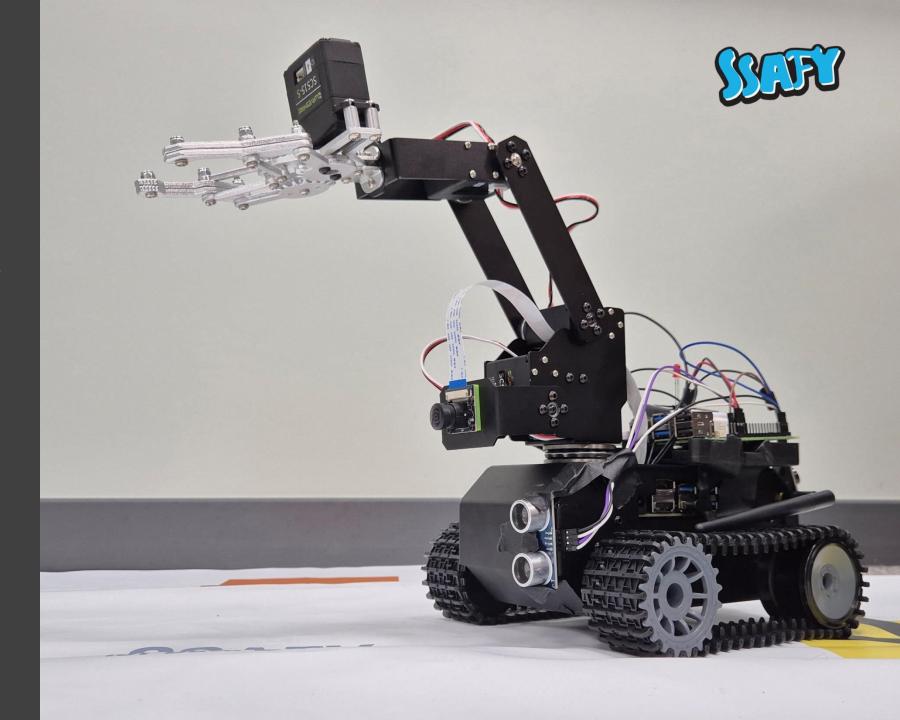
24.05.24 응용프로젝트 인공지능 AGV

SSAFY 11th Embedded 김남균 이동우



시연영상





https://www.youtube.com/watch?v=uZl3ZouLiyY

Contents







Using Al 생성형 Al 활용



요구사항과 우리의 솔루션 시스템 구조

우리 조의 차별화된 구현 고전 기법

보완점과 활용예시



Objectives

- Demand, Specification
- Solution
- System Outline
- Network structure
- GUI structure

Demand



SSAFY 출신 개발자 이동우 씨는, 삼성전자 평택캠퍼스에서 사람과 함께 일을할 수 있는 무인운반차량을 개발 프로젝트에 참여하게 되었습니다. AI 딥러닝모델을 활용한 "인공지능 무인운반차량(AGV)"을 제작해 보자.



Demand





차람이랑 같이 일하는 반자율 주행 AGV가 필요해요! 위험한 장비가 많으니깐 확실 하게 라인을 따라가고, 장애물 이 있으면 멈출 수 있게 해주세 요~~

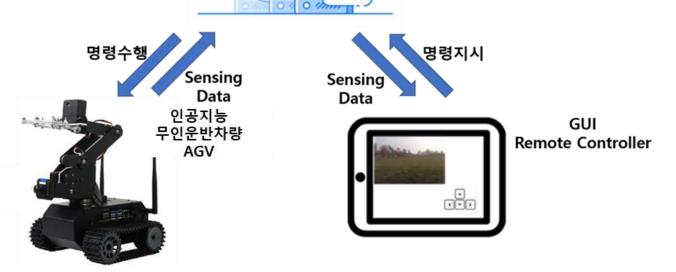


전장.. 또 야근 각이야 `

Project Goal



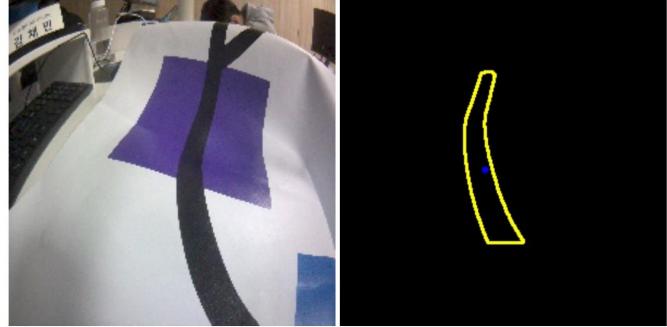
- 인공지능과 Computer Vision을 이용하여 움직이는 인공지능 무인운반차량 (AGV)
- 원격으로 인공지능 무인운반차량(AGV)을 제어하는 GUI 원격 제어기
- OpenAI API 를 이용한 로그데이터 분석





