

KIM HO GWON

DEVELOPER PORTFOLIO

이름	김 호 권
생년월일	1992. 09. 02
이메일	gisf456@gmail.com
전화번호	010-3935-8229
주소	대전광역시 중구 유천동

CONTENTS

01. 자기소개

02. 경력

03. JAVA를 이용한 도서관
웹페이지 제작

04. Python을 이용한 이미지 감지
웹페이지 제작

05. AI기반 자율주행 솔루션
개발 플랫폼

KIM HO GWON



ABOUT ME

birth: 1992.09.02

address: 대전 중구 유천동

tel: 010-3935-8229

email: gisf456@gmail.com

kakaolD: gisf456

EDUCATION

2011년 대구가톨릭대 도서관학과 학사 입학

2017년 대구가톨릭대 도서관학과 학사 졸업

2023년 AI(인공지능) SW 개발자 과정 이수

EXPERIENCE

2017~2018 경기화성교육도서관 근무

2019~2022 (주)크레아소프트 근무

2023.1~2023.6 대전전자도서관 근무

SKILL

Front-End: HTML, CSS, Javascript, jQuery

Back-End: soring framework,

myBatis, Flask, migrate, SQLAlchemy

Language: Java, Python

AI: 머신러닝, 딥러닝

LICENSE

ITQ 한글파워포인트(2011.06.16 취득)

ITQ 한글엑셀(2014.05.29 취득)

MOS MASTER(2014.07.09 취득)

워드 프로세서(2016.11.21 취득)

Project 01.

JAVA를 이용한 도서관 웹페이지 제작

About project

이용자들에게 도서 자료에 대한 정보를 제공하고 공지사항, FAQ등의 정보와 원하는 도서를 신청할 수 있는 기능을 제공해 줄 수 있는 웹페이지를 만듦.

Project 01.

JAVA를 이용한 도서관 웹사이트 제작

공지사항

공지사항

제목 ▼ 검색어를 입력하세요.

Q

번호	제목	작성자	등록일	조회수
11	공지사항발행	mini	2024-01-17	0
9	주간추가도서목록입니다	mini	2024-01-17	0

공지사항등록

« ‹ › »

작업 기간	2023.12~2024.01
인력 구성	5명
프로젝트 목적	개발자과정의 중간 프로젝트로 JAVA의 숙련도 향상을 위하여 팀으로 진행함
프로젝트 내용	이용자들에게 도서 자료에 대한 정보를 제공할 수 있으며 원하는 도서를 신청할 수 있는 기능을 제공해 줄 수 있는 웹 서비스
주요 업무 및 상세 역할	1) 프로젝트 총괄 관리 업무 2) DB의 공지사항 정의 3) 공지사항 게시판 기능 구현까지 전담하여 업무 수행
사용언어 및 개발 환경	Java, Oracle, apache Tomcat, HTML
참고 자료	https://github.com/KimHoGwon/first_project.git

JAVA를 이용한 도서관 웹페이지 제작

답변 코드

[illegible]

```
<div class="card card-outline card-info">  
    <div class="card-header">  
        <h4>글등록</h4>  
    </div>  
    <div class="card-body">  
        <form role="form" method="post" action="regist.do" name="registForm">  
            <div class="form-group">  
                <input type="text" id="n_writer" title="작성자" readonly  
                    name="writer" class="form-control nullable" value="{loginUser.username }">  
                <input type="hidden" id="memberid" name="memberid" value="{loginUser.memberid }">  
            </div>  
            <div class="form-group">  
                <label for="n_title">제목</label>  
                <input type="text" id="n_title" name='n_title' class="form-control" placeholder="제목을 쓰세요">  
            </div>  
            <div class="form-group">  
                <label for="n_content">내용</label>  
                <textarea class="form-control" name="n_content" id="n_content" rows="15" placeholder="공지사항을 입력하세요"></textarea>  
            </div>  
        </form>  
    </div>  
    <div class="card-footer">  
        <button type="button" class="btn btn-primary" id="registBtn" onclick="regist_go();">등록</button>  
        &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <button type="button" class="btn" id="cancelBtn" onclick="CloseWindow();">취소</button>  
    </div>
```

```
<div class="card-body">
  <div class="form-group col-sm-12">
    <label for="n_title">제 목</label>
    <input type="text" class="form-control" id="n_title"
      value="${notice.n_title}" readonly />
  </div>
  <div class="col-sm-12 row">
    <div class="form-group col-sm-4">
      <label for="n_writer">작성자</label>
      <input type="text" class="form-control" id="n_writer"
        value="${notice.n_writer}" readonly />
    </div>
    <div class="form-group col-sm-4">
      <label for="n_date">작성일</label>
      <input type="text" class="form-control" id="n_date"
        value="<fmt:formatDate value="${notice.n_date}" pattern="yyyy-MM-dd" />" readonly />
    </div>
    <div class="form-group col-sm-4">
      <label for="n_hit">조회수</label>
      <input type="text" class="form-control" id="n_hit" value="${notice.n_hit}" readonly />
    </div>
  </div>
  <div class="form-group col-sm-12">
    <style>
      .detail-content {
        white-space: pre-wrap;
      }
    </style>
    <label for="n_content">내 용</label>
    <div class="detail-content">${notice.n_content }</div>
  </div>
</div>
```

