https://doi.org/10.7840/kics.2024.49.7.1021

Deep-Learning Based Missing Child Detection Assistance System Using Autonomous Robot

Yeong Eun Choi◆, Su Hyeon Kang*, So Yeon Kim*, Soo Young Shin°

ABSTRACT

This paper proposes a deep learning-based missing child detection assistance system using autonomous robots. The proposed system utilizes the clothing and height information of the missing child, which is received upon reporting a missing child. It employs deep learning-based multi-label classification techniques to search for clothing information and utilizes object detection techniques along with distance information to search for height information. Autonomous robots are employed, utilizing SLAM(Simultaneous Localization and Mapping) to perform path planning and localization based on a pre-generated 2D map. Finally, when the autonomous robot successfully identifies the missing child using clothing and height information, its location is transmitted to the Ground Control Station (GCS).

Key Words : Height estimation, Multi-label classification, SLAM(Simultaneous Localization and Mapping), Path planning

Ⅰ. Introduction

According to data released by the Ministry of Health and Welfare, the annual number of reported missing children in South Korea ranged from approximately 20,000 cases in 2018~2021[1]. And as of April of 2022, there were about 7,500 reported cases. While the annual discovery rate of missing children stands at 99.5%, indicating a high success rate in locating them, there are still accumulated cases of children missing for over a year, totaling 871 cases as of April 2022. This underscores the significant scale of the issue of missing children. Therefore, early detection when a child goes missing is crucial to prevent long-term disappearances. To address this issue, various technologies and systems have been developed to detect missing child early.

One such example is the multi-complex processing signal technology through mobile devices[2]. This technology utilizes devices worn by the missing child, such as smartphones, necklaces, or wristbands, to receive a variety of signals, including Wi-Fi, LTE, and GPS, accurately pinpointing the missing child's location. However, this technology may face limitations as there is a possibility that the missing child may not have the device with them, and it may not function correctly in areas with signal interference or weak signal reception. For these reasons, camera-based technologies that can be applied without the need for the missing child's consent or cooperation

※ "This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ICAN(ICT Challenge and Advanced Network of HRD) program(IITP-2024-RS-2022-00156394) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation"

※ "This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Innovative Human Resource Development for Local Intellectualization support program (IITP-2024-2020-0-01612) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation)"

◆ First Author : Kumoh National Institute of Technology, choiye1122@naver.com, 학생회원

° Corresponding Author : Kumoh National Institute of Technology, wdragon@kumoh.ac.kr, 종신회원

* Kumoh National Institute of Technology, 20236101@kumoh.ac.kr; dus5125@kumoh.ac.kr

논문번호 : 202311-138-D-RN, Received November 10, 2023; Revised January 6, 2024; Accepted March 25, 2024

are preferred over mobile device tracking technologies.

Technologies utilizing cameras use computer vision systems. These are particularly prevalent in CCTV(Closed-Circuit Television) systems, where they enable the collection of large amounts of data relatively easily from video footage and offer the advantage of being able to survey wide areas.

However, CCTV systems could have blind spots where certain areas cannot be observed may have difficulty identifying objects when they are far away or small in size. Therefore, in this paper, we propose deploying autonomous robots in multi-use facilities to address these limitations.

In addition, a deep learning based missing person recognition system is applied. This system has shown strong performance in pattern recognition and classification from data. However, in the case of deep-learning based facial recognition technology, it cannot identify faces When a person is positioned opposite to the camera and wearing a hat. Therefore, training deep learning models using external features such as a person's clothing or height information can be useful for identifying and searching for missing individuals[3].

Considering the constraints mentioned above and the existing limitations of current missing child prevention systems, this paper adopts the following deep learning and fundamental autonomous driving technologies. Firstly, to extract height information, a depth camera, which provides pixel-level depth information, is utilized to adopt a height estimation algorithm[4]. Subsequently, for obtaining clothing information, a more detailed multi-label classification, as opposed to traditional image classification methods, is used due to the diverse nature of clothing[5]. Additionally, for autonomous operation of robots indoors, Simultaneous Localization and Mapping (SLAM) technology is utilized to acquire spatial information (map) and real-time location information[6]. Finally, based on spatial information (map), navigation is facilitated through path planning to reach the destination[7].