• 인공지능과 Computer Vision을 이용하여 움직이는 인공지능 무인운반차량 (AGV)

• OpenCV 영상처리를 토하 제어



x_moment: 133, flag: 0, mid: 112, ROI shape: (224, 224, 3)

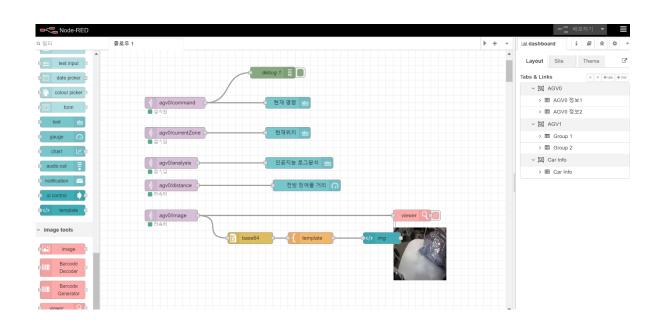


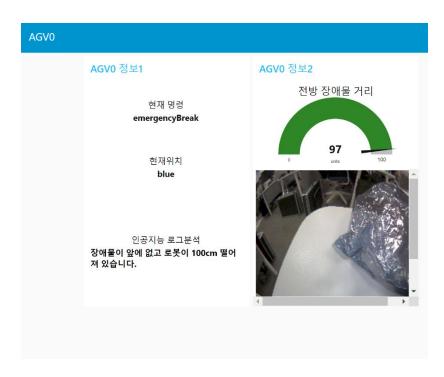
- 원격으로 인공지능 무인운반차량(AGV)을 제어하는 GUI 원격 제어기
- Raspberry Pi 5와 Qt를 사용한 GUI 원격 제어기





