based solely on visual detection should minimize the impact of objects that are not on the operational path.

(4) The robot tractor can decelerate to 3.6 km/h after passing over a deceleration zone at its maximum speed (8.8 km/h).

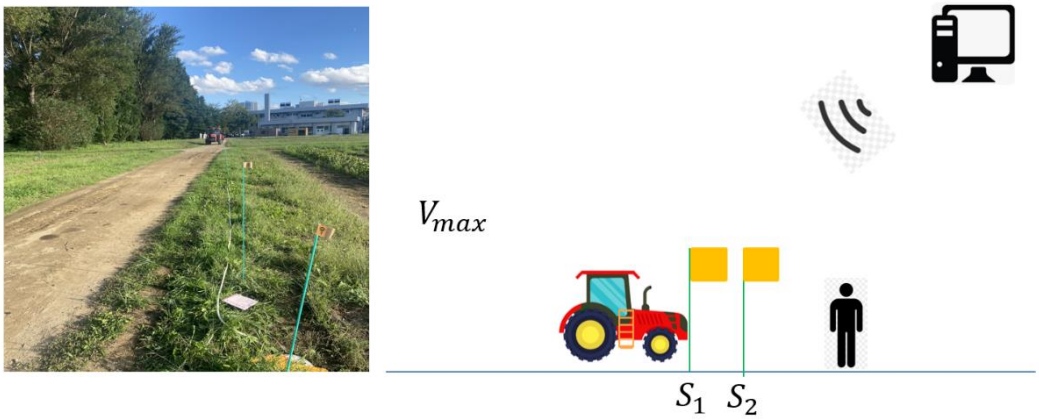


Figure 4.5-1 Experiment for measuring the braking time and braking distance.

We assumed an 8-m zone in front of the tractor as the visual stopping safety zone, and beyond 8 m was the safe zone. A test operator stood at the boundary of the 8-m zone judged by the system and recorded the tractor's stopping position S_2 relative to the operator's position. Subsequently, the robot tractor was remotely controlled to travel from a distant point towards the operator at its maximum speed of 8.8 km/h. After automatic stopping based on visual positioning, the position S_1 where the tractor came to a halt was recorded.

The breaking distance S_d is defined as:

$$S_d = S_1 - S_2 \tag{4-8}$$

The braking time, defined as time td from the vehicle's horn blast (automatically activated when an obstacle is detected in the stopping zone) to the time at which the vehicle comes to a complete stop was measured in each experimental group three times, and the results are shown in Table 17.

Table 17 The braking time and distance for each braking method at maximum speed.

Breaking type	$\overline{t_d}$ (s)	$\overline{S_d}$ (m)
Visual breaking	**3.5**	**7.4**
LiDAR breaking	2.1	3.7

[footnote] $\overline{t_d}$: the average braking time; $\overline{S_d}$: the average braking distance.

According to principle (1) outlined above, the visual stopping zone should be greater than the braking distance at maximum speed, i.e., L_1 > 7.4 m. We thus used 8 m as the value of L_1. According to principle (3), visual detection should minimize the impact of objects that are not on the operational path. The commonly used operating width for rotary operations is 2.6 m or 2.8 m (i.e., 2.8 m for W). We set the value of W as 3.2 m. The visual stopping zone was thus an 8 m × 3.2 m rectangular area in front of the tractor as shown in *Figure 2.1-5*. According to principle (4), after setting the stopping safety zone as an 8 m × 3.2 m rectangular area, we determined that at 8 m in front of the robot tractor, a deceleration distance of 3.6 m is required to reduce the speed from 8.8 km/h to 3.6 km/h. We thus set the L_2 value as 4 m.

According to principle (2), as the final safety zone, braking based solely on LiDAR detection should ensure the avoidance of collisions with pedestrians crossing at a speed of 5 km/h in the vicinity of the robot. Considering (i) the robot tractor as a rectangular rigid body with a total length of 4,505 mm and a width of 1,980 mm and (ii) the LiDAR scanning range as a sector (adjustable-angle) specified as a semi-circle of 180° in front of the robot with a scanning radius *R*, *Figure 4.5-2* depicts the position relationship between the tractor and pedestrians crossing at 5 km/h.

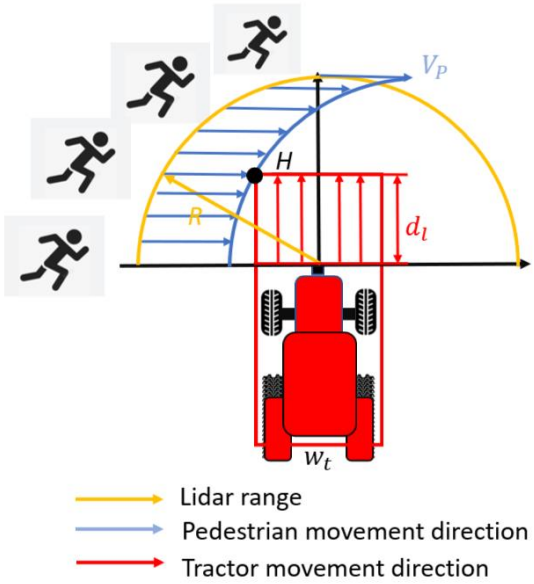


Figure 4.5-2 The position relationship between the tractor and pedestrians crossing at 5 km/h. Assuming that the initial position of the tractor is the origin of the coordinate system with the LiDAR position. At the coordinates of point $H(-\frac{w_t}{2}, \bar{d}_l)$*, a pedestrian who is crossing at 5 km/h* (V_P) *would collide with the tractor.*

Based on the Pythagorean theorem, the LiDAR scanning radius R is thus calculated as:

$$R > \sqrt[2]{(-\frac{w_t}{2} - V_P * \bar{t}_l)^2 + (\bar{d}_l)^2} \tag{4-9}$$

where $\bar{d}_l$ is the average LiDAR detection braking distance, $\bar{t}_l$ is the average LiDAR detection braking time, w_t is the tractor width. We calculated that R > 5.47 m and set the value of R as 5.5 m as shown in *Figure 2.1-5*.

4.6 Results and Discussion

By using the PnP (Perspective-n-Point) method in conjunction with bounding box and considering that the bottom line of a bounding box surrounding a person is always at the person's feet, we can assume that the points on this line are on a plane determined by four control points on the ground. Therefore, by assuming $Z_w = 0$, compensate for the missing Z-axis information in monocular vision, enabling us to position the target. The error between the predicted values and the actual values for measurements taken at every 0.5 meters within a range of 15 meters, the result as shown in *Figure 4.6-1*.

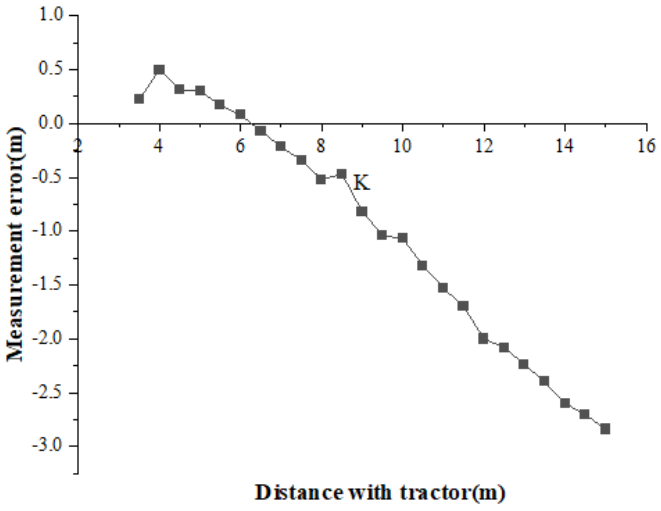


Figure 4.6-1 Measurement error in the X-direction at the row data.