Accordingly, this paper proposes and implements research on missing child exploration using deep learning and autonomous driving robots, utilizing the four key technologies mentioned above. Through this, in the event of a missing child, robots equipped with rapid information processing and decision-making capabilities can respond swiftly and contribute to accident prevention with minimizing blind spots.

In Chapter II of this paper, an explanation of the entire system is provided, along with detailed processes related to height estimation, multi-label classification, SLAM, and navigation. Chapter III discusses the implementation methods and results of these detailed processes. Lastly, in Chapter IV, the paper summarizes and presents its conclusions.

Ⅱ. System

2.1 System Model

The overall system algorithm of this study is implemented in the ROS (Robot Operating System) environment, which includes hardware control, communication between detailed processes (multi-label classification, height estimation, SLAM, and navigation), and the use of 3D visualization tools.

Fig. 1 illustrates the overall system model. The system initiates the search upon receiving information about the missing child, including height and clothing details. Robot navigation for the search is performed

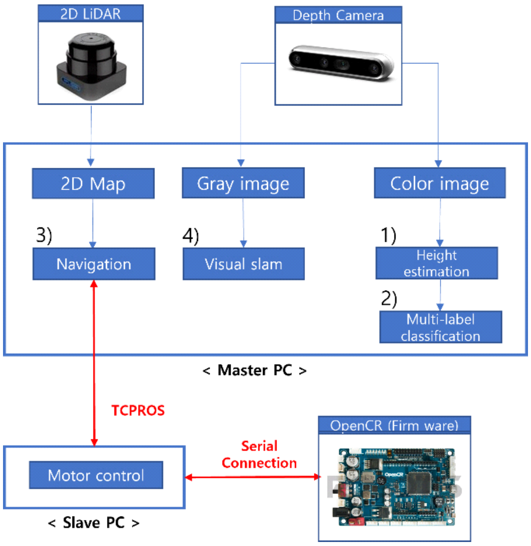


Fig. 1. System Model

on a pre-drawn 2D map using 2D LiDAR. Additionally, visual SLAM for environmental perception is conducted based on feature points extracted from gray images obtained from the depth camera. Simultaneously, height estimation and multi-label classification for finding the missing child are executed sequentially, with color images from the depth camera being processed through their respective deep learning models. All three functionalities (deep learning, navigation, SLAM) communicate through ROS on the master PC. Motor control on the slave PC receives ROS messages through a TCPROS connection with the navigation on the master PC, enabling the operation of motors connected to the robot controller firmware.

2.2 Height estimation

To conduct height estimation(Represented as 1) in Fig. 1), the process begins by detecting individuals using the color data from the depth camera. Subsequently, a 3D coordinate array is created using Depth data. Then, for each person, their individual 3D coordinates are examined to extract distances. Out of the extracted distances, Y-values with significant deviations from the Z-median value are removed as outliers. The resulting Y-direction length information is then used to estimate the height of each person.

2.3 Multi-label classification

To conduct height estimation(Represented as 1) in Fig. 1), the process begins by detecting individuals using the color data from the depth camera. Subsequently, a 3D coordinate array is created using Depth data. Then, for each person, their individual 3D coordinates are examined to extract distances. Out of the extracted distances, Y-values with significant deviations from the Z-median value are removed as outliers. The resulting Y-direction length information is then used to estimate the height of each person.