• 서버리스 기반으로 인공지능 무인운반차량(AGV)과 GUI 원격 제어기의 통신을 담당하는 Cloud Database







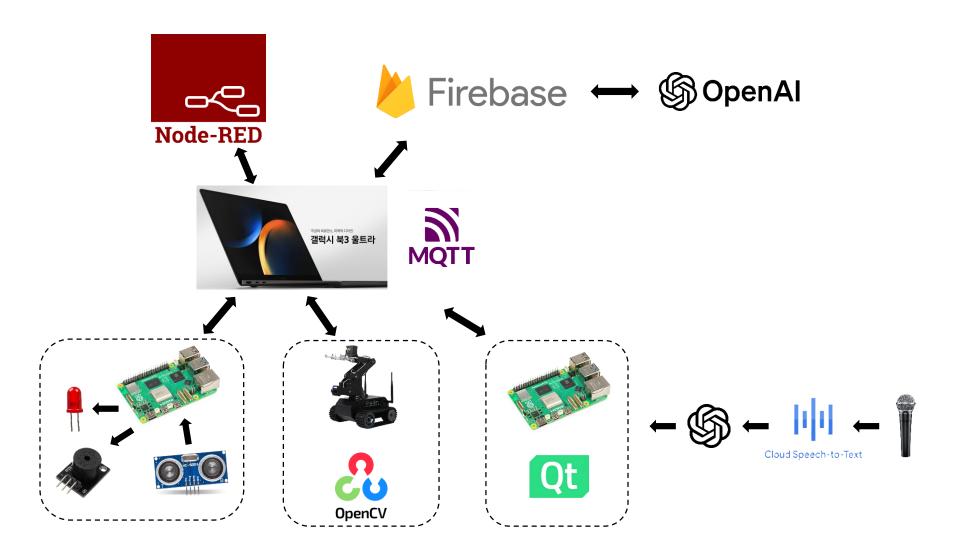
• OpenAI API 를 이용한 로그데이터 분석





Specification – System Outline



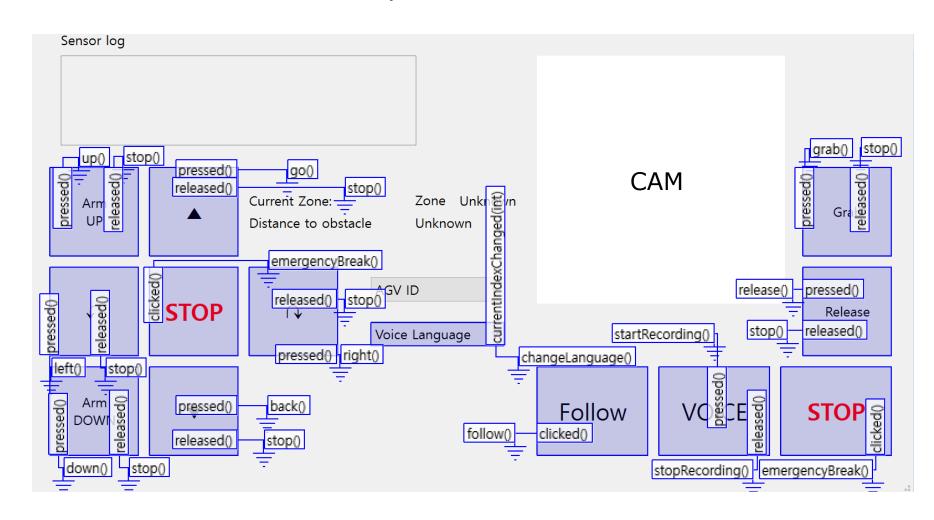


Specification - GUI



• RC 조종기와 같이 버튼 누르는 동안 동작, 떼면 정지

- 실시간 영상
- 실시간 로그
- 초음파 센서
- 현재 위치
- 언어 선택
- AGV 선택





Contribution

- Why Classical?
- Classical Computer Vision
- Classical Control PID controller

Why Classical?



우리의 목표

- 한정된 라인 위에서 Line tracing
- Working area 판별
- 물체 object detection

한정된 개발기간

- 1주일 남짓.. 🦃
- object detection 대신 수동제어
- 변화가 적은 실내 환경!



-> 딥러닝이 아닌, 고전적인 영상처리와 제어기법을 고려할 수 있다.

Why Classical?



고전 Computer Viston 장점

- **논리적으로 100% 제어** (조건 만족 시)
 - -> Rule based model vs AI based model 차이점
- 학습과정 없이 바로 현장 투입 가능
- 자원 효율성: 적은 시간복잡도, 적은 컴퓨팅 자원
- 딥러닝도 이미지 전처리 과정이 필수



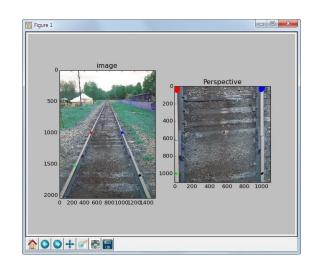
돌아왔구나.. 고전파..