Each group of experiments was conducted five times, and we used the average of the results.

The average relative error of the 5.6 % at 15 m; the maximum error occurs at the farthest point of 15 m, which is 2.84 m. The maximum relative error remains at around 19 % which is too large to be useable. This is due to the inherent errors of the monocular

camera itself, as well as a pixel error associated with the use of the bounding box method. We will discuss the methods to correct this error in the next section.

In this chapter, we also established a method for determining the safety zone for different vehicles in remote safety systems. Through these two methods, we can calculate the distance between monitored vehicles and detected obstacle targets and assess the vehicle's safety status based on the safety zone, thereby ensuring the safe operation of robot vehicles.

CHAPTER 5. POSITIONING RESULTS CORRECTION

5.1 Introduction

The objective of this chapter is to refine the positioning results of the detector developed in Chapter 3 and the positioning method proposed in Chapter 4. From the results of the previous chapter, we obtained a positioning error with an average relative error of 5.6% at 15 meters. The maximum error occurs at the farthest point of 15 meters, which is 2.84 meters. The maximum relative error remains at around 19%. This is not only due to the accuracy disadvantage of monocular vision compared to other high-precision sensors, but also because when using a bounding box to mark a person, the bottom line of the bounding box does not precisely land at the person's feet. Instead, it experiences a pixel offset that varies with the person's position. This results in a minor positioning error when the subject is close to the camera. However, as the distance between the person and the camera increases, this error becomes significant due to the imaging characteristics of the monocular camera. In this chapter, we propose a method to analyze and correct this pixel offset using statistical techniques, thereby improving positioning accuracy Positioning pixel error by using monocular camera.

The bounding box of the object in an image is detected using YOLOv5s. If an object is detected, we can obtain the pixel coordinates of each bounding box. For the location estimation, we used a single pixel $p_i(x_i, y_i)$ which was located at the center of that bounding box's bottom edge to represent each detection. The actual pixel position where the target is located will have deviation ε_p from the pixel position of p_i as shown in *Figure 5.1-1*. Due to the imaging principle of a monocular camera, this

deviation may cause more significant positioning errors at greater distances. The deviation of the object in the same position may be different in different frames. This phenomenon exists in many detection algorithms that use the bounding box to mark objects.

(a) (b)

Figure 5.1-1 Deviation of the bounding box and ground truth.

(a) Deviation in the close range. (b) Deviation in the long range. The purple box is the predicted box of the YOLO network, and the red line is the ground truth of the position where the obstacle is standing. We selected the point at the center of the line connecting both feet as the location on the human body to draw the red line.

5.2 Positioning result correction methods for human targets

5.2.1 Q-Q plot

If the ground truth has a pixel coordinate in the Y-direction of y_g, we define ε_p as follows:

$$\varepsilon_p = y_i - y_g \tag{5-1}$$

To investigate the probability distribution of this deviation in the YOLOv5s algorithm, we counted the deviation of the detected YOLOv5s predicting the bounding box of the target from the actual value in 1-m increments within a range of 15 m as shown in *Figure 5.2-1*. All of the actual values were marked manually in a subjective manner. Since the markers beyond the range of 15 m will produce significant errors due to small-pixel-value floating, and the objects that are far away from the tractor are not part of this study object of interest, only the distribution within the 15-m range was thus considered. A total of 158 ground truths were labeled, and to compare the differences in ε_p between each distance range, we performed a t-test (Daniel et al., 2018) for each adjacent distance range. The t-test requires that the data obey a normal distribution, and we made a Q-Q plot (quantile-quantile plot) (Chambers et al., 2018) for each interval as shown in *Figure 5.2-2*.

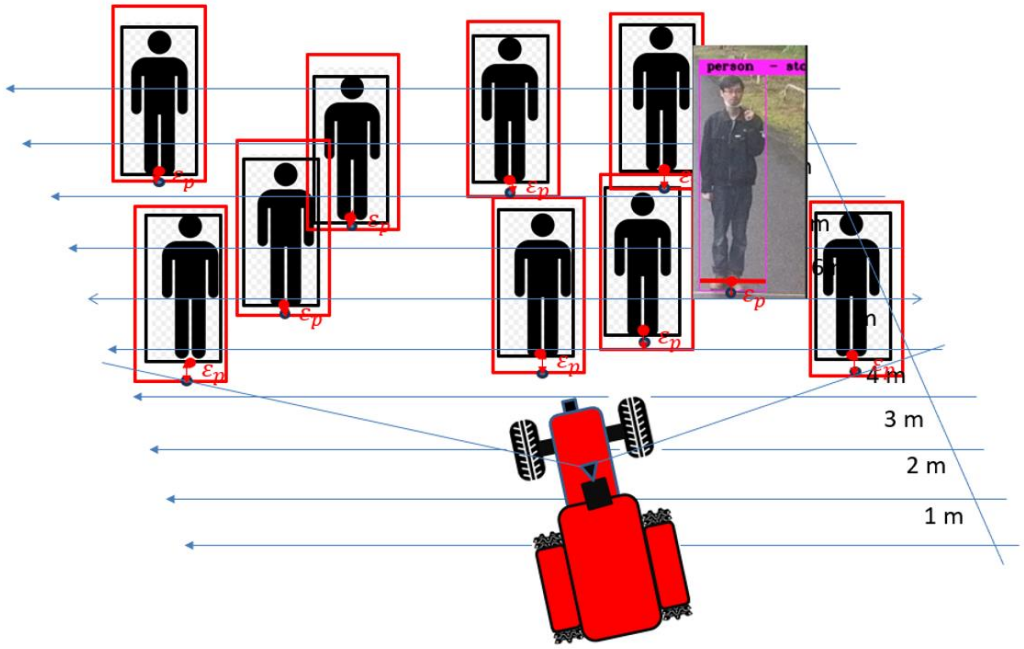


Figure 5.2-1 Experiment for counted the deviation of the detected YOLOv5s predicting the bounding box