- **기능 소개**
 - 공지사항 등록
 - 공지사항 상세 페이지
- **작업 내용**
 - 데이터베이스와 연동하여 공지사항 등록
 - 데이터베이스에 있는 등록된 공지사항을 상세페이지에

수정 코드

```
<div class="card-header">
    <h3>공지사항 수정</h3>
</div>
<div class="card-body">
    <form role="form" method="post" action="/modify.do" name="modifyForm">
        <input type="hidden" name="_n_number" value="{notice.n_number}" />

        <div class="form-group">
            <label for="n_writer">작성자</label>
            <input type="text" id="n_writer" readonly name="n_writer" class="form-control" value="{notice.n_writer}">
        </div>
        <div class="form-group">
            <label for="n_title">제목</label>
            <input type="text" id="n_title" value="{notice.n_title}" name='n_title' class="form-control" placeholder="제목을 쓰세요">
        </div>
        <div class="form-group">
            <label for="n_content">내용</label>
            <textarea id="n_content" class="form-control" rows="5" name="n_content">{notice.n_content}</textarea>
        </div>
    </form>
</div>
<div class="card-footer">
    <button type="button" class="btn btn-warning" id="modifyBtn" onclick="modify_submit();">수정</button>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
    <button type="button" class="btn" id="cancelBtn" onclick="history.go(-1);">취소</button>

```

삭제 코드

```
@GetMapping("/remove")
public ModelAndView remove(int n_number, ModelAndView mnv) throws Exception {
    String url = "/notice/remove_success";

    ModelAndView notice = new ModelAndView();
    noticeService.remove(n_number);

    mnv.setViewName(url);
    return mnv;
}

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<script>
    alert("${param.pno}공지가 삭제되었습니다.");
    window.location.href = 'list.do';
    window.opener.location.reload();
</script>
```

- 공지사항 수정
- 공지사항 삭제

- 데이터베이스에 있는 등록된 공지사항을 수정하여 데이터베이스에 전송
- 데이터베이스에 있는 공지사항을 삭제

Project 02.

Python을 이용한 이미지 감지 웹페이지 제작

About project

이용자들에게 이미지 감지를 할 수 있는 기능을 제공해 줄 수 있으며, 공지사항을 통해 홈페이지에 대한 정보를 제공할 수 있고 운영자는 회원 관리를 할 수 있는 웹페이지를 만들

Project 02.


Python을 이용한 이미지 감지 웹페이지 제작

회리예스 아사

공지사항 회원일람 감지 김호권123 로그아웃

김호권의 홈페이지 입니다.

이 홈페이지는 Flask 수업 활동으로 만들어졌습니다.
공지사항, 회원목록, 감지 3개의 메뉴를 가지고 있습니다.
이미지의 고양이는 귀엽습니다.
감사합니다.



작업 기간	2024.02.05~2024.02.29
인력 구성	1명
프로젝트 목적	개발자과정의 중간 프로젝트로 Python의 숙련도 향상을 위하여 Flask를 이용하여 진행함
프로젝트 내용	공지사항 게시판과 이미지 감지를 할 수 있는 기능을 가진 게시판 제공해 줄 수 있으며, 운영자는 회원 관리를 할 수 있는 웹 서비스
주요 업무 및 상세 역할	1) 메인화면 구성 2) 공지사항 게시판 구현 3) 회원 관련 기능 구현 4) 이미지 감지 구현
사용언어 및 개발 환경	Python, Flask, HTML
참고 자료	https://github.com/KimHoGwon/flask.git

로그인 코드

공기사정		
상대습도		
절대		
공기중 수증기 양과 해당압력		
계량단위	계량단위	계량단위
mm	2004.12.17	h
계량		
기압은 1013h		
기압계는 1013h는 1013h		
기압계는 1013h는 1013h		
기압계는 1013h는 1013h		

- 기능 소개
 - 로그인
 - 회원가입
- 작업 내용
 - 이용자에게 사용자명, 메일주소, 비밀번호를 받고 데이터베이스에 저장하여 회원 가입
 - 데이터베이스에 있는 데이터를 비교하여 로그인 구현

로그인 코드

```
<form method="post" action="{{ url_for('auth.login') }}" onsubmit="return submit();">
    {% for message in get_flashed_messages() %}
    <span class="dt-auth-flash">{{ message }}</span>
    {% endfor %}
    {{ form.csrf_token }}
    <input type="hidden" name="next" value="{{next}}" />
    {{ form.email(class="form-control dt-auth-input",placeholder="메일 주소") }}
    {{ form.password(class="form-control dt-auth-input",placeholder="비밀번호") }}
    {{ form.submit(class="btn btn-md btn-primary btn-block dt-auth-btn") }}
```