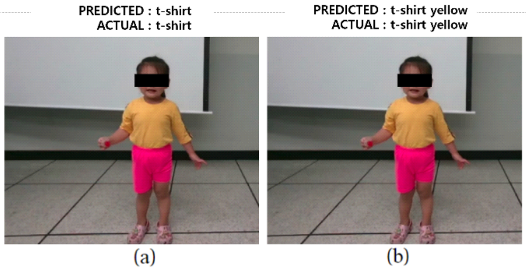


Fig. 2. Classification comparison : (a) single-label, (b) multi-label

2.4 SLAM and Navigation

In the robot's movement section for missing child search, navigation(Represented as 3) in Fig. 1) and SLAM(Represented as 4) in Fig. 1) are utilized. The robot performs Navigation based on a pre-drawn 2D Map[9]. For Path Planning in Navigation, both the DWA (Dynamic Window Approach) algorithm suitable for dynamic environments and the A* algorithm suitable for static environments are employed together[10]. Additionally, the TSP (Travelling Salesman Problem) algorithm is used, where after determining the number and coordinates of waypoints, the optimal path is planned by considering the cost between each point. After visiting all waypoints, the robot returns to the starting point and repeats this process until the missing child is located.

For the assessment of the missing child's location and the surrounding environment, the robot employs the ORB SLAM3 algorithm, a Visual SLAM(Represented as 4) in Fig. 1) technique using cameras[11]. While navigating through the environment following its path planning, the robot utilizes ORB SLAM3 to construct a 3D map. This real-time 3D map allows the robot to continually determine its current position and assess the surroundings within the multi-purpose facility. Ultimately, once the algorithms, including Navigation and ORB SLAM3, have successfully detected both the child's height and clothing information, motor control is initiated to stop the robot. Subsequently, using the SLAM's localization capabilities, the approximate child's position is transmitted to the Ground Control Center(GCS).

Ⅲ. Implementation

The hardware configuration for system implementation is depicted in Fig. 3. A laptop serves as the Master PC, controlling other devices such as the 2D Lidar and Depth Camera, as well as managing detailed processes. The Companion board functions

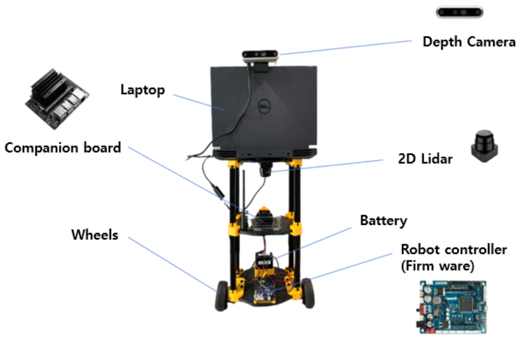


Fig. 3. Hardware configuration

Table 1. Hardware specification

Device	name
Laptop (Master PC)	NVIDIA GeForce RTX 3060 Laptop
Companion board (Slave PC)	Jetson Nano 4GB
2D Lidar	RPLidar S1
Depth camera	Intel realsense depth camera D455
Robot controller (Firm ware)	OpenCR 1.0

as the Slave PC, equipped with firmware for motor control and connected to enable robot movement control as instructed by the Master PC.

The entire algorithm has been implemented within a Docker container running on Ubuntu 18.04, with ROS Melodic installed. Within this environment, all processes, including multi-label classification, height estimation, SLAM (Simultaneous Localization and Mapping), and Navigation, communicate with each other and execute as part of the integrated system.

The experiment was conducted using a female subject with a height of 167cm and wearing a striped top with black pants. The robot search for a person matching this description while moving around. When the robot detected the target, it report the discovery to the GCS. The following provides detailed explanations for each of the processes.