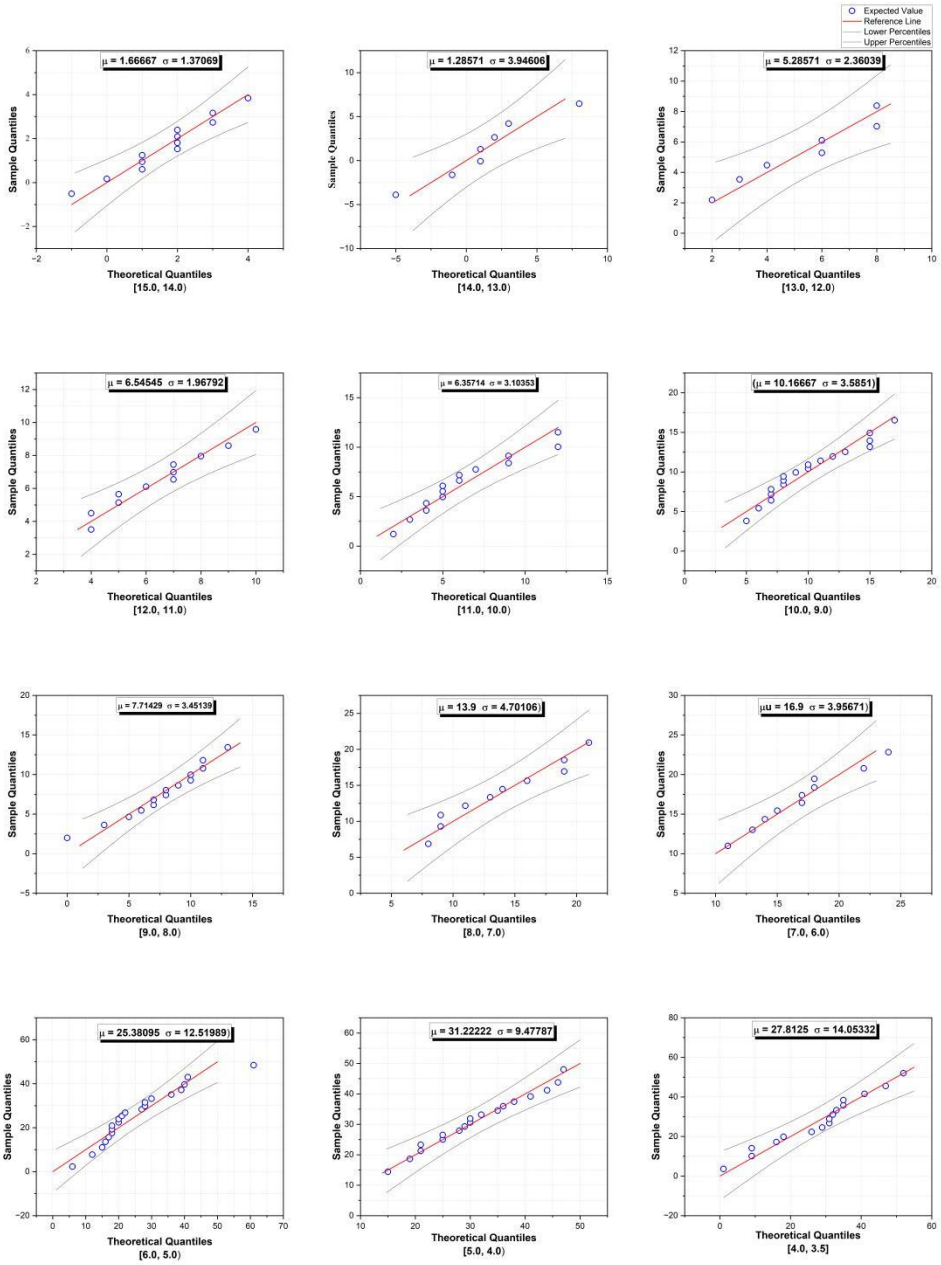


Figure 5.2-2 Q-Q plot for each interval of the 15-m range.

The points will fall on the 45° reference line if the data in each range are normally distributed.

5.2.2 T-test

From the Q-Q plot, we can find that the probability distributions of all deviations on the 12 intervals obey a normal distribution. For each interval with the adjacent intervals, we applied a two-sample t-test; $\bar{x}$ and σ represent the sample mean and standard deviation of ε in each range, respectively. The null hypothesis is that two populations have an equal mean, and the alternative hypothesis is that the two means are not equal:

$$H_0: \mu_1 = \mu_2 \; H_1: \mu_1 \neq \mu_2 \tag{5-2}$$

For equal variance is assumed: In this case the test statistic t:

$$t = \frac{(\overline{x}_1 - \overline{x}_2)}{\sigma_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{5-3}$$

where $\overline{x}_1$ and $\overline{x}_2$ are sample means, n_1 and n_2 are sample sizes, ${\sigma_1}^2$ and ${\sigma_2}^2$ are sample variances and pooled variance σ_p is used:

$$\sigma_p = \sqrt{\frac{(n_1 - 1){\sigma_1}^2 + (n_2 - 1){\sigma_2}^2}{n_1 + n_2 - 2}} \tag{5-4}$$

Degrees of freedom (DF) v:

$$v = n_1 + n_2 - 2 \tag{5-5}$$

For equal variance is not assumed: In this case the usual two sample t-statistic no longer has a t-distribution and approximate test statistic, t' is used:

$$t' = \frac{(\overline{x}_1 - \overline{x}_2)}{\sqrt{\frac{{\sigma_1}^2}{n_1} + \frac{{\sigma_2}^2}{n_2}}} \tag{5-6}$$

The t-distribution with v' (DF) is used to approximate the distribution of t' where:

$$v' = \frac{\left({\sigma_1}^2/n_1 + {\sigma_2}^2/n_2\right)^2}{\frac{({\sigma_1}^2/n_1)^2}{n_1-1} + \frac{({\sigma_2}^2/n_2)^2}{n_2-1}} \quad (5\text{-}7)$$

If a p-value reported from a t-test is < 0.05, we considered the result statistically significant, and if a p-value was > 0.05, the result was considered insignificant. As shown in Table 5-1, the p-values of the adjacent intervals at 13 m, 10 m, 8 m, and 6 m are all < 0.05, and we thus consider that there were significant differences in the probability distributions of the regional deviation on both sides of the above ranges. The merged intervals as shown in *Figure 5.2-3* within the 15-m range can be found in Table 18.

Figure 5.2-3 Merged range intervals with similar probability distributions.

The merging of the range intervals with similar probability distributions produced five intervals.

Table 18 Probability distribution for εp in each interval of the 15-m range.

Interval (m)	n	$\bar{x}$	σ	t'	v'	P
[15.0, 14.0)	12	1.7	1.37	0.25	6.86	0.812
[14.0, 13.0)	7	1.3	3.90	-2.30	9.81	**0.045**
[13.0, 12.0)	7	5.3	2.36	-1.18	11.17	0.264
[12.0, 11.0)	11	6.55	1.97	0.18	22.16	0.855
[11.0, 10.0)	14	6.36	3.10	-3.22	29.60	**0.003**
[10.0, 9.0)	18	10.2	3.59	1.96	28.58	0.060
[9.0, 8.0)	14	7.71	3.45	-3.54	15.66	**0.003**
[8.0, 7.0)	10	13.9	4.70	-1.54	17.50	0.141
[7.0, 6.0)	10	16.9	3.96	-2.82	26.66	**0.009**
[6.0, 5.0)	21	25.4	12.52	-1.66	36.49	0.106
[5.0, 4.0)	18	31.2	9.48	0.82	25.85	0.420
[4.0, 3.5]	16	27.8	14.05	N/A	N/A	N/A

[footnote] n: the number of detections; t': the t-value with reference to the next adjacent interval; v':the degrees of freedom; P: the p-value with reference to the next adjacent interval. Only the data assuming unequal variances is listed, and the differences between the two hypotheses are consistent based on the calculations.

Table 19 Probability distribution for εp in the merged intervals of the 15-m range.

Interval (m)	n	$\overline{x}$	σ
[15.0, 13.0)	19	1.53	2.52
[13.0, 10.0)	32	6.19	2.57
[10.0, 8.0)	32	9.09	3.68
[8.0, 6.0)	20	15.4	4.50
[6.0, 3.5)	55	28	12.13

[footnote] n: the number of detections; $\overline{x}$: the mean average; σ: standard deviation.