회원가입 코드

```
<form method="post" action="{% url_for('auth.signup', next=request.args.get('next')) %}"
      class="form-signin">
  {{ form.csrf_token }}
  {% for message in get_flashed_messages() %}
  <div class="dt-auth-flash">{{ message }}</div>
  {% endfor %}

  {{ form.username(size=30, class="form-control dt-auth-input", placeholder="사용자명") }}
  {{ form.email(class="form-control dt-auth-input",placeholder="메일 주소") }}
  {{ form.password(class="form-control dt-auth-input",placeholder="비밀번호") }}
  {{ form.submit(class="btn btn-md btn-primary btn-block dt-auth-btn") }}
```

회원 편집, 비밀번호 확인

회원 편집

127.0.0.1:5000/408

회원정보를 입력하세요

회원명

이메일

비밀번호

저장

취소

삭제

비밀번호 확인

현재 패스워드 전송

비밀번호 변경

새로운 비밀번호

비밀번호 확인

확인

- 기능 소개
 - 회원 편집
 - 비밀번호 확인 및 변경

- 작업 내용
 - 회원 데이터를 편집하기 위해 코드 작성
 - 회원 데이터 편집을 위해선 비밀번호 확인절차를 먼저 실행하고 맞으면 비밀번호 변경 및 편집 가능

회원 편집 코드

```
<form id="user-edit-form" action="/crud/users/{{ user.id }}" method="POST" role="form" onsubmit="return user_edit_go();">
  {{ form.csrf_token }}
  <div class="card-body register-card-body">
    <div class="input-group mb-3">
      <input type="text" class="form-control" id="username" name="username" placeholder="사용자명" value="{{ user.username }}">
    </div>
    <div class="input-group mb-3">
      <input type="email" class="form-control" id="email" name="email" placeholder="메일 주소" value="{{ user.email }}">
    </div>
    <div class="input-group mb-3" onclick="OpenWindow('/crud/pwd_check?id={{ user.id }}','비밀번호변경','500','200');">
      <input type="password" class="form-control" id="password" name="password" placeholder="비밀번호" value="" readonly>
    </div>
    <div class="row">
      <div class="col-md-4">
        <button type="submit" id="submit" name="submit" class="btn btn-primary btn-block" value="갱신" onclick="return validatePassword();">갱신</button>
      </div>
    </div>
  </div>
```

비밀번호 확인 코드

```
<form method="POST" role="form" >
  {{ form.csrf_token }}
  <input type="hidden" name="user_id" value="{{id}}" />
  <label for="pwd_input" >현재 패스워드</label>
  <input type="password" name="password" />
  <input type="submit" value="전송">
</form>
```

이미지 감지 게시판



• 기능 소개

- 이미지 감지 게시판
- 이미지 등록 후 감지 버튼을 눌러서 이미지의 태그 작성

• 작업 내용

- 이미지 감지 게시판 코드 작성
- 태그를 등록하고 이미지를 감지하면 인식한 태그를 나오게 구현

이미지 등록 코드

```
<form action="{ url_for('detector.upload_image') }" method="post" enctype="multipart/form-data"
novalidate="novalidate">
    {{ form.csrf_token }}
    <div>
        <label>
            <span> {{ form.image(class="form-control-file") }} </span>
        </label>
    </div>
    {% for error in form.image.errors %}
    <span style="color: red;">{{ error }}</span>
    {% endfor %}
```

이미지 감지 코드

```
for box, label, score in zip(output["boxes"], output["labels"], output["scores"]):
    if score > 0.5 and labels[label] not in tags:
        color = make_color(labels)
        line = make_line(result_image)
        c1 = (int(box[0]), int(box[1]))
        c2 = (int(box[2]), int(box[3]))
        cv2 = draw_lines(c1, c2, result_image, line, color)
        cv2 = draw_texts(result_image, line, c1, cv2, color, labels, label)
        tags.append(labels[label])
```

공지사항 게시판

공지사항

순번	제목	작성자명	날짜
19	공지44	김호권123	2024.03.04
18	공지23	김호권123	2024.03.04
17	공지1	김호권123	2024.03.04
16	공지	김호권123	2024.03.04
15	공지	김호권11	2024.02.28
14	공지사항입니다.	김호권11	2024.02.28
6	공지사항입니다.	김호권	2024.02.26
4	공지1	김호권	2024.02.26

공지사항 목록

기능 소개

- 공지사항 게시판
- 공지사항 등록, 수정, 삭제

작업 내용

- 공지사항 게시판 구현
- 공지사항의 기능을 구현하기 위한 코드를 작성

공지사항 목록 코드

```
{% for notice in notices %}
<tr>
  <td><a href="{{ url_for('notice.notice_detail', notice_id=notice.id) }}">{{ notice.id }}</a></td>
  <td>{{ notice.title }}</td>
  <td>{{ notice.author }}</td>
  <td>{{ notice.created_at.strftime("%Y.%m.%d") }}</td>
</tr>
{% endfor %}
</tbody>
</table>
<a href="{{ url_for('notice.create_notice') }}"><input type="submit" value="작성"></a>
```

공지사항 작성 코드

```
from flask_wtf import FlaskForm
from wtforms import StringField, TextAreaField, SubmitField
from wtforms.validators import DataRequired
from wtforms import DateField

class NoticeForm(FlaskForm):
    title = StringField('제목', validators=[DataRequired()])
    author = StringField('작성자명', validators=[DataRequired()])
    content = TextAreaField('내용', validators=[DataRequired()])
    submit = SubmitField('작성')
```

Project 03.