3.1 Height estimation

For height estimation, the RealSense D455 camera, capable of measuring distances between objects and the camera, was utilized. The height estimation algorithm, depicted in Fig. 4, is implemented as a single node, which is the smallest executable unit in ROS. It all begins when the system receives a message containing the missing person's height information through the 'input' topic. This message is subscribed to within the RECEIVE_HEIGHT function. The message is used for comparison with the results of estimating the person in the camera images during the ESTIMATION function. In the ESTIMATION function, before performing height estimation, a confidence score threshold of 0.9 (90%) was set for the person class to ensure accurate person extraction. Then, assuming that the camera and the person are horizontally aligned, experiments were conducted with four different individuals for height estimation. The results showed that the highest accuracy was achieved at a distance of over 2.4 meters. Therefore, considering a slight margin of error, height estimation is only performed when the distance of the detected object to the camera falls within the range of 2.4(±0.1) meters.

```
Algorithm 1 Person Height Estimation
  Initialize ROS node ('Person Height Estimation')
  Create Subscriber to 'Input' topic, set callback as RECEIVE_HEIGHT
  Create publisher for 'height_detect' topic
  Initialize pipeline
  Initialize point_cloud
  Load TensorFlow model

  function RECEIVE_HEIGHT(msg)
      Update find_height = msg.height
  end function

  function ESTIMATION
      while True do
          frames ← pipeline.wait_for_frames()
          depth_frame ← frames.get_depth_frame()
          color_frame ← frames.get_color_frame()
          points ← point_cloud.calculate(depth_frame)
          for idx in range(int(num)) do                ▷ num is detection number
              class ← classes[idx]
              score ← scores[idx]
              box ← boxes[idx]
              if score > 0.9 and class = 1 then             ▷ class 1 is person
                  dist ← depth.get_distance(int(center_x), int(center_y))
                  if dist > 2.3 and dist < 2.5 then
                      Draw bounding box on color_image
                      Calculate object's height from 3D points
                      if height = find_height then
                          Publish 'height_detect' topic
                          Stop the pipeline
                      end if
                      Add height and distance text on the image
                  end if
              end if
          end for
          Display annotated color_image
      end while
  end function
```

Fig. 4. Height estimation algorithm

The Table 2 shows the results of height estimation based on actual measurements of 10 individuals and the average error. The measurements were taken when detected within a range of 2.4(±0.1) meters from the camera, and the average detection distance was 2.389 meters, with an average error of 1.3 centimeters.

Table 2. Height estimation accuracy

Count	Distance(m)	Ground truth(cm)	Result(cm)	Error(cm)
1	2.45	155	155	0
2	2.32	158	156	2
3	2.41	163	164	1
4	2.43	165	163	2
5	2.33	173	173	0
6	2.4	175	174	1
7	2.39	175	174	1
8	2.32	176	175	1
9	2.42	177	180	3
10	2.42	180	178	2
average distance	2.389		average error	1.3

3.2 Multi-label classification

For pretraining the multi-label classification model, the dataset utilized was the Clothing Dataset available on Kaggle. The dataset consisted of over 2000 tops classified into 9 classes and 900 bottoms classified into 9 classes, based on two criteria: color and type. Additionally, the entire dataset of top images and bottom images was split into 85% for training and 15% for validation. The model used for training is the pre-trained ResNet50 deep learning model from PyTorch. The training parameters include 50 epochs and a batch size of 32.

The multi-label classification algorithm, depicted in Fig. 6 is a single node that subscribes to the 'Input' topic, receiving messages in the RECEIVE_CLOTHES function. The messages received through this topic contain information about the missing person's clothing color and type. This information is utilized within the LIVESTREAM function, where the process involves obtaining frame images one by one from the video stream. These frame images are then processed through the multi-label classification model for real-time inference. By comparing the received information with the inference results, if they match for five consecutive times within a 15-second timeframe, it is considered a detection event, and a 'detect' topic is published.

The Fig. 5 depicts the training loss and validation loss for the top and bottom. As the number of epochs increases, in Fig.5. (a), it can be seen that the loss gradually decreases, while in Fig. 5. (b), the loss is shown to decrease sharply in the early stages.

The Table 7 represents the results of testing different combinations of colors and types for tops and bottoms. Among the 14 combinations for tops, 10 were successfully detected, showing a 61.1% proba-

Table 3. Function description1

Function	description
RECEIVED_HEIGHT	Getting missing child's Height information
ESTIMATION	Height estimation livestreaming and comparing with input

Table 4. Dataset information

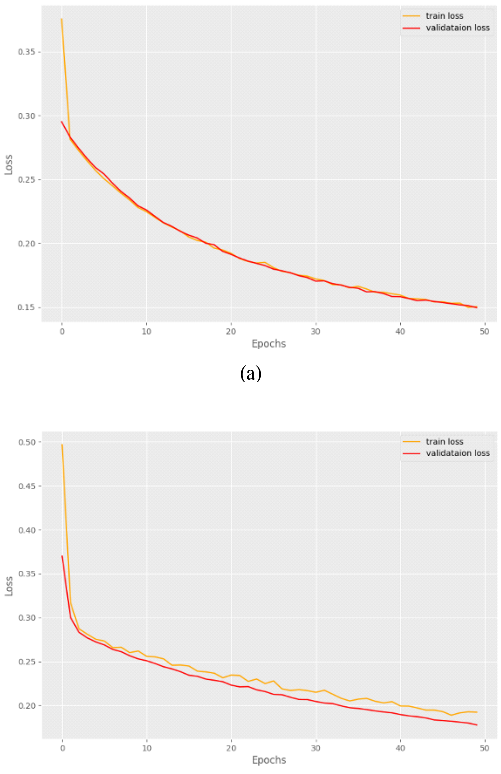
Dataset type	Training dataset	Validation dataset	Total dataset
Top	1,700	300	2000
Bottom	765	135	900

Table 5. Parameter value

parameter	value	parameter	value
Training model	Pytorch : Resnet50	Epoch	50
Batch size	32		

Table 6. Function description2

Function	description
RECEIVED_CLOTHES	Getting missing child's clothes information
LIVESTREAM	Multilabel classification livestreaming and comparing with input



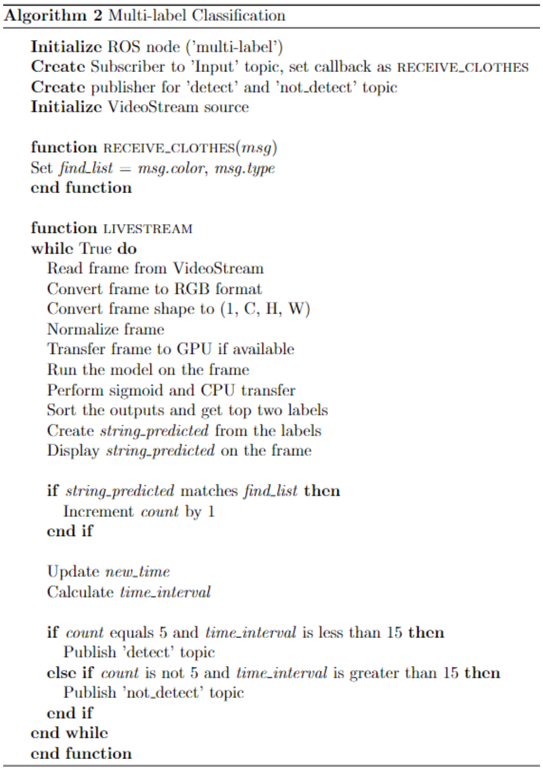
(a)

(b)

Fig. 5. Multi-label classification loss : (a) Top, (b) Bottom

Table 7. Multi-label classification accuracy

Dataset	Color	Type	Combination	No. of detection	Probability
Top	7	2	14	10	61.1%
Bottom	6	3	18	11	71.4%



Algorithm 2 Multi-label Classification

```
Initialize ROS node ('multi-label')
Create Subscriber to 'Input' topic, set callback as RECEIVE_CLOTHES
Create publisher for 'detect' and 'not_detect' topic
Initialize VideoStream source

function RECEIVE_CLOTHES($msg$)
Set $find\_list = msg.color, msg.type$
end function

function LIVESTREAM
while True do
  Read frame from VideoStream
  Convert frame to RGB format
  Convert frame shape to (1, C, H, W)
  Normalize frame
  Transfer frame to GPU if available
  Run the model on the frame
  Perform sigmoid and CPU transfer
  Sort the outputs and get top two labels
  Create $string\_predicted$ from the labels
  Display $string\_predicted$ on the frame

  if $string\_predicted$ matches $find\_list$ then
    Increment $count$ by 1
  end if

  Update $new\_time$
  Calculate $time\_interval$

  if $count$ equals 5 and $time\_interval$ is less than 15 then
    Publish 'detect' topic
  else if $count$ is not 5 and $time\_interval$ is greater than 15 then
    Publish 'not_detect' topic
  end if
end while
end function
```