5.2.3 Positioning result correction

After the statistical analyses, we observed that in most cases, εp is positive. This is because, at this camera angle, the size of the bounding box assigned by YOLO to the detected person is often larger than the person's actual size, and this value increases as the distance between the camera and the person decreases. For the points to be measured in the pixel ranges of these five intervals as shown in Table 19, we apply the pixel coordinate correction in different intervals, assuming that the pixel coordinates of the YOLO predicted the point of an object falling in interval i is $p_i(x_i, y_i)$. The distance is then calculated using the pixel points representing the object as $p_c(x_i, y_i - \overline{\varepsilon}_i)$, where $\overline{\varepsilon}_i$ is obtained by calculating the regression equation of sample deviation and y_i in distance interval i as shown in *Figure 5.2-4*.

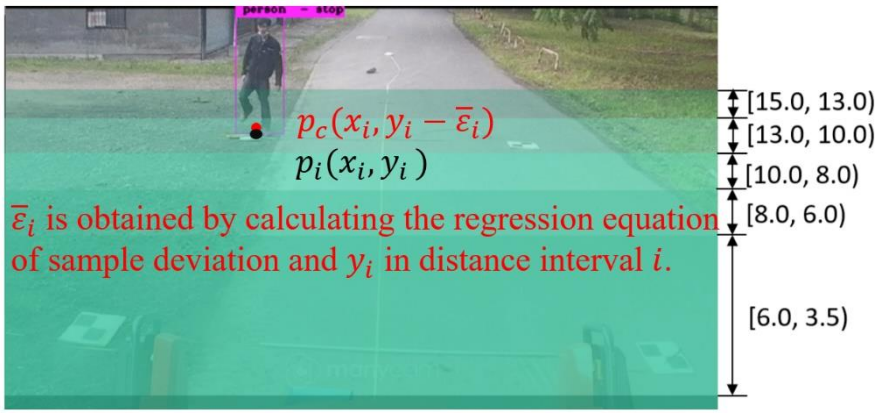


Figure 5.2-4 Positioning result correction.

Using the pixel point p_c instead of p_i for calculations.

5.3 Positioning result correction methods for tractor targets

5.3.1 Keypoint detection

Regarding the positioning correction of tractor targets, due to the significantly larger size of tractors compared to humans, using a single pixel to represent the tractor's position would result in significant errors in tractor positioning.

The method for keypoint detection adopts the approach of capturing the output results of tractor instances from the YOLOv5 network and re-inputting the cropped images into YOLO POSE framework for keypoint detection (Maji et al., 2022). The training images consist of a dataset of 100 tractor keypoints. For each anchor, 39 elements associated with 15 tractor keypoints {wheel 1 ~ 4 top, wheel 1 ~4 bottom, 2 front headlights， Front window top 2 corner, rear window top 2 corner, center bottom of the bumper} The target bounding box is determined by 6 elements: (C_x, C_y, W, H, box_{conf}, $class_{conf}$). C_x and C_y represent the horizontal and vertical coordinates of the center point of the bounding box, W and H represent the width and height of the

bounding box, respectively; box_{conf} and $class_{conf}$ represent the bounding box confidence and the predicted class confidence.

5.3.2 Results and discussion

The experiment adopts the official MS COCO-provided validation standard based on object key point similarity, known as L_{oks} (object keypoint similarity), to compute the average accuracy. L_{oks} for the key point type *i* is given as

$$L_{oks} = \frac{\sum_i \left[exp\left[-\frac{{d_i}^2}{2s^2 k^2 i} \right]_{\delta(v_i>0)} \right]}{\sum_i \delta(v_i>0)} \tag{5-8}$$

where d_{i} is the Euclidean distance between the ground truth and predicted keypoint *i*; L_{oks} is the constant for keypoint *I*, *s* is the scale of the ground truth object; s^2 hence becomes the object's segmented area. d^2 is the squared Euclidean distance between the detected keypoint location and the ground truth keypoint location. *k* is a decay constant used to control the decay of keypoint category *i*, δ is the impulse function. $v_{\text{i}}$is the ground truth visibility flag for keypoint *i*

Table 20 Results of object detection on the YOLOv5s-pose use OKS Loss.

Detector	Backbone	*mAP* (%)	*FPS*
YOLOv5s-pose	CSPDarknet	63.3	10

The output results are as shown in the table 20, with a mAP of 33.3 and only 10 fps. Additional post-processing is still needed to enhance performance. The low accuracy might be due to the insufficient size of the dataset. Another possible reason is that the YOLO POSE model, originally designed for human joint detection, does not perform well on other objects.

Our future work involves creating an instance segmentation dataset for tractor wheels using the YOLOv8-pose model (Jocher, 2023), with 100 images currently available as shown in *Figure 5.3-1*, This model has been previously used for detecting keypoints of various objects. However, to ensure the effectiveness of the instance segmentation model, a large amount of data is generally required. The goal is to create 300 images now.

Figure 5.3-1 Tractor instance segmentation dataset

5.4 Results and discussion

This chapter explained that the reason for the large errors using the positioning method from the previous chapter is due to pixel offset inherent in the bounding box-based positioning approach. It explores the distribution of this pixel offset using Q-Q plots and T-tests. Finally, the results were corrected using a piecewise regression function, and positioning experiments were conducted using this revised method. At the same time, this chapter also attempts to use the YOLO POSE model for keypoint detection on tractors to correct positioning accuracy. However, this task currently requires further research and discussion due to its low recognition accuracy. Future work involves creating an instance segmentation dataset for tractor wheels and tracks to achieve auxiliary positioning.

Field experiments were conducted using the revised data with the correction method introduced in this chapter. The actual distances were measured in spaces between 3.5 m and 15 m by the RTK-GPS as shown in *Figure 5.4-1*. The difference between the actual and predicted distances was recorded for all points at 0.5-m intervals. Each group of experiments was conducted five times, and we used the average of the results. The error results are shown in *Figure 5.4-2* Further objects tend to have a larger error, meaning that the error variance is more prominent. The root mean square (RMS) error of the measured distance was 0.403 m and had an average relative error of 2.6% at 15 m; the maximum error occurs at the farthest point of 15 m, which is 0.77 m. The maximum relative error remains at around 5%, which does not significantly impact the safety system's judgment within the ROI; we thus consider it an acceptable value. We concluded that the remote safety system can correctly detect and accurately locate targets.

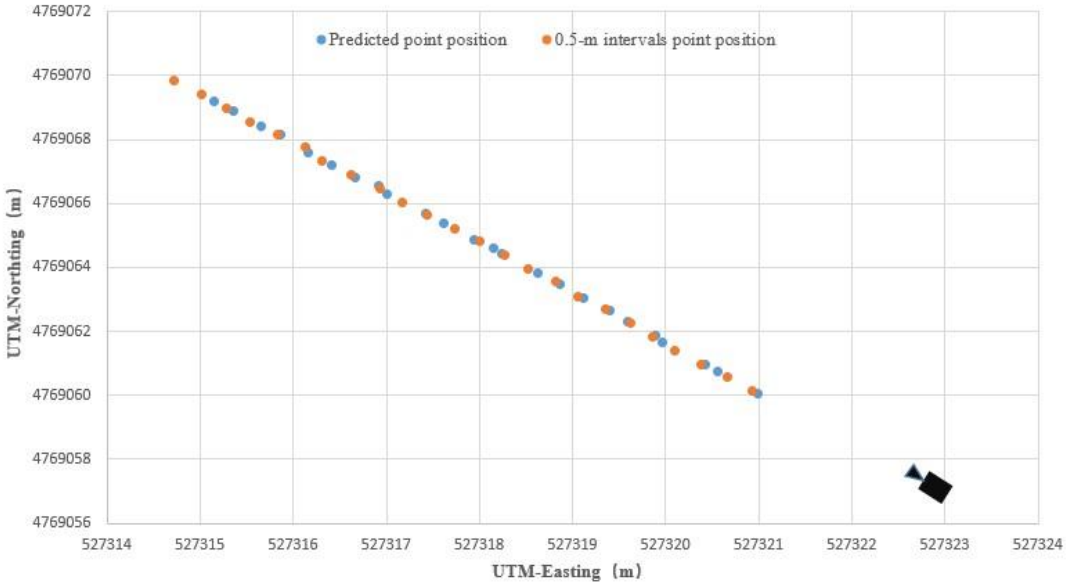


Figure 5.4-1 Position of the range point.

