객체 탐지 기반 보안형 배달 로봇

문지우 [1], 손수민 [1], 위유정 [1], 유수민 [1], 하유빈 [1]
[1] 이화여자대학교 휴먼기계바이오공학부 학부생

mjw0111@ewhain.net, iamsoomin@ewhain.net,
yj2070@ewhain.net, tnals6739@ewhain.net, yubin01@ewhain.net

Object Detection-Based Secure Delivery Robot

Jiwoo Moon[1], Hyunji Seo[1], Soomin Son[1], Yujeong Wi[1], Soomin Yu[1], Yubin Ha[1]
[1]Dept. of Mechanical and Biomedical, EWHA Womans University

요 약

본 연구는 본인 인증 및 객체 탐지 알고리즘 기반의 보안 기능을 탑재한 배달 로봇을 제안한다. 기존 배달 로봇의 물품 도난 위험성 해결을 위한 보안 시스템 도입과 장애물 대처 및 강화학습 기반 경로계획 주행으로 보안성과 주행 안정성을 갖춘 배달 로봇을 고안하였다.

1. 서론

자율주행 배달 로봇은 비대면 서비스에 따른 심리적 피로도 감소와 배달 비용 절감에 따른 높은 경제성이 장점이다. 최근 실내외 자율주행 배달 로봇 수요의 증가로 관련 연구 및 상업화가 활발히 이루어지고 있다. 하지만 일반적으로 기존의 배달 로봇은 식당에서의 음식 서빙, 공장 내 물건 운반, 건물 내 길 안내 등 도난 우려가 적은 장소에서 제한적으로 사용된다. 본 연구는 이를 해결하고자 본인 인증 및 객체 탐지 기반 픽업 시스템으로 보안성을 강화한 배달 로봇을 제안한다. 또한 장애물 대처와 강화학습 기반 경로계획으로 주행 제어의 완전성을 더하고자 하였다.

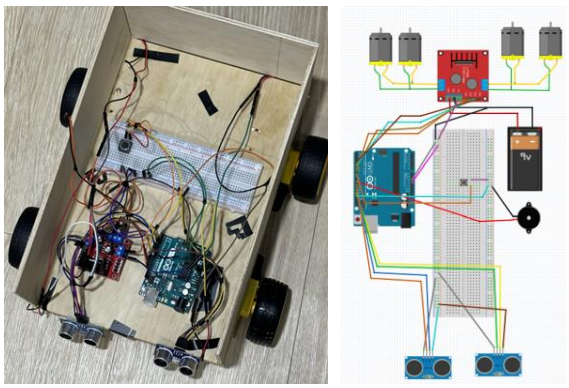
2. 본론

본 프로젝트는 배달 로봇의 외형 디자인, 주행 경로 설정 및 로봇 주행 제어, 본인 인증 픽업 시스템을 주요 기능으로 가진다. 따라서 프로젝트 참여자는 설계, 제어, 서비스 파트로 나뉘어 프로젝트를 수행하였다.

2-1. 설계

2.1.1 로봇 제작 수행 및 문제 해결

초기 계획은 12v DC 모터를 사용하고, 아두이노 보드 2 개를 사용하여 보드 간 통신을 하도록 제작할 예정이었다. 하지만 회로 내 과도한 발열과 그로 인한 잦은 부품 고장, 그리고 모터의 무게에 비해 낮은 토크 문제가 있어서 계획을 수정하였다.[1]
수정 후에는 소형 DC 모터로 변경하고, 아두이노 보드 1 개로 바퀴들을 병렬 연결함으로써 본체 무게를 경량화하였다. 그 결과, 최종 회로도 및 실제 완성본은 아래와 같다.



<사진 1, 2> 로봇 최종 내부 회로 및 회로 모식도

2.1.2. 기본 주행 코드 작성

스위치를 이용 motor on/off 기능을 수행하고, 거리 60 cm 이하의 조건에서 5 초간 모터 정지 후 다시 작동하는 장애물 감지를 위한 기본 주행 코드를 작성하여 실행하였다.

2-2. 제어

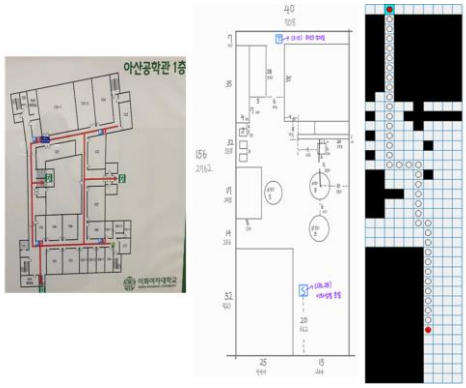
2.2.1. Q-learning 기반 로봇 주행 경로 결정

그리드 맵 기반 가상 주행 공간에서의 경로 탐색은 Matlab 환경에서 Q-learning 알고리즘으로 진행되었다. 최단 거리 탐색을 위해 모든 행동에 대한 보상을 -1 로 설정하고, 종료 상태 도달에 대한 보상을 +50 으로 정하였다.[2]

2.2.2. PWM 신호 활용 주행 모드 전환

주행 공간의 특성에 따라 3 가지의 주행 모드로 모터를 제어하였다. 첫째, '직선 주행' 모드는 PWM 신

호 특성과 주행의 안정성 및 주행 시간을 고려하여 153 을 출력한다. 둘째,’ 90 도 방향 전환’ 모드는 바퀴가 헛도는 상황을 방지하여 140 을 출력한다. 마지막,’ 경사로 주행’ 모드는 가속 억제를 위해 102 를 출력한다.