Fig. 6. Multilabel classification algorithm

bility of success. For bottoms, out of 18 combinations, 11 were successfully detected, demonstrating a 71.4% success rate.

3.3 SLAM and Navigation

Before conducting Navigation, coordinates for four waypoints were predefined. Subsequently, a 2D Map created with LiDAR SLAM was loaded, and the TSP (Traveling Salesman Problem) route planning algorithm was applied for Navigation. The results of this operation are depicted in Fig. 7, where the robot navigates through four waypoints in the order depicted in the figure. Simultaneously, the robot utilized ORB SLAM3, implemented with a camera, to gain real-time situational awareness of its surroundings. The results are illustrated in Fig. 8.

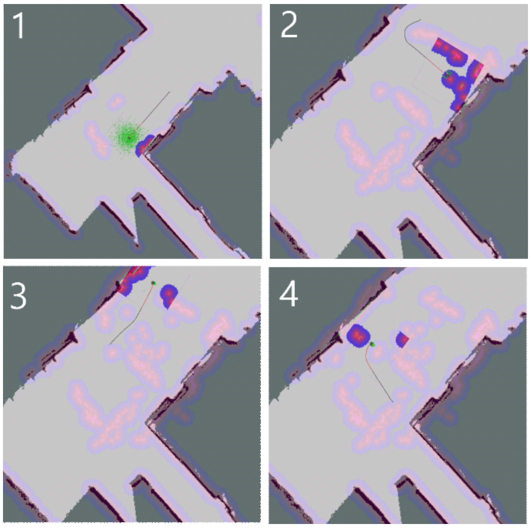
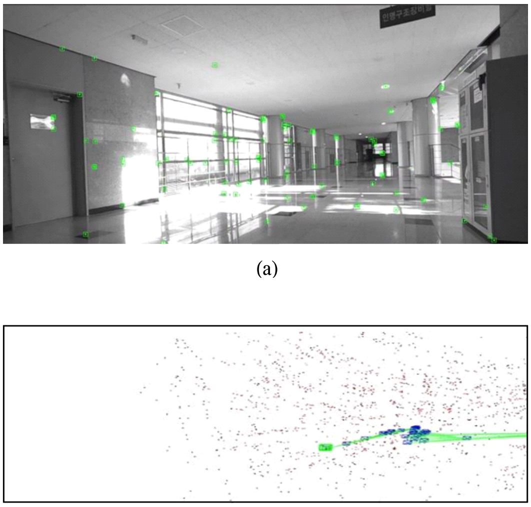


Fig. 7. TSP algorithm implementation



(a)

(b)

Fig. 8. ORB SLAM3 : (a) Current Frame, (b) Map Viewer

3.4 System implementation

The entire system initiates exploration upon receiving input information consisting of height data (167cm) and clothing details, including top (stripe, longsleeve) and bottom (black, pants) attributes. During the exploration, the robot performs two stages of processing: height estimation and multi-label classification, all based on the provided information. Starting from the top-left corner and proceeding counterclockwise, the Fig. 9 represents path planning, deep learning, real-time 3D map generation using ORB