Blue points are the coordinates predicted by the system. Orange points are the coordinate of points measured at 0.5-m intervals using RTK-GPS.

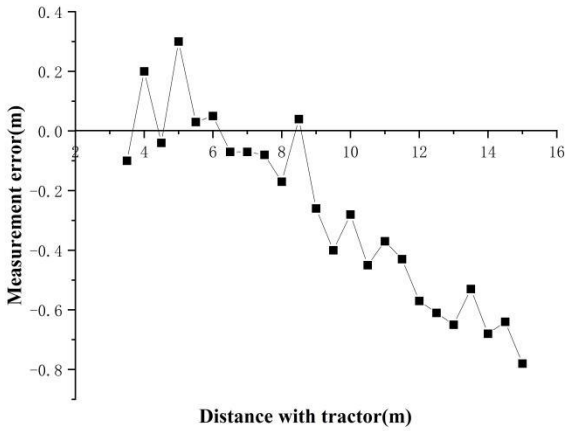


Figure 5.4-2 Measurement error in the X-direction.

The distance error tends to increase proportionally to the distance in our method. This is due to the imaging characteristics of monocular cameras and the method we used to calculate the target position by using the bounding box position.

CHAPTER 6. FIELD EXPERIMENTS FOR THE ROBOT AGRICULTRUE MACHINERY

6.1 Introduction

This chapter presents some farm experiments conducted using the remote safety system, exploring the operation of the system on various platforms, its performance in different environments, and the technical feasibility of the system.

6.2 Materials and methods of field experiments

6.2.1 Field experiments for multi-robot

In October 2021 we set up a remote monitoring room at the Iwamizawa City Data Center as shown in *Figure 6.2-2c* and monitored two robot tractors at the Hokkaido University farm (37 km away) as shown in *Figure 6.2-2b* and another two robot tractors at the Iwamizawa Nishiyauchi farm (7 km away) as shown in *Figure 6.2-2a* as the multi-robot remote monitoring experiment map as shown in *Figure 6.2-1*.

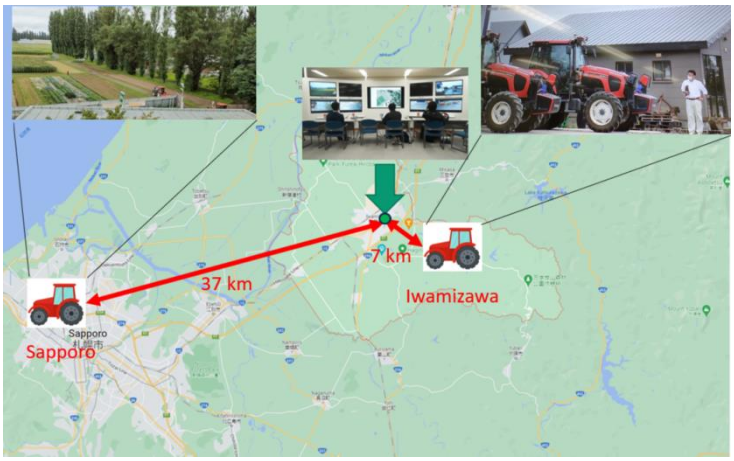


Figure 6.2-1 Remote monitoring experiment map in Hokkaido.

Remote monitoring room at the Iwamizawa City Data Center. Two robot tractors at the Hokkaido University farm. Two robot tractors at Iwamizawa Nishiyauchi farm.

The operation and monitoring of the four tractors are carried out by a single person in the middle, while the students on both sides are responsible only for observing and preventing any unexpected situations as shown in *Figure 6.2-2c*.

(a) (b) (c)

Figure 6.2-2 The remote monitoring experiment.

(a) The Iwamizawa field running experiment. (b) The Sapporo field running experiment. (c) The Iwamizawa City remote monitoring room. In the remote monitoring room, eight screens on the left and right sides are responsible for displaying the image information transmitted from the front and rear cameras of the

four robot tractors. The large screen in the middle shows the positions of the four robot tractors on a map using geographic information system (GIS) software.

The four robot tractors (Kubota MR1000A crawler-type, Kubota MR1000A wheel-type, Yanmar EG105 crawler-type, Yanmar EG83 wheel-type) started from their respective starting points and went along the farm road, arrived at the appropriate field and moved within the field via two paths, and then returned to the starting point the same way. We let obstacles randomly invade the tractors' paths during this time. Every target was detected correctly in the experiment, and all robot tractors acted correctly according to the safety index.

6.2.2 Field experiments for robot EV

Field experiments were also conducted in Tsurunuma and Noto farm. We used this system to monitor the robot EV developed by Yamasaki and Noguchi (2023) as shown in *Figure 6.2-3*, which operates in grape vineyards. EV starts from the warehouse of the vineyard, moves between the rows of trees in the grape estate following a predetermined path, and then returns to the vineyard's warehouse after moving between two rows of trees as shown in *Figure 6.2-4*. We let obstacles randomly invade the EV ' paths. EV started from Every target was detected correctly in the experiment, and all robot tractors acted correctly according to the safety index.