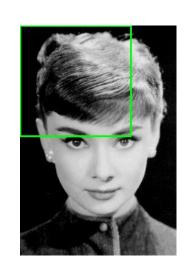
고전 Computer Viston 단점

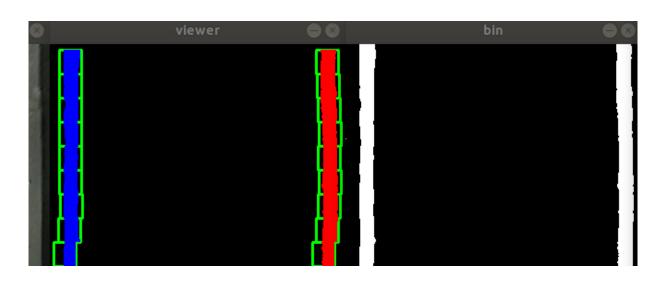
• Robust하지 않음 ex) 실외 환경



차선인식에 쓰이는 다양한 고전 Computer Viston 기법들







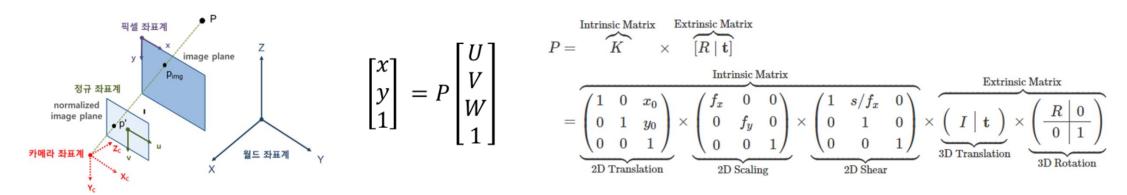
SIFT나 ORB, SURF 같은 특징점 추출, Homography, 슬라이딩 윈도우 같은 다양한 기법

우리가 쓴 고전 Computer Viston

1주일 안에 구현 가능한 것들로 만들었다.



Calibration – Intrinsic parameter, distortion coefficients 필요



여러 장의 체크보드 이미지와 OpenCV API를 통해 초점거리, 스케일, 원점 이동 정도, 왜곡계수를 알 수 있다!

우리의 과정



체크보드 이미지 약 20장 촬영 후 opencv를 활용해 컬리브레이션 결과로 나온 파라미터들을 .npy로 저장

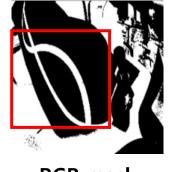
```
camera_matrix [[143.1741169 0. 144.52728132]
  [ 0. 189.22983068 142.75844376]
  [ 0. 0. 1. ]]
dist_coeffs [[-3.41473631e-01 1.67474878e-01 4.81115840e-05 1.18391685e-03
  -5.85043848e-02]]
```



ROI(Region Of Interesting), RGB masking







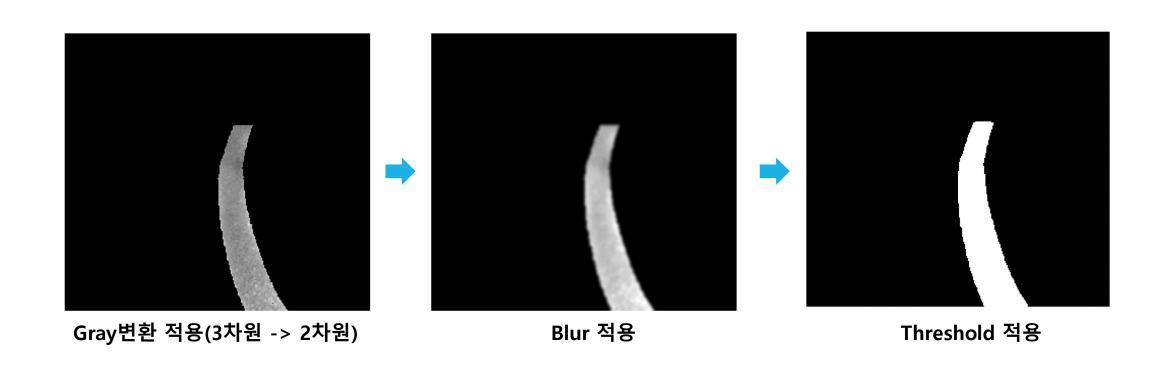
※ 검은 라인의 경우 HSV마스킹보다 RGB마스킹이 더 뚜렷하게 검출되어 라인만 특수하게 RGB마스킹 적용