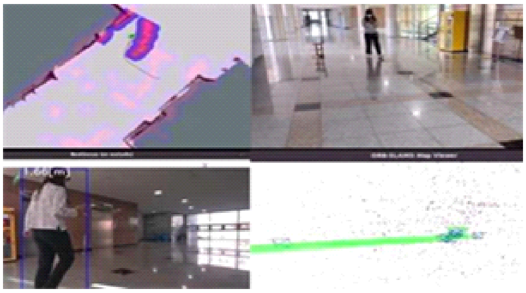


Fig. 9. Entire algorithm implementation

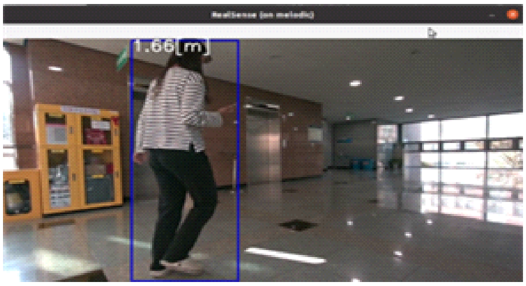
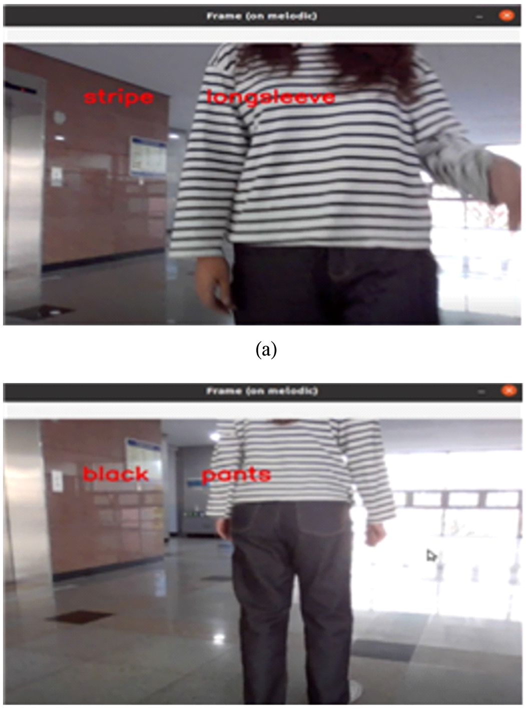


Fig. 10. Height estimation result



(a)

(b)

Fig. 11. Multi-label classification result

SLAM3, and the exterior view. Each of these processes is carried out concurrently. During the execution of the TSP path planning algorithm, the robot, at the second waypoint out of the total four waypoints, discovers a target with matching height information as input. Following that, multi-label classification was performed, and the detection results for height and attire information are depicted in Fig. 10 and Fig. 11, respectively.

Ⅳ. Conclusion

In this paper, the objective is to deploy robots in facilities with high population density, such as indoor multi-purpose complexes, where the probability of child abduction is high. The purpose is to use these robots to provide assistance and aid in rapidly locating missing children. Through this approach, it is possible to prevent the occurrence of prolonged cases of missing children in the long term.

During the development of the robot in this study, there were errors in the multi-label classification process where the inferred values differed from the actual values. This is believed to have occurred due to biased learning for specific data during the data preprocessing stage, resulting in biased outcomes. Therefore, for achieving higher AI performance, the intention is to improve by collecting data more evenly across the entire dataset to ensure accurate results.

Additionally, in this study, as path planning is conducted in a dynamic environment, errors can arise in robot localization and mapping. To address this, in future research, the plan is to explore the integration of IMU sensors with ORB SLAM3.

Finally, experiments are planned in the future to objectively demonstrate the performance of the proposed system by examining how quickly and accurately missing children can be located depending on the utilization of the suggested system.

References

[1] J. Y. Park, "*871 명, 실종 1 년 넘도록 집에 못 간 아이들*(2022)," Accessed Date (2022-11-20), [Online]. Available: https://www.hani.co.kr/arti/society/health/1044385.html.

[2] W.-C. Choi and J.-Y. Na, "A study on

development of missing child prevention system based on spatial information," in *Proc. Korean Soc. for Geospatial Inf. Sci.*, pp. 351-353, Jeju, Republic of Korea, Nov. 2018.