Figure 6.2-3 Robot EV

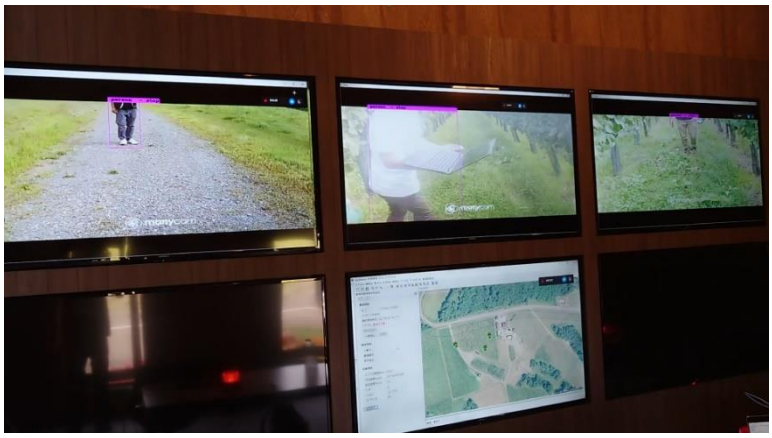


Figure 6.2-4 Monitoring room in Tsurunuma Improve Center

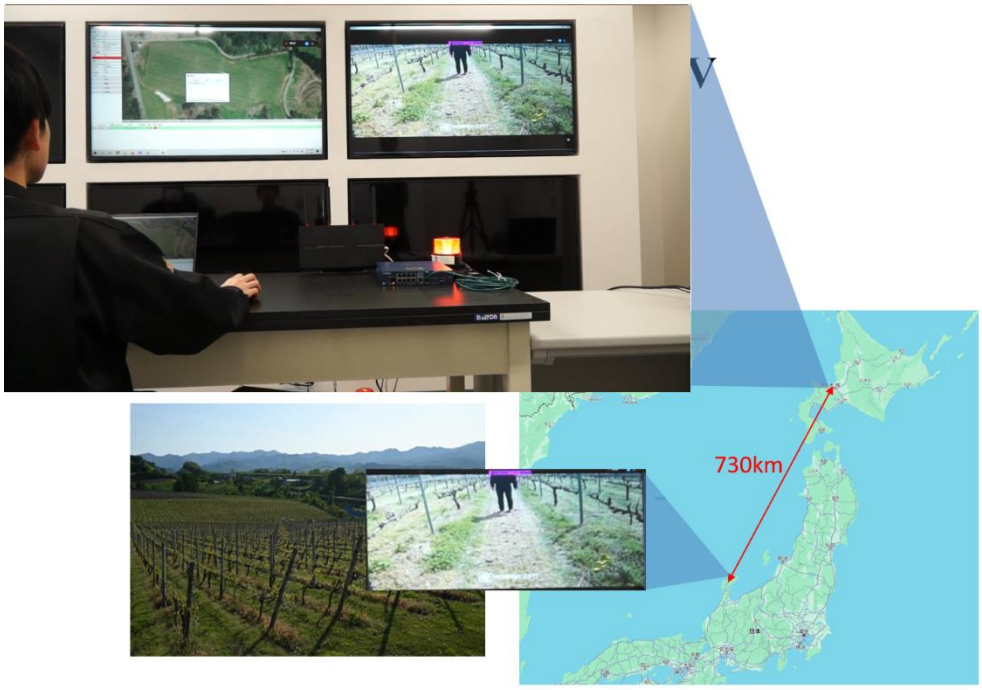


Figure 6.2-5 Monitoring an EV working in the Noto vineyard.

located 730 kilometers away from Hokkaido University.

6.2.3 Results of the field experiments.

Such field experiments were conducted multiple times during 2022–2023 in Tsurunuma valley, Iwamizawa Nishiyauchi farm, the Hokkaido University farm, and Noto farm on the tractor or electric vehicle (EV) developed by Yamasaki and Noguchi (2023). The monitoring locations during the experiment, the robot positions, the distances between each location, the number of robots monitored simultaneously, the instances of obstacle intrusion, and the successful detection of obstacles are presented in Table 21. All obstacles were correctly detected.

Table 21 Field experiments for robot tractor.

Time	Monitoring location	Robot location	D (km)	N	T_o	T_s
2022-05-13	Iwamizawa	Iwamizawa	7	1	8	8
2022-05-19	Iwamizawa	Iwamizawa	7	2	8	8
2022-06-08	Iwamizawa	Sapporo/Tsurunuma	37/29	2	9	9
2022-06-21	Iwamizawa	Sapporo/Tsurunuma/Iwamizawa	37/29/7	3	12	12
2022-07-06	Iwamizawa	Sapporo/Tsurunuma/Iwamizawa	37/29/7	3	12	12
2022-07-12	Iwamizawa	Sapporo/Tsurunuma/Iwamizawa	37/29/7	3	12	12
2022-08-02	Tsurunuma	Tsurunuma	2.8	2	6	6
2022-08-03	Tsurunuma	Tsurunuma	2.8	2	8	8
2022-08-08	Sapporo	Sapporo	1	1	4	4
2022-08-24	Tsurunuma	Tsurunuma	2.8	2	10	10
2022-08-26	Tsurunuma	Tsurunuma	2.8	2	8	8
2022-09-02	Sapporo	Sapporo	0.3	2	8	8
2022-09-21	Sapporo	Sapporo/ Tsurunuma	1/37	3	12	12
2022-10-02	Sapporo	Sapporo	0.3	2	6	6
2022-10-21	Sapporo	Sapporo	0.3	2	4	4
2022-11-09	Sapporo	Sapporo	0.3	1	9	9
2023-04-10	Sapporo	Noto	730	1	6	6
2023-05-22	Tsurunuma	Sapporo/ Tsurunuma	56/2.8	2	6	6
2023-05-24	Sapporo	Tsurunuma	56	2	6	6
2023-05-26	Sapporo	Sapporo	1	2	6	6
2023-06-09	Sapporo	Sapporo/Tsurunuma/Iwamizawa	1/56/37	4	14	14
2023-06-15	Sapporo	Sapporo	0.3	1	4	4

[footnote] D: the distance between monitoring room and robot location; N: number of robots; T_o: times of obstacle intrusion; T_s: times of successful detection.

6.3 Materials and methods of feasibility experiments

6.3.1 Technical feasibility experiments

For the multi-target detection test. We counted the results of a 30-sec video of the detection of multiple people that was recorded in a field experiment on the Iwamizawa City farm. We compared the actual condition of all characters within 15 m of the tractor appearing in the video and the safety conditions predicted by the safety system, as shown in *Figure 6.3-1*. The solid line in the figure represents the real safety condition, and the dashed line represents the predicted safety condition. At 3–8 sec, 12 sec, and 27–30 sec, an error in the estimation of the safety condition occurred due to the effective occlusion generated by the previous person, and no prediction bounding box or wrong bounding box position was generated for these person as shown in *Figure 6.3-2*.

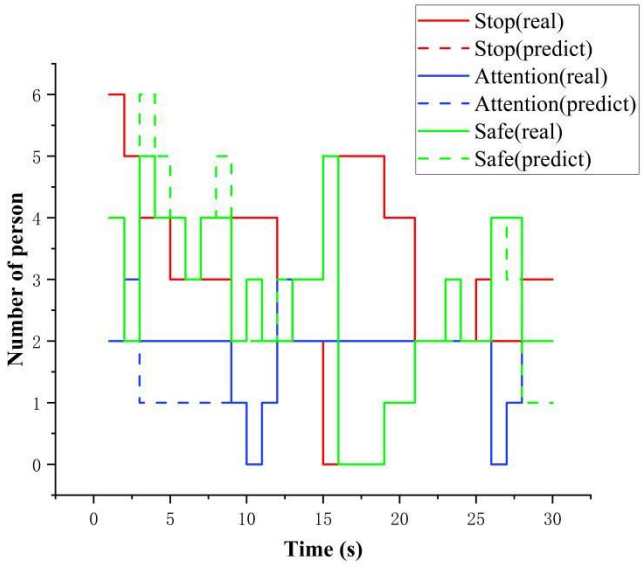


Figure 6.3-1 Real or predict safety conditions for each character, safety condition according to Table 3. If the same-color solid line and dashed line do not overlap at a certain frame, it means that there is a prediction error. The dashed line above the solid line indicates a false positive, while the dashed line

below the solid line indicates a false negative.

Figure 6.3-2 Missed detection due to effective occlusion.

For the three characters on the upper right side of the road, only two bounding boxes were generated.

It should be noted that there were also some small targets beyond 15 m from the tractor that were not detected. This was due to the lower detection ability of YOLO for small-scale targets compared to large-scale targets, and it will cause missed detections due to occlusion, for which the assistance of LiDAR is necessary for completing the safety evaluation. From the results, it can be seen that the system is prone to missed detections or false alarms due to occlusions. Fortunately, since the occluding objects are always behind the missed or falsely detected objects, although the judgment results of the occluded objects may be affected, the final safety judgment of the robot tractor is generally not affected.

We tested the robot tractors' braking distance at different speeds. The experiment was conducted at the Hokkaido University farm as shown in *Figure 6.3-3*.