HSV mask

RGB mask



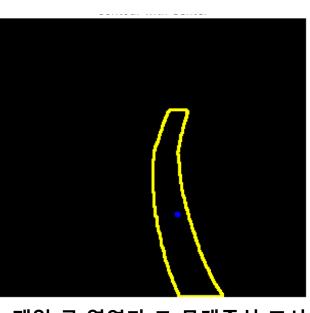
GrayScale, Gaussian Blur, Threshold





Counter의 MAX, moment 표시

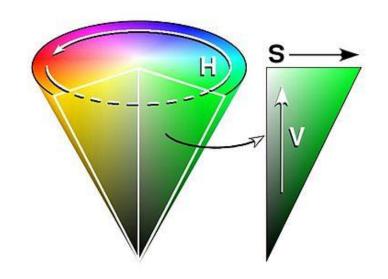




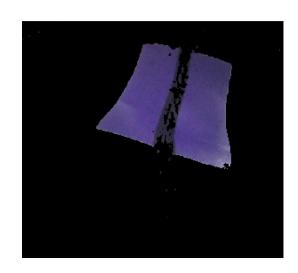
제일 큰 영역과 그 무게중심 표시



Working Area recognition – HSV masking





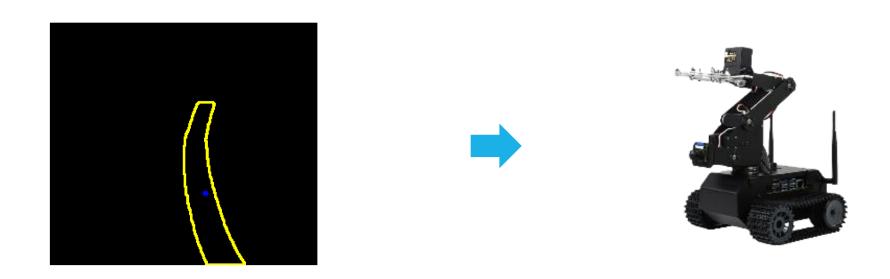


H만 원하는 색상 영역으로 마스킹, S와 V는 threshold를 주어 마스킹
-> 흰색, 검은색 탐지 피함 유도, 유연성

Conventional Control – PID controller



자동제어가 필요하다!



자동 라인 트레이싱을 위해 요구되는 사항: 카메라로 촬영한 라인의 상태에 따라 AGV의 DC모터 제어 필요

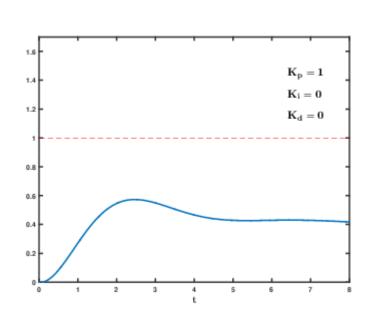
"무게중심이 가운데로 가도록 모터를 제어하기"

Conventional Control – PID controller

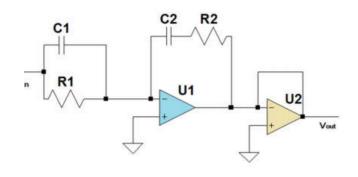


고전 제어 – PID Controller란?

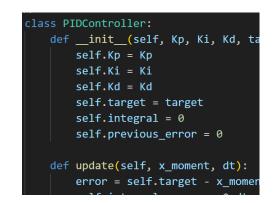
피드백을 활용한 비례, 적분, 미분기를 달아 원하는 값으로 제어 파라미터 조정으로 원하는 제어기 제작 가능 – 다양한 응용, 안정성, 오차보정







디지털 세계







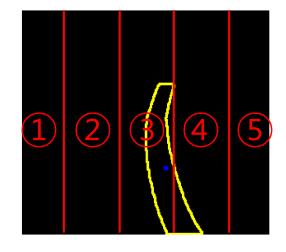
우리는 이산 세계! 클래스로 구현 가능

Conventional Control – PID controller



AGV에서 PID Controller 적용

기존에 고안한 방식



PID 제어 이용

```
# PID 컨트롤러 초기화
Kp = 0.005
Ki = 0.0001
Kd = 0.0005
dt = 0.02
target = 112 # 중앙값으로 설정
```

pid = PIDController(Kp, Ki, Kd, target)
left_speed, right_speed = pid_control(x_moment, pid, dt)

무게중심의 위치에 따라 5단계로 나눠 제어하자!

오차 보정 지속 업데이트, 보다 스무스하게 제어





- google speech to text API, openai API on Raspberry Pi 5
- MQTT Network
- Openai API data log feedback