AI기반 자율주행 솔루션 개발 플랫폼

About project

AI 기반 자율주행 솔루션을 개발하고 이를 데모 및 시연 영상을 공유하여 사용자 및 개발자들이 쉽게 접근하고, 질문하고 멘토링을 받을 수 있도록 지원하며, AI 기반 자율주행 솔루션에 관한 최신 정보와 자료를 공유하는 플랫폼을 통해 개발자들 간의 지식 공유와 기술 발전을 촉진한다.

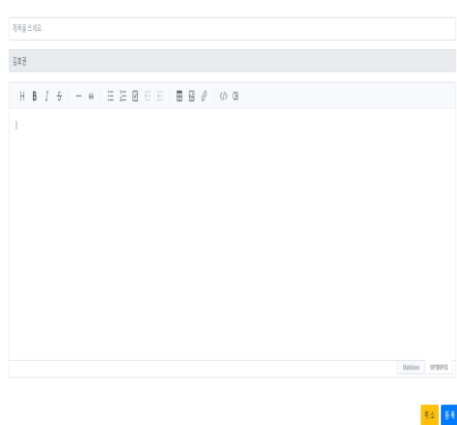
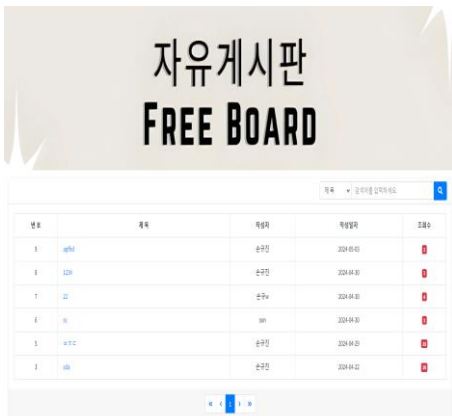
Project 03.

AI기반 자율주행 솔루션 개발 플랫폼



작업 기간	2024.03~2024.05
인력 구성	4명
프로젝트 목적	개발자과정의 최종 프로젝트로 지금까지 배운 모든 언어와 기술의 숙련도 향상을 위하여 팀으로 진행함
프로젝트 내용	AI 기반 자율주행 솔루션을 개발하고 이를 데모 및 시연 영상을 공유하여 사용자 및 개발자들이 쉽게 접근하고, 질문하고 멘토링을 받을 수 있도록 지원하며, AI 기반 자율주행 솔루션에 관한 최신 정보와 자료를 공유하는 플랫폼을 통해 개발자들 간의 지식 공유와 기술 발전을 촉진한다.
주요 업무 및 상세 역할	1) 프로젝트 총괄 관리 업무 2) DB의 자유게시판 정의 3) 자유게시판 구현 4) 체험관 페이지 제작 및 구현
사용언어 및 개발 환경	Java, Oracle, apache Tomcat, HTML, Arduino, Flask, Python, Pytorch, TensonFlow, Opencv, Kera, Yolov5, CSS, Javascript, ToastUI
참고 자료	https://github.com/snoowert/onestep.git

자유게시판 메인, 게시물 등록



- 기능 소개
 - 자유게시판 목록
 - 게시물 등록 페이지

• 작업 내용

- 자유게시판의 메인화면으로 화면 구성은 번호, 제목, 작성자, 작성일자, 조회수, 마지막으로 제목, 작성자로 검색할 수 있는 검색 바로 구성
- 자유게시판이란걸 확실히 하기위해 위에 배너를 넣음
- 자유게시판 등록 화면에는 이미지를 넣을 수 있도록 ToastUI를 사용

자유게시판 메인화면 코드

```
<tbody>
  <c:if test="${empty freeList}">
    <tr>
      <td colspan="5">해당 내용이 없습니다.</td>
    </tr>
  </c:if>
  <c:forEach items="${freeList}" var="free">
    <tr>
      <td>${free.freeid}</td>
      <td id="boardTitle" style="text-align:left;max-width: 100px; overflow: hidden; white-space: nowrap; text-overflow: ellipsis;">
        <a href="detail.do?freeid=${free.freeid}&from=List">
          <span class="col-sm-12">${free.freetitle}</span>
        </a>
      </td>
      <td>${free.writer}</td>
      <td>
        <fmt:formatDate value="${free.freeregdate}" pattern="yyyy-MM-dd"/>
      </td>
      <td><span class="badge bg-red">${free.freeviewpoint}</span></td>
    </tr>
  </c:forEach>
</tbody>
```