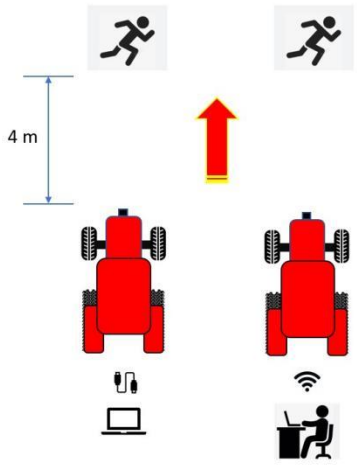


Figure 6.3-3 Remote-control performance test experiment.

A person stood 4 m in front of the tractor and invaded the tractor route under the network connection or the cable connection to the tractor control computer, respectively (with the LiDAR off), and we measured and recorded the braking distance of the tractor. Each respective test was conducted five times.

The average braking distance at different speeds is shown in Table 22. The results indicate that remote transmission affects robot-tractor braking due to network delays. The increase in vehicle speed increases this effect, but it is acceptable for the robot tractor, which always runs at a low speed (normally 3 km/h). Additionally, the robot tractor will slow down before a potential obstacle enters its path, ensuring safety during the tractor's regular operation under remote control. When only LiDAR is used as a safety device, setting the LiDAR detection range too far can lead to issues where the tractor stops, for example, when detecting corn stalks while passing through a cornfield. When only a camera is used as a safety device, network delays may pose some security risks for remote control; thus, linkage with locally connected security

devices (e.g., LiDAR) is essential.

Because visual braking relies on transmitting images captured by the robot's camera to a monitoring station via a mobile Wi-Fi network for remote processing and then sending back the image-processing results to the robot for execution, both the braking distance and the braking time can be influenced by the speed and stability of the network communication. We conducted tests using two different networks to assess their impact on remote monitoring. One network was a standard LTE mobile router available for commercial lease which had relatively unstable network connectivity, a minimum download speed of 20 Mbps, and an upload speed of 5 Mbps. On the day of testing, the network exhibited a download speed of 317 Mbps and an upload speed of 20 Mbps. The other network was a 5G mobile router provided by NTT East Japan, which had more stable network connectivity. The 5G network utilizes a technology known as Multi-Access Edge Computing (MEC). The delay generated during signal transmission by the tractor comprises latency in the wireless segment and latency in the wired segment. MEC minimizes the latency in the wireless segment by locating servers closer to the end devices in terms of physical distance. In the experiments, the server used was situated in a building approximately 400 meters away from the tractor. On the day of testing, this network demonstrated a download speed of 264 Mbps and an upload speed of 126 Mbps. Under different network conditions, we conducted tests to determine the time and distance required for the robot to come to a complete stop at its maximum speed while solely visual braking was relied upon during remote control. The testing methodology was consistent with the procedures outlined above in Section 4.5.

The test results are shown in Table 23. It can be observed that under the three different network environments, the impact on braking time is minimal. However, the

upload speed does have some influence on the braking distance. We believe that this is because an insufficient upload speed can lead to delays when transmitting the images captured by the tractor robot's front camera to the remote end, resulting in a delay in receiving commands from the remote monitoring end. Note that the command transmission requires very little network bandwidth, and thus the download speed has a minimal impact on the tractor's braking distance. The braking time, in contrast, is controlled primarily by the tractor robot's local program and is not significantly affected by the network environment. In summary, for remotely controlled tractor robots using this approach, the safety strategy should be determined based on the minimum upload speed according to the specific network environment they are in. Additionally, lower network latency and upload speeds contribute to the improvement of the robot's safety.

Table 22 Results of the remote-control performance test.

Control Method	Robot speed (km/h)	Braking distance (m)
Edge	2	1.05
	3.5	1.17
	5	1.68
remote	2	1.17
	3.5	1.41
	5	2.09

Table 23 Results of the remote-control performance test.

Network type	$\bar{t}$ *(s)*	$\bar{d}$ (m)	Download (Mbps)	Upload (Mbps)
5G	3.5	7.4	264	**126**
LTE	**3.6**	**10.2**	**317**	20
LTE	3.5	11	30	10

[footnote] $\bar{t}$: the average braking time, $\bar{d}$: the average braking distance.

6.3.2 Environment feasibility experiments

To explore the system's stability under different environmental conditions, we monitored the robot both at night as shown in *Figure 6.3-1* and in snowy weather as shown in *Figure 6.3-2*. The experiments were conducted at the Iwamizawa Farm. For the night experiment, the test started from the entrance of the farmland, and detection was carried out on images captured using only the tractor headlights as a light source. The experimental method was the same as the one described in section 6.2, and all targets were correctly detected.

Figure 6.3-4 Night experiment in Iwamizawa

Figure 6.3-5 Snowy experiment in Iwamizawa

Due to the inability to enter the farmland in snowy weather, the experiment was conducted on the farm road beside the farmland. The tractor started from the Nishiyauchi Farm, passed through the farm road outside the Nishiyauchi Farm, returned at the entrance of the farmland, and then went back to the Nishiyauchi Farm. The testing method was the same as the previously mentioned experiments. All targets were correctly detected.

6.3.3 Conclusions

In the application of robotic agricultural machinery, ensuring safety is of utmost importance. Therefore, we conducted numerous field experiments during the period of 2022-2023. The field tests for multi-robot monitoring demonstrated that the system could assist a small number of operators in controlling multiple robotic farm machines, ensuring safety during remote monitoring. The surveillance of different platforms of tractors and EVs showed that the system is not limited to a single type of farm machine, but is versatile across various platforms. The monitoring experiment in Noto proved that the system is capable of supervising robots over long distances. Experiments conducted in snowy conditions and at night confirmed the system's usability in a variety of different environments. The experiments on multi-person detection indicated that the system could accurately assess safety levels in complex scenarios. At the same time, experiments on network conditions also confirmed that the system is subject to network environment constraints. Therefore, combining local 2-D laser sensors to ensure robot safety in cases of network fluctuations or failures is crucial. In summary, the feasibility of the remote safety system has been confirmed through multiple field experiments.

CHAPTER 7. RESARCH SUMMARY

To ensure the safety of level 3 remote monitoring for the robot agriculture machinery, we developed a remote safety system that uses a monocular camera and 2D-LiDAR to obtain data and a YOLOv5s model as a detector with a detection mAP value of 87.3% for human and tractor detection. This system can perform an image analysis at 32 FPS on a Quadro P4000 GPU-based workstation. We used PnP method with the bounding box to calculate the obstacle distances to ensure safety during remote monitoring the robot, data is sent to the remote end for processing, and the detected images are corrected using methods derived from Q-Q plots and T-tests. After data correction, obstacle distances are predicted with an average relative error of 2.6%, maximum error of 0.77 m at 15 m, which is meeting the requirements for safe usage, and the remote control of multiple robot tractors according to the safety index can be performed. This novel system is not restricted by physical distance and is low cost, with high stability and resilience to environmental factors. The system achieved a 100% success rate in responding to obstacle intrusions in 2022-2023 field experiments. The present experimental results demonstrate that the new system can assist an individual operator with the remote monitoring of robot tractors, thus saving labor costs and improving efficiency in agricultural applications. However, this system also has its deficiencies, including limited precision in tractor positioning, a restricted range of visual localization objects (only humans and other tractors), and susceptibility to network environmental constraints. We plan to address these issues in our future research.

References

Albiero, D., Garcia, A. P., Umezu, C. K., and de Paulo, R. L. (2022). Swarm robots in mechanized agricultural operations: a review about challenges for research. Computers and Electronics in Agriculture, 193, 106608.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... and Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8(1), 1-74.