google speech to text API & openAl API on Raspberry Pi 5





근무자가 Voice 버튼을 눌러 녹음하고, 떼서 녹음을 종료, 문자열 리스트로 변환

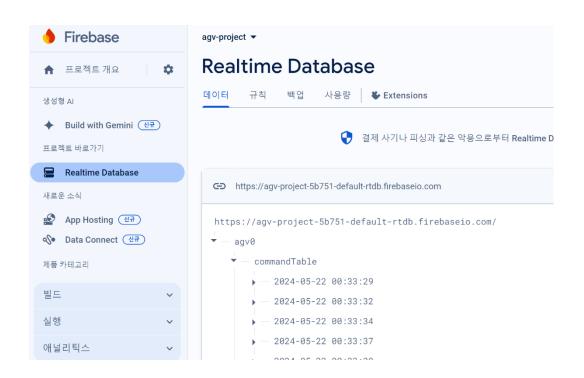
SEFT

Openai API data log feedback



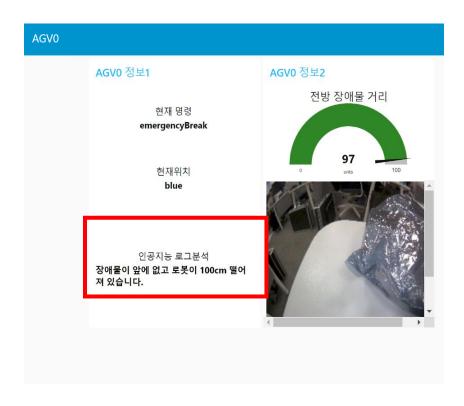














Additional

- 더 보완할 점, 발전가능한 점
- 활용예시

더 보완할 점, 발전가능한 점



기대사항

- 음성제어와 수동모드 모두 제공, 근무자 근무 효율과 편의성 증대
- 다양한 AGV, 언어 지원

더 보완할 점

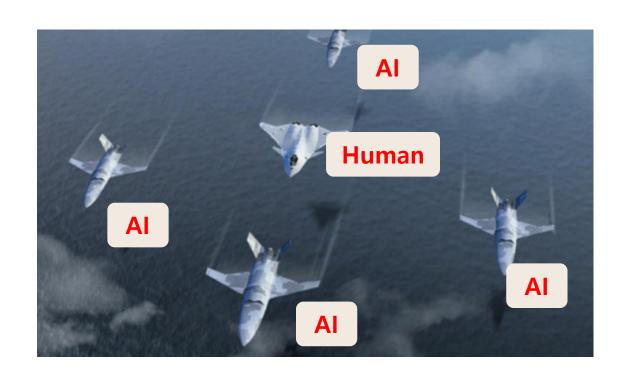
- 물체인식에서 수동 -> Object detection 딥러닝 기술 적용 가능
- 경로가 비효율적 -> dijkstra 등 경로생성 알고리즘을 활용 가능

발전 가능한 점

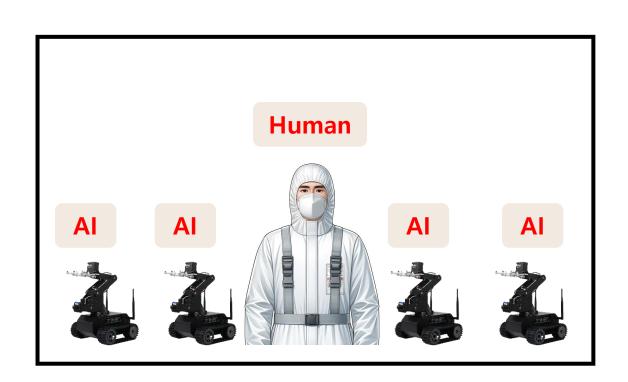
- ROS를 사용한다면 더 강력한 프레임워크 사용 가능
- 버튼 음성 제어 -> "하이 빅스비" 로 기기호출로 업그레이드 가능
- OpenCV 4.1.1 (GPU x) -> GPU 사용 가능 버전 사용 가능
- MQTTT 영상 전송시 다른 통신방식을 사용한다면 네트워크 부담 감소 가능
- 장애물 인식 시 Potential field 이용해 효과적인 회피 가능



활용예시



6세대 전투기를 아시나요?



스마트 팩토리에도 적용!



다시 평택 공장으로

상황예시: 에칭공정은 시간이 많이 걸린다. 이슈가 발생해 삼성전자의 설비 엔지니어가 ASML

에서 출장 나온 장비사와 얘기를 나눠야 하는 상황이다.





여기서 장비사랑 같이 계속 지켜봐야하는데.. 공구함 놓고왔네

> 1번은 포토공정 라인 가서 공 구함 가져오고 나머지는 여기서 대기해.



Thank you for watching!

감사합니다.