[3] B. H. Park, B. S. Ko, H. W. Lee, Y. A. Cho, S. H. Park, and M. K. Choi, "Development of identical person identification technology by face recognition: Comparison of face area recognition in id card and face area recognition in real-time images," in *Proc. KIIT Conf.*, pp. 672-676, Jeju, Republic of Korea, Jun. 2023.

[4] D. S. Lee, J. S. Kim, S. C. Jeong, and S. K. Kwon, "Human height estimation by color deep learning and depth 3d conversion," *Applied Sci.*, vol. 10, no. 16, p. 5531, Aug. 2020. (https://doi.org/10.3390/app10165531)

[5] L. d. S. Nolasco, A. E. Lazzaretti, and B. M. Mulinari, "DeepDFML-NILM: A new cnn-based architecture for detection, feature extraction and multi-label classification in NILM signals," *IEEE Sensors J.*, vol. 22, no. 1, pp. 501-509, Jan. 2022. (https://doi.org/10.1109/JSEN.2021.3127322)

[6] Y. K. Tee and Y. C. Han, "Lidar-based 2d slam for mobile robot in an indoor environment : A review" in *Int. Conf.. Green Energy, Comput. and Sustainable Technol.* (GECOST), pp. 1-7, Miri, Malaysi, Sep. 2021. (https://doi.org/10.1109/GECOST52368.2021.9538731)

[7] Y. Li, J. Li, W. Zhou, Q. Yao, J. Nie, and X. Qi, "Robot path planning navigation for dense planting red jujube orchards based on the joint improved a* and dwa algorithms under laser slam," *Agriculture*, vol. 12, no. 9, p. 1445, Sep. 2022. (https://doi.org/10.3390/agriculture12091445)

[8] J.-Y. Kim, J.-H. Kim, and J.-W. Jeong, "Detection of the assembling positions of computer parts based on image classification and object detection," in *Proc. Korean Inst. Inf. Sci. and Eng.,* pp. 1448-1450, Online, Dec. 2020.

[9] Z. Dilkashbek and S.-i. Choi, "Lidar based slam and navigation of indoor mobile robot," in *Proc. The IEIE*, pp. 1080-1083, Incheon, Republic of Korea, Nov. 2021.

[10] J. K. Goyal and K. S. Nagla, "A new approach of path planning for mobile robots," in *ICACCI*, pp. 863-867, Delhi, India, Sep. 2014. (https://doi.org/10.1109/ICACCI.2014.6968200)

[11] Y. J. Ock, H. S. Kang, and J. M. Lee, "Modified orb-slam algorithm for precise indoor navigation of a mobile robot," *J. Korea Robotics Soc.*, vol. 15, no. 3, pp. 205-211, Sep. 2020. (https://doi.org/10.7746/jkros.2020.15.3.205)

Yeong Eun Choi

Feb. 2023 : B.S. degree, Kumoh National Institute of Technology

Mar. 2023~Current : M.S. student, Kumoh National Institute of Technology

<Research Interests> Autonomous driving, Deep learing

[ORCID:0009-0009-7380-3684]

Su Hyeon Kang

Feb. 2023 : B.S. degree, Kumoh National Institute of Technology

Mar. 2023~Current : M.S. student, Kumoh National Institute of Technology

<Research Interests> SLAM, Robot Localization, Navigation

[ORCID:0009-0005-2377-8725]

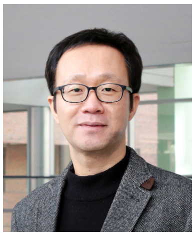
So Yeon Kim

Feb. 2023 : B.S. degree, Kumoh National Institute of Technology

<Research Interests> ICT technology, Deep learning

[ORCID:0009-0007-3134-8447]

Soo Young Shin



Feb. 1999 : B.S. degree, Seoul University

Feb. 2001 : M.S. degree, Seoul University

Mar. 2010~Current : Professor Kumoh National Institute of Technology, Gumi, Gyeongsangbuk-do, South Korea

<Research Interests> Wireless communications, Deep learning, Machine learning, Autonomous driving

[ORCID:0000-0002-2526-2395]