Arnold, E., Al-Jarrah, O. Y., Dianati, M., Fallah, S., Oxtoby, D., and Mouzakitis, A. (2019). A survey on 3d object detection methods for autonomous driving applications. IEEE Transactions on Intelligent Transportation Systems, 20(10), 3782-3795.

Bhat, S. A., Hussain, I., and Huang, N. F. (2023). Soil suitability classification for crop selection in precision agriculture using GBRT-based hybrid DNN surrogate models. Ecological Informatics, 75, 102109.

Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv: 2004.10934.

Brunetti, A., Buongiorno, D., Trotta, G. F., and Bevilacqua, V. (2018). Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. Neurocomputing, 300, 17-33.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... and Zieba, K. (2016). End to end learning for self-driving cars. arXiv preprint arXiv: 1604.07316.

Cabinet Office, (2017). What is the Quasi-Zenith Satellite System (QZSS)? Quasi-Zenith Satell. Syst. URL: http://qzss.go.jp/en/overview/services/sv02_why.html Accessed 2023.10.17.

Cabinet Office, (2017). Centimeter Level Augmentation Service (CLAS). URL: https://qzss.go.jp/en/overview/services/sv06_clas.html. Accessed 2023.10.17.

Castanedo, F. (2013). A review of data fusion techniques. The Scientific World Journal, 2013, 704504.

Chambers, J. M., Cleveland, W. S., Kleiner, B., and Tukey, P. A. (2018). Graphical methods for data analysis. Chapman and Hall/CRC.

Chateau, T., Debain, C., Collange, F., Trassoudaine, L., & Alizon, J. (2000). Automatic guidance of agricultural vehicles using a laser sensor. Computers and electronics in agriculture, 28(3), 243-257.

Dai, J., Li, Y., He, K., and Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. Advances in Neural Information Processing Systems, 29, 379-387.

Daniel, W. W., and Cross, C. L. (2018). Biostatistics: a foundation for analysis in the health sciences. Wiley.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255).

Ferreira, R. E., Bresolin, T., Rosa, G. J., and Dórea, J. R. (2022). Using dorsal surface for individual identification of dairy calves through 3D deep learning algorithms. Computers and Electronics in Agriculture, 201, 107272.

Fischler, M. A., and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395.

Firkat, E., An, F., Peng, B., Zhang, J., Mijit, T., Ahat, A., ... & Hamdulla, A. (2023). FGSeg: Field-ground segmentation for agricultural robot based on LiDAR. Computers and Electronics in Agriculture, 211, 107965.

Gai, J., Xiang, L., and Tang, L. (2021). Using a depth camera for crop row detection and mapping for under-canopy navigation of agricultural robotic vehicle. Computers and Electronics in Agriculture, 188, 106301.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

Guo, L., Zhang, Q., and Han, S. (2002). Agricultural machinery safety alert system using ultrasonic sensors. Journal of Agricultural Safety and Health, 8(4), 385.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Jocher G., Nishimura K., Mineeva T., and Vilariño R. (2022). YOLOv5. URL: https://github.com/ultralytics/yolov5

Jocher G (2023) YOLOv8 Pose Models URL: https://github.com/ultralytics/ultralytics/issues/1915 Accessed 2023.12.10

Tzutalin. (2015). LabelImg. URL: https://github.com/HumanSignal/labelImg Accessed 2023.11.6.

Li, Y., and Ibanez-Guzman, J. (2020). Lidar for autonomous driving: The principles, challenges, and trends for automotive Lidar and perception systems. IEEE Signal Processing Magazine, 37(4), 50-61.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... and Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

Ma, J., Wu, Y., Liu, B., Zhang, W., Wang, B., Chen, Z., ... and Guo, A. (2023). Wheat Yield Prediction Using Unmanned Aerial Vehicle RGB-Imagery-Based Convolutional Neural Network and Limited Training Samples. Remote Sensing, 15(23), 5444.

MAFF. (2017) (2019) (2021) Safety Assurance Guidelines for Agricultural Machinery Autonomous Navigation.URL:https://www.maff.go.jp/j/press/seisan/gizyutu/attach/pdf/210326-3.pdf Accessed 2023.10.25.

MAFF. (2020) Demonstration of Smart Agriculture Model Utilizing Local 5G Technology.URL:https://www.affrc.maff.go.jp/docs/smart_agri_pro/kanren/R2seika/R2_2-5_s.pdf Accessed 2023.11.1.

MAFF. (2021) The 96th Statistical Yearbook of Ministry of Ministry of Agriculture, Forestry, and Fisheries. URL:https://www.maff.go.jp/e/data/stat/96th/index.html Accessed 2023.11.1.

MAFF. (2023) What is remote-monitoring robot agriculture machinery? URL:https://www.maff.go.jp/j/keiei/nougyou_jinzaiikusei_kakuho/attach/pdf/smart_kyoiku-18.pdf Accessed 2023.11.20.

Maji, D., Nagori, S., Mathew, M., and Poddar, D. (2022). Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2637-2646).

Moorehead, S. J., Wellington, C. K., Gilmore, B. J., and Vallespi, C. (2012, October). Automating orchards: A system of autonomous tractors for orchard maintenance. In Proceedings of the IEEE international conference of intelligent robots and systems, workshop on agricultural robotics.

Noguchi, N., Ishii, K., and Terao, H. (1997). Development of an agricultural mobile robot using a geomagnetic direction sensor and image sensors. Journal of Agricultural Engineering Research, 67(1), 1-15.

Noguchi, N. (2000). Engineering challenges in agricultural mobile robot towards information agriculture. In Proceedings of the XIV Memorial CIGR World Congress, 2000 (pp. 147-154).

Noguchi, N., and Barawid Jr, O. C. (2011). Robot farming system using multiple robot tractors in Japan agriculture. IFAC Proceedings Volumes, 44(1), 633-637.

Noguchi, N. (2023). Listening to Professor Noguchi of Hokkaido University: Challenges and Future Prospects of Smart Agriculture in Mountainous Areas, MichibikiHP.URL:https://qzss.go.jp/usage/userreport/noguchi_230508.html Accessed 23.11.1.

Plaut, D. C. (1986). Experiments on Learning by Back Propagation.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Redmon, J. and Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv: 1804.02767.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems, 28, 91-99.

Roboflow, Inc. (2022). Annotation tool. URL: https://roboflow.com/annotate Accessed 2023.10.25.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

Shen, M., Wang, Y., Jiang, Y., Ji, H., Wang, B., and Huang, Z. (2019). A new positioning method based on multiple ultrasonic sensors for autonomous mobile robot. Sensors, 20(1), 17.

Shorten, C., and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 1-48.

Sun, B., Li, W., Liu, H., Yan, J., Gao, S., and Feng, P. (2021). Obstacle detection of intelligent vehicle based on fusion of LiDAR and machine vision. Engineering Letters, 29(2), 722-730.

UN DESA. (2022) World population prospects 2022—Summary of Results. URL:https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/wpp2022_summary_of_results.pdf Accessed 2023.11.1.

Yamasaki, Y., and Noguchi, N. (2023). Research on autonomous driving technology for a robot vehicle in mountainous farmland using the Quasi-Zenith Satellite System. Smart Agricultural Technology, 3, 100141.

Zhang, C., and Noguchi, N. (2017). Development of a multi-robot tractor system for agriculture field work. Computers and Electronics in Agriculture, 142, 79-90.

Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6848-6856).

Zhang, Z. (2000). A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11), 1330-1334.