게시글 등록 코드

```
<div class="card-body pad">
  <form role="form" method="post" action="regist.do" name="registForm">
    <input type="hidden" id="freecontent" name="freecontent">
    <div class="form-group">
      <input type="text" id="freetitle" title="제목"
        name='freetitle' class="form-control notNull" placeholder="제목을 쓰세요">
    </div>
    <div class="form-group">
      <input type="text" id="writer" title="작성자" readonly
        name="writer" class="form-control notNull" value="${loginUser.username }">
      <input type="hidden" id="memberid" name="memberid" value="${loginUser.memberid }">
    </div>
    <div class="form-group" id="content">
      <div>
    </div>
  </form>
</div><!--end card-body -->
```


게시글 상세 페이지 코드

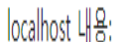
[illegible]

댓글 등록 코드

```
<sec:authorize access="isAuthenticated()">
<form class="comment-form insert-form" action="commentregist" method="post">
  <input type="hidden" name="freeid" value="{free.freeid}" />
  <input type="hidden" name="memberid" value="{loginUser.memberid}" />
  <div class="row border py-2">
    <input class="font-weight-bold form-control col-sm-12 mb-2" readonly name="username" value="{loginUser.username}" />
    <textarea class="reply-form-textarea form-control" name="commentcontent" required="required" rows="3">내용을 입력하세요.</textarea>
  </div>
  <div class="row">
    <div class="col-sm-6">
      <button type="button" class="btn btn-outline-secondary" onclick="goToMainPage()">메인으로</button>
    </div>
    <div class="col-sm-6 text-right">
      <button type="submit" class="btn btn-primary">등록</button>
    </div>
  </div>
</form>
</sec:authorize>
```

- **기능 소개**
 - 자유게시판 상세페이지
 - 댓글 등록, 수정, 삭제
- **작업 내용**
 - 자유게시판의 상세 페이지로 수정, 삭제 버튼과 제목, 작성자, 작성일, 조회수, 내용으로 구성되어 있으며 밑에는 댓글과 댓글을 입력할 수 있는 창이 있다.
 - 게시글과 댓글은 쓴 사람과 로그인 한 사람이 같아야만 수정, 삭제버튼이 나온다.

게시글 수정 코드



게시글이 삭제되었습니다.



```
<div class="card-body">
    <form role="form" method="post" action="modify.do" name="modifyForm" onsubmit="return false;">
        <input type="hidden" name="freeid" value="{free.freeid}" />
        <div class="form-group">
            <input type="text" id="freetitle" name='freetitle' class="form-control" value="{free.freetitle }">
        </div>
        <div class="form-group">
            <input type="text" id="writer" readonly name="writer" class="form-control" value="{free.writer }">
            <input type="hidden" id="memberid" readonly name="memberid" class="form-control" value="{free.memberid }">
        </div>
        <div class="form-group">
            <div id="content"></div>
            <input type="hidden" name="freecontent" id="freecontent"/>
        </div>
    </form>
</div><!--end card-body -->
<div class="card-footer" style="display:none"></div>
<div class="card-footer text-right"> <!-- 수정된 부분 -->
    <button type="button" class="btn btn-warning" id="modifyBtn" onclick="modify_submit();">수 정</button>
    &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~
    <button type="button" class="btn btn-default" id="cancelBtn" onclick="history.go(-1);">취 소</button>
</div>
```