<사진 3> 로봇 주행 공간 및 예상 경로 맵

2-3. 서비스

2.3.1. 학생증 이름 및 학번 인식

Python 언어를 사용하여 OCR 알고리즘을 수행하였다. 7 자리 숫자는 학번, 2~4 글자 한글은 한국어 이름이라고 판별하였다. 이 과정에서 ‘총장’, ‘신한은행’ 등의 조건에는 해당하지만 이름이 아닌 단어들은 불용어 처리를 해 로직을 완성하였다. 본인 인증을 위한 이름과 학번을 모두 성공적으로 완료하였다.

2.3.2. 픽업 완료 판단

Python 언어를 바탕으로 OpenCV[3], YOLOv8[4], DeepSort[5] 알고리즘을 활용하여 물체 탐지 및 물체 위치 추적을 진행하였다. 프로젝트 초기에는 영상 화면을 3 개 구역으로 나눈 뒤, 각 구역에 대해 YOLO와 DeepSort 알고리즘을 독립적으로 적용했지만, 3 개의 모델을 동시에 실행하니 프레임 처리 속도와 정확도가 저하된다는 문제점이 있었다. 이를 화면 분할을 하지 않은 상태에서 객체의 위치를 탐지하고, 반환된 좌표를 이용해 물체의 중심점 좌표를 계산했다. 이 중심점 좌표과 영상의 3 개 구역 중 어디에 위치하는지를 바탕으로 객체의 구역을 추적했고, 중심점이 영상의 프레임에서 벗어나는 경우 픽업이 완료되었다고 판단했다.

3. 결론

본 프로젝트에서는 학생들의 효율적인 쉬는 시간 활용을 목적으로 매점-강의실 배달 로봇을 제작하였다. 예약 기반의 배달 로봇은 주행의 안정성, 강화학습 알고리즘에 따른 효율적인 주행 경로 탐색, 본인 인증 시스템 도입에 따른 보안성 강화를 특징으로 한다. 따라서, 쉬는 시간에 매점 이용자가 많아져 매점 이용에 불편함을 겪거나, 연속적인 강의로 강의실 이동에 쉬는 시간을 소비하게 되는 학생들을 이용 대상으로 한다. 본 프로젝트에서 제작한 배달로봇의 완성품은 아래와 같다.

<사진 4> 배달로봇 최종 완성본

Appendix

Algorithm 1 Object Tracking and Text Extraction using YOLO, DeepSort, and Azure OCR

```
Import Libraries: datetime, os, YOLO, cv2, DeepSort, ComputerVisionClient, etc.
Define Constants: CONFIDENCE_THRESHOLD, COLORS, SUBSCRIPTION_KEY, ENDPOINT_URL
Initialize:
    video_cap, YOLO model, DeepSort tracker
    computervision_client for Azure OCR
    Dictionaries and sets for tracking: tracks, counts, zones, etc.
Define Function: get_zone(x, y)
if x in first third of frame then return "zone1"
else if x in second third of frame then return "zone2"
else return "zone3"
end if
Define Function: extract_and_check_text(image_path)
    Send image to Azure OCR and check text patterns
    Validate extracted text with expected patterns
Create Directories: "tracked_images", "id_cards" if not exists
while True do
    Capture Frame: Read frame from video capture
    if frame not available then
        break
    end if
    Draw Zone Lines on Frame
    Run YOLO Model: Detect objects in frame
    Filter Detections: By confidence threshold
    Update Tracks: Using DeepSort tracker
    for each confirmed track do
        Increment frame counter
        if frame counter reaches threshold then
            Capture and Check ID Card Text
        end if
        Save ROI Images for New Tracks
        Update active tracks and object counts
    end for
    Check Cleared Zones:
    for each zone in counts do
        if zone previously occupied and now empty then
            Append to cleared_zones
        end if
    end for
    Identify Disappeared Tracks
    Update previous active tracks
    Display Frame with FPS
    if exit key is pressed then
        break
    end if
end while
Cleanup: Release video capture, close windows
Output: Print cleared zones and disappeared track IDs
```

<서비스 본인 인증 픽업 판단 Pseudo Code>

본 논문은 과학기술정보통신부 대학디지털교육역량 강화사업의 지원을 통해 수행한 ICT 멘토링 프로젝트 결과물입니다

참고문헌

[1] Peerzada. DC motor speed control through arduino and L298N motor driver using PID controller. International Journal of Electrical Engineering & Emerging Technology, 4(2), 21-24, 2021

[2] MathWorks, "Getting Started with Reinforcement Learning Toolbox,", 2024. [Online]. Available: https://kr.mathworks.com/help/reinforcement-learning/getting-started-with-reinforcement-learning-toolbox.html

[3] Gary Bradski, Adrian Kaehler, “Learning OpenCV: Computer Vision with the OpenCV Library”, O'Reilly Media, Inc., Sep. 2008

[4] A. Bewley, "Simple Online and Realtime Tracking with a Deep Association Metric," *arXiv preprint arXiv:1703.07402*, Mar. 2017.

[5] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.