

2024/02/21 회의록

팀명 : 무인계획

회의 참석자 : 신재환, 최승렬, 류은환

회의 일자 : 2024-02-21(18:30~21:30)

회의 장소 : 취창업 라운지

회의 내용

주제 구현 방법 및 근거 파트 초안 작성

주제 구현 방법 및 근거 파트 부분에서 소프트웨어 부분을 여러 모듈로 나누고 각 모듈별로 기능과 역할을 단계별로 작성함.

Camera Module

- 30fps로 이미지 capture 후 Video Save Module과 Perception Moudle input으로 넣어줌.

GPS Module

- time, 위도, 경도 정보를 받아옴

Perception Module

- Camera Module에서 넘어온 데이터를 YOLOv8을 이용해 차량 object detection, 차량마다 int형 라벨링값을 줌.
- Bounding box crop 후 다음과 같은 단계를 밟음.
 - 회색조 변환
 - 이미지를 회색조로 변환하면서 번호판 인식에 필요 없는 컬러 정보를 줄여 연산속도 향상
 - 이미지 이진화
 - 회색조로 변환된 이미지에서 윤곽선 검출을 위해 threshold를 적용하여 threshold 보다 낮은 픽셀은 '0'으로 만들어 검은 픽셀로 만들고 높은 픽셀은 '255'로 만들어 흰 픽셀로 만들어서 완전한 흑백 이미지로 변환함.
 - 윤곽선 검출
 - 이진화된 이미지에서 불필요한 잡음을 제거하기 위해 가우시안 필터를 적용하며 Open CV 라이브러리의 cv2.findContours() 함수를 사용하여 윤곽선을 검출함.
 - 번호판 영역 검출
 - 윤곽선이 검출된 이미지에서 cv2.rectangle() 함수를 사용하여 모든 윤곽선의 테두리를 그리고 번호판 숫자의 사이즈에 근접한 테두리만 남김. 번호판의 포맷은 4개의 숫자가 연속 되므로 4개의 번호가 연속되는 지점의 좌표를 저장함. 해당 좌표를 기준으로 이미지를 똑바로 세우고 좌표 영역만 검출됨.
 - 번호판 인식
 - 검출된 번호판 영역 이미지에서 가우시안 필터를 적용해 다시 한번 잡음을 제거하고 Tesseract OCR 엔진을 사용하여 번호판을 인식함. 조사 결과 국내 모든 차량의 번호판에는 1자리 국문이 포함되어 있으며 이 1자리 국문의 경우 비사업용 32가지, 사업용 4가지, 택배용 1가지, 대여용 3가지, 군용 5가지 등 정해진 국문이 존재함. 따라서 OCR결과를 format과 맞는지 비교 후 detection된 차량과 data type char로 labeling 추가.
 - format과 맞고 번호판 라벨링이 되면 원본 영상으로 돌아와 capture
 - 캡처된 사진에 이름을 차량번호로 지정하고 캡처 폴더에 저장

- OpenCV를 이용해 차선을 인식함 (2가지 방법이 있음)

1. Canny + Hough

a. 회색조 변환

- i. 이미지를 회색조로 변환하면서 차선 인식에 필요 없는 컬러 정보를 줄여 연산속도 향상

b. 이미지 이진화

- i. 회색조로 변환된 이미지에서 윤곽선 검출을 위해 threshold를 적용하여 threshold 보다 낮은 픽셀은 '0'으로 만들어 검은 픽셀로 만들고 높은 픽셀은 '255'로 만들어 흰 픽셀로 만들어서 완전한 흑백 이미지로 변환함

c. Canny Edge Detection으로 에지를 추출

- i. Gaussian Filter로 이미지의 잡음을 제거
- ii. Sobel Filter를 사용하여 Gradient의 크기를 구한다
- iii. Non-maximum suppression을 적용하여 경계선 검출기에서 거짓 반응을 제거
- iv. 경계선으로써 가능성 있는 픽셀을 골라내기 위해 double threshold 방식을 적용한다
- v. 앞서 double threshold 방식에서 maxVal을 넘은 부분을 strong edge, minVal과 maxVal 사이의 부분을 weak edge로 설정하여 strong edge와 연결되어 있는 weak edge를 edge로 판단하고 그렇지 않는 부분은 제거. (Hysteresis thresholding)

d. Hough 변환

- i. Hough 변환을 사용하여 검출된 에지 영상에서의 직선 성분을 추출
- ii. OpenCV의 HoughLinesP 함수를 사용하여 Hough 변환
- iii. 이때 최소 검출 길이를 길게 설정하여 차선만 검출
- iv. 점선과 실선은 길이로 구분 (점선+실선 복선일 경우 점선도 구분해야 함)

2. Perspective Transform (Bird Eye View)

a. Camera Calibration

- i. OpenCV의 undistort 함수를 사용해 왜곡된 이미지를 보정

b. Perspective Transform

- i. 원본 이미지와 변환 후 데이터 간의 관계를 설명하는 행렬식을 계산 (OpenCV의 getPerspectiveTransform 함수를 사용)
- ii. 행렬식을 이용하여 원본 이미지를 원하는 이미지의 크기와 위치로 변환 (OpenCV의 warpPerspective 함수를 사용)

c. Color Filtering(HLS, LAB color space)

- i. 채널 컬러 변경 (OpenCV의 cvtColor 함수를 사용)
- ii. 남겨둘 채널의 픽셀값만 가져오기(LAB에서 B채널(황색), HLS에서 L채널(백색))

d. 픽셀 값 정규화 및 이미지 픽셀 합치기

- i. 이미지 픽셀의 최댓값 구하기
- ii. (255/최댓값)의 비율로 이미지 전체 픽셀을 정규화
- iii. 특정 경계값을 설정하여, thresholding 적용
- iv. 두 이미지의 픽셀 데이터를 하나의 이미지로 병합(B채널과 L채널)

e. Window Search

- i. 직사각형 박스를 탐색 영역으로 설정
- ii. 높이 방향의 픽셀 수가 가장 많은 컬럼 위치를 탐색의 초기 기준값으로 설정
- iii. 직사각형을 높이 방향으로 전개시켜, 다음 단계의 직사각형을 탐색 및 영역 내의 픽셀 위치 데이터 획득

• 백색 실선, 황색 실선 구분

- 실선 영역 색상 확인

• 라벨링된 차량이 화면 밖으로 나가면 캡처 폴더에 있는 해당 사진을 지움.

• input : Camera

• output : 차량 바운딩 박스 좌표값, Open Cv로 검출된 차선 좌표값, labeling 값

Decision Module

- Perception Module에서 넘어온 차량의 바운딩 박스 좌표값, 차선 좌표값, data type char라벨링값을 이용해 차량 바운딩박스 좌표 중앙값이 차선 좌표값이랑 겹치면 즉 차량 전폭 반이상이 차선을 넘어가면 차선위반을 했다고 판단함.
- 차선을 넘어갔다고 판단하면 Alarm Module에게 알림
- 만약 Perception Module에서 차량의 바운딩박스 좌표값, 차선 좌표값은 잘 넘어오는데 data type char형 labeling값이 NULL이고, 차량 전폭 반이상이 차선을 넘어가면 data type char labeling 값이 들어올 때까지 int형 labeling으로 대기 후 char labeling이 넘어오면 차선위반했다고 판단함.
- 차량의 바운딩박스 좌표값의 중간 값과 전 프레임에서의 차량의 바운딩박스 좌표값의 중간 값 사이에 차선 영역이 존재한다면 차량 전폭 반 이상이 차선을 넘어갔다고 판단
- input: 차량 바운딩 박스 좌표값, OpenCV로 검출된 차선 좌표값, labeling 값
- output: event

Alarm Module

- Decision Module에서 차선위반했다는 event가 발생해 Alarm Module input으로 들어오면 부저를 울려 사용자한테 알려줌.

Video Save Module

- 1분 단위로 짜르면서 상시녹화
- buffer size을 1분단위 (30fps * 60)로 설정
- event가 들어오면 30초 후 buffer을 이벤트 폴더에 라벨링값을 제목으로 하는 폴더 생성 후 저장
- event가 발생한 라벨링 데이터 저장
- char형식 라벨링 데이터가 있을 경우 캡처 폴더에 있는 해당 차량 사진을 복사해 이벤트 폴더에 있는 폴더에 저장
- char형식 라벨링 데이터가 없을 경우 char형식 라벨링 데이터가 생길 때 까지 대기
- char형식 라벨링 데이터가 없고 라벨링 데이터가 더 이상 안 들어올 경우 이벤트 폴더에 있는 폴더 삭제
- event가 발생한 GPS Module output(시간, 위도, 경도) 이벤트 폴더에 있는 폴더에 저장
- 영상 녹화 시 타임코드(년·월·일·시·분) 삽입
- input: 이미지 데이터, event
- output: 상시 녹화, 이벤트 녹화

AP Module

- Wifi 통신을 하기 위한 Module

Report Module (모바일 앱)

- 스트리밍 영상 재생 기능
- 영상 저장 기능
- 신고 기능
 1. 사용자가 저장된 위반 영상을 확인 (스트리밍)
 2. 사용자가 신고할 위반 영상을 선택 (영상 저장 및 신고 폼(안전신문고)으로 이동)
 3. 신고 폼 자동 입력 (신고유형, 신고발생지역, 제목, 신고내용, 차량번호, 발생일자, 발생시각)
 4. 사용자가 확인 후 신고

Yolo와 Opencv에 관해서 여러 버전의 성능을 비교해보고, 영상을 확인해보며 추가 조사를 진행함.

[참고자료]

인공지능 주차장 차단기 만들기 - 차량 + 번호판 인식

* 가장 쉽게 모터를 움직이는 방법 DynamiKontrol

<https://dk.m47rix.com/>

 <https://www.youtube.com/watch?v=mmFrZV1iH0c>




미디어파이프 YOLO 비교

[Pose Estimation] YOLOv5, MediaPipe로 Multi Pose 구현 시도해보기

Media pipe Media pipe는 Google에서 제작한 Machine Learning Solution으로 얼굴추적, 손추적, 객체 인식과 같은 다양한 기능들을 제공한다. 또한, 사람의 자세 인식 Pose 추적 이 가능하지만 오직 한 사람의 Pose를 추정하는 기능을 제공한다. 한 사람의 Pose를 추적하는 코드를 구현하기를 원한다면, 이 전 게시글인 아래의 포스팅을 확인하길 바란다.


<https://olrlabt.tistory.com/50> [Pose Estimation] MediaPipe Pose / 미디어 파이프로 사람 자세 인식 JavaScript 구현하기 Media pipe Media pipe는 Google에서 제작한

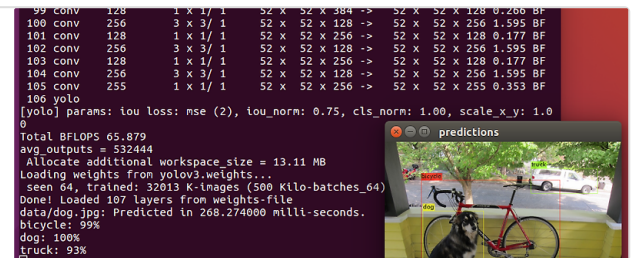
 <https://olrlabt.tistory.com/51>

Yolo 설치가이드

[8] Ubuntu에 YOLOv3 설치하기

YOLOv3 설치하기 1) 터미널을 켜 후 경로를 이동한다. 아래의 경로로 이동을 하는데 저는 opencv-4.0.1버전이어서 저 령게 이동했습니다. cd opencv/opencv-4.0.1/build 2) git 설치 후 다운받기 darknet은 pjreddie와 alexeyab 두 가 지의 버전이 있습니다. 저는 pjreddie버전을 이용하였습니다. 라즈베리파이에 설치하시는 분들은 alexeyab 버전을 이용

 <https://pmin-a.tistory.com/13>



<https://www.youtube.com/watch?v=pXX3eu3h5tc>

<https://www.youtube.com/watch?v=aWMkgbLWr-c>

<https://www.youtube.com/watch?v=RVz8r-O0QUs>

젯슨나노 yolov4 tiny

<https://www.youtube.com/watch?v=bsv28PuKiIM>

첨부 자료들 추가 조사

- 주제를 선정한 이유 및 근거 중 선정 배경에서 사용되는 교통 법규위반별 교통사고 건수와 가해자 법규 위반별 현황, 교통 사고 연도별 현황, 국민 차량 보유량이나 연도별 차량 등록 대수 등의 통계 비율 자료를 경찰청 통계자료나 도로교통공단 교통사고분석시스템(TAAS)를 참고하여 조사함.
- 스마트 국민제보 앱의 리뷰를 조사하여 앱의 신고 절차의 복잡성과 번거로움을 확인함.
- 교통 법규 위반에 대한 경각심이 낮음을 보여줄 수 있는 시민들의 인터뷰들을 찾아봄.(ex 교통법규를 위반했지만 이 정도는 봐달라는 내용이 담긴 자료)
- 경찰 인력의 부족함과 교통단속 경찰의 비율이 낮음을 보여주는 기사와 통계자료를 조사함.
- 교통 법규 위반 신고 건수 중 각 위반 항목 별 신고 건수 통계 자료에 대한 자료를 요청하고자 경찰청 02-3150-0645(교통조사계), 02-3150-2852(교통 안전계 단속 담당), 02-3150-2632(정보공개담당), 02-3150-0658(교통안전계 통계 담당)에 전화를 걸어 해당 내용에 대한 문의를 진행함.

다음 회의 일정

- 2024/02/23 (18:00)에 디스코드 음성채널에서 회의를 진행하기로 함
- 주제 신청서를 수정 및 보완하고 이어서 계속 작성하기로 함.