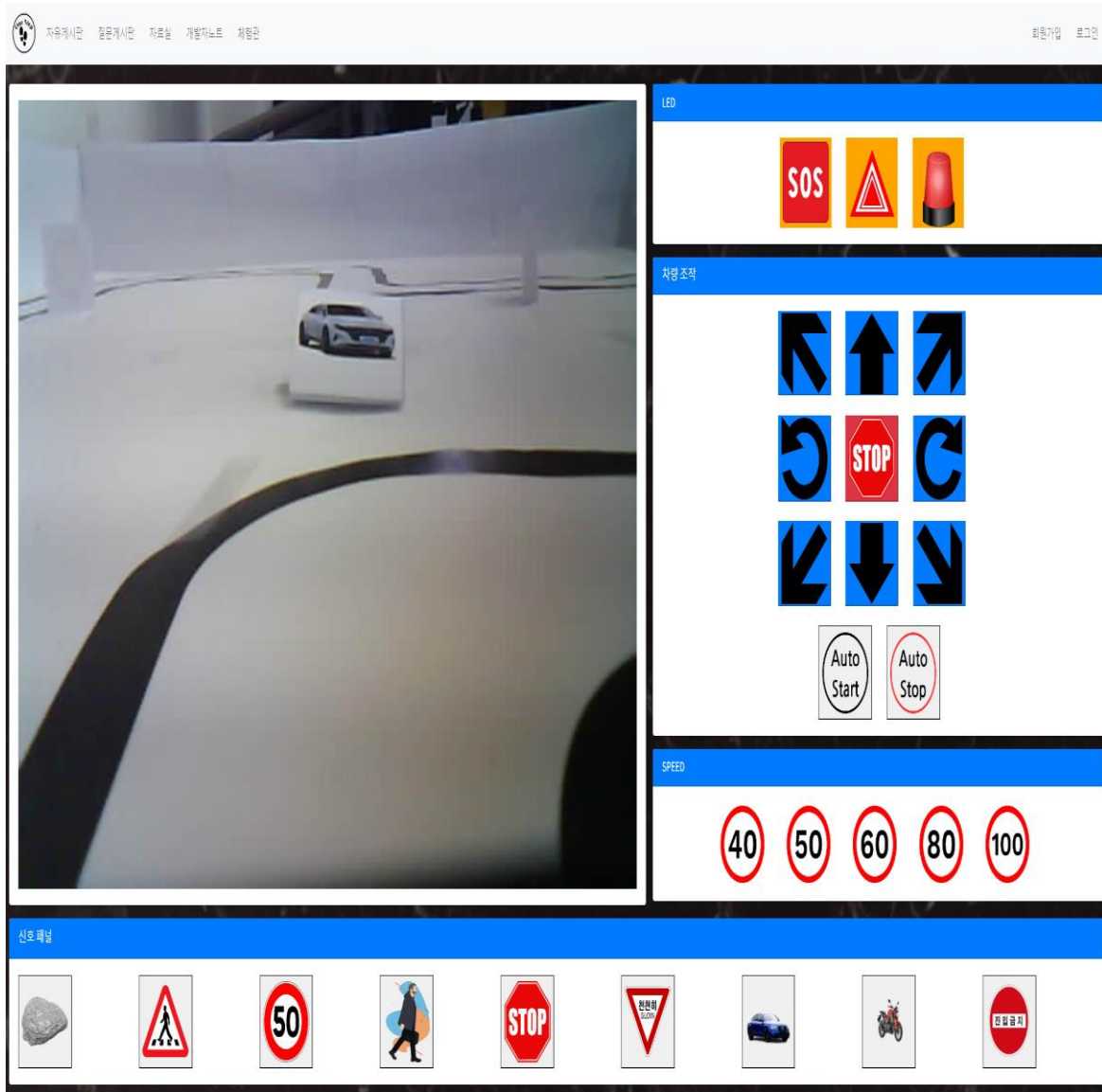
게시글 삭제 코드

```
function submit_go(url) {
    if(url=="remove.do" && !confirm("정말로 삭제하시겠습니까?")){
        return;
    }
    formObj.action=url;
    formObj.submit();
}
```

- 기능 소개
 - 게시물 수정
 - 게시물 삭제

- **작업 내용**
 - 게시글 상세페이지에서 수정 버튼을 눌러서 들어오는 화면을 구현
 - ToastUI를 사용하여 이미지도 수정 가능
 - 삭제버튼을 누르면 삭제 확인 메시지 출력 후 삭제

체험관 페이지



실시간 영상 연결 코드

```
<div class="col-md-7">
  <div class="card card-primary">

    <div class="card-body">
      <img id="videoElement" style="width:100%; height:973px;"></img>
    </div>

  </div>
</div>

window.onload = function() {
  var videoElement = document.getElementById('videoElement');
  var sourceUrl = 'http://127.0.0.1:5000/onestep/video';
  videoElement.src = sourceUrl;
};
```

```
<div>
  <iframe src="http://127.0.0.1:5000/onestep/video"
    style="width: 470px; height: 270px; border-radius: 10px; border:1px solid black;"></iframe>
</div>
```

```
@onestep.route('/')
def car():
    return render_template('car.html')

def send_video():
    global thread_frame

    while True:
        ret, buffer11 = cv2.imencode('.jpg', thread_frame)
        frame11 = buffer11.tobytes()
        yield (b'--frame\r\n' b'Content-Type: image/jpeg\r\n\r\n' + frame11 + b'\r\n\r\n')

@onestep.route('/video')
def video():
    return Response(send_video(), mimetype='multipart/x-mixed-replace; boundary=frame')
```

[illegible]

```
@onestep.route('/light', methods=['POST'])
def light():
    data = request.get_json()
    direction = data.get('direction')

    if direction == 'sos':
        urlopen('http://' + ip + "/action?go=sos")
    elif direction == 'light':
        urlopen('http://' + ip + "/action?go=light")
    elif direction == 'danger':
        urlopen('http://' + ip + "/action?go=danger")
```

LED



SOS



비상등

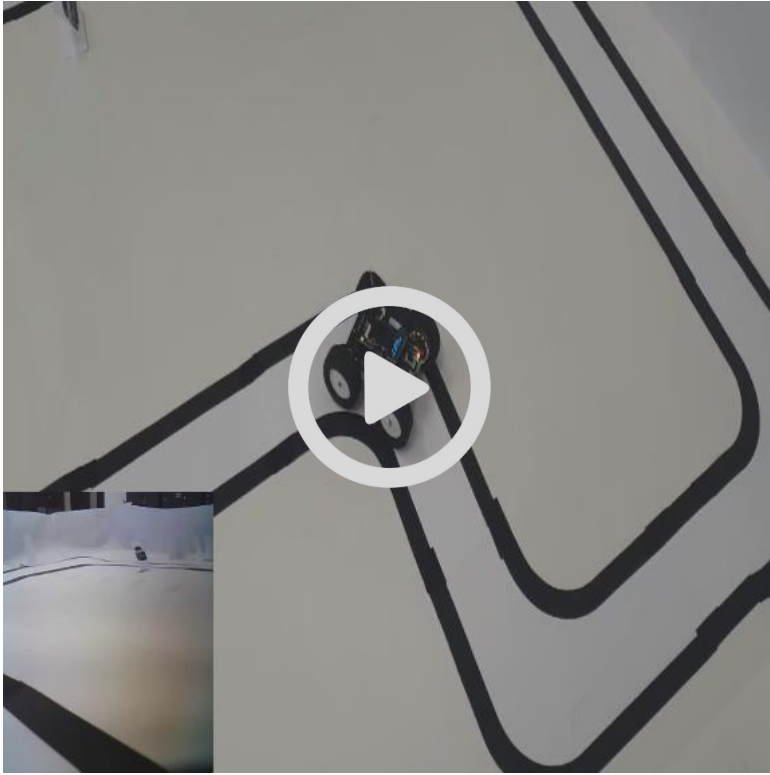


페이드

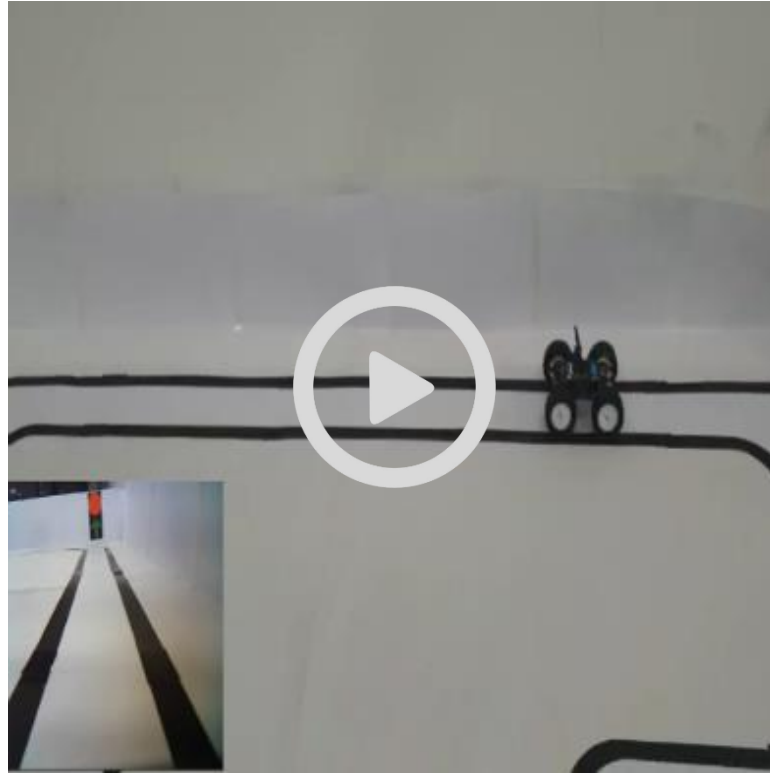
```
<div class="card-body">  
    <div class="row justify-content-center">  
        <button class="image-button btn-primary" onclick="manualControl('left');">  
              
        </button>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <button class="image-button btn-primary" onclick="manualControl('forward')">  
              
        </button>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
        <button class="image-button btn-primary" onclick="manualControl('right');">  
              
        </button>&nbsp;&nbsp;&~  
    </div>  
</div>
```

```
@onestep.route('/manual-control', methods=['POST'])
def manual_control():
    global auto
    data = request.get_json()
    direction = data.get('direction')
    if auto==False:
        if direction == 'forward':
            urlopen('http://' + ip + "/action?go=forward")
        elif direction == 'left':
            urlopen('http://' + ip + "/action?go=left")
        elif direction == 'right':
            urlopen('http://' + ip + "/action?go=right")
        elif direction == 'stop':
            urlopen('http://' + ip + "/action?go=stop")
        elif direction == 'turn_left':
            urlopen('http://' + ip + "/action?go=turn_left")
        elif direction == 'turn_right':
            urlopen('http://' + ip + "/action?go=turn_right")
        elif direction == 'bleft':
            urlopen('http://' + ip + "/action?go=bleft")
        elif direction == 'backward':
            urlopen('http://' + ip + "/action?go=backward")
        elif direction == 'bright':
            urlopen('http://' + ip + "/action?go=bright")
    return jsonify(status="success", message=f"수동 주행 - {direction}")
```


차량 조작(수동)



좌회전



전진



우회전

차량 조작(수동)



후진
좌회전



후진



후진
우회전

차량 조작(수동)



좌측 턴



우측 턴

```
<div class="row justify-content-center">  
    <button class="image-button" onclick="startDriving()" style="width:116px;height:116px">  
          
    </button>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    <button class="image-button" onclick="stopDriving()" style="width:116px;height:116px">  
          
    </button>  
</div>
```

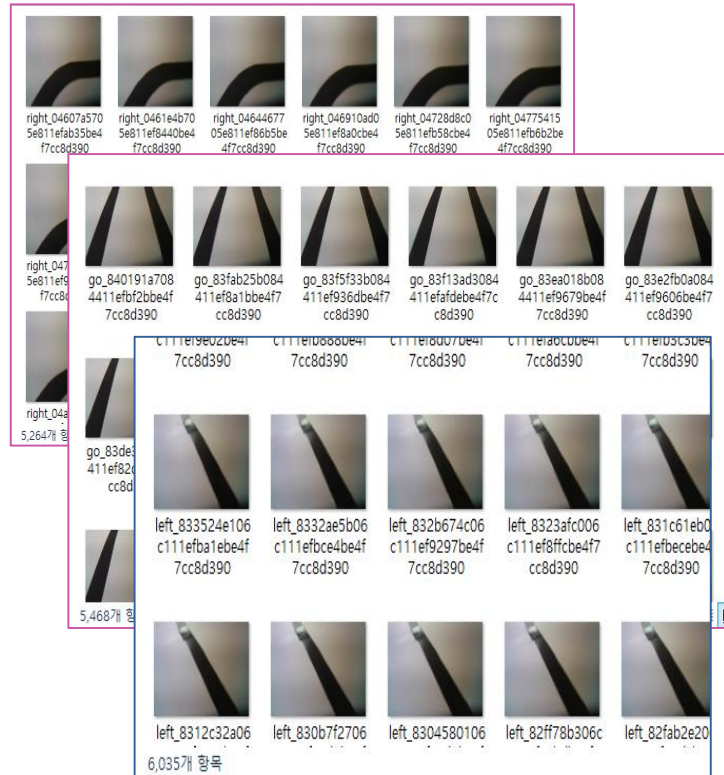
```
function startDriving() {
  fetch('http://127.0.0.1:5000/onestep/start-driving', { method: 'POST' })
    .then(response => response.json())
    .then(data => console.log('주행 시작:', data))
    .catch(error => console.error('주행 시작 오류:', error));
}
```

```
function stopDriving() {
  fetch('http://127.0.0.1:5000/onestep/stop-driving', { method: 'POST' })
    .then(response => response.json())
    .then(data => console.log('긴급 정지:', data))
    .catch(error => console.error('긴급 정지 오류:', error));
}
```

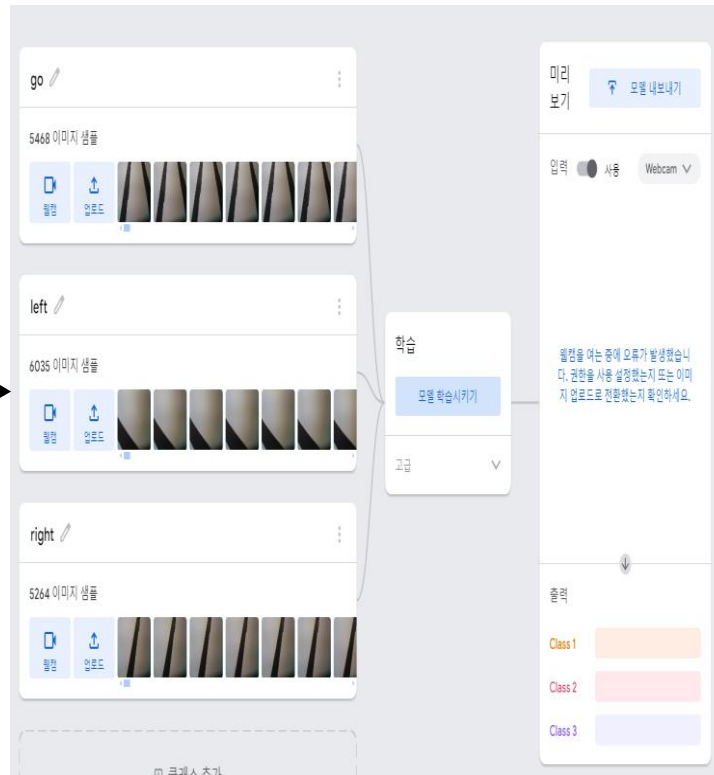
```
# 자율주행 시작 제어
@onestep.route('/start-driving', methods=['POST'])
def start_driving():
    global auto
    try:
        if auto==False:
            auto=True
            urlopen('http://' + ip + '/action?go=speed50')
            return jsonify(status="success", message="주행 시작됨")
    except Exception as e:
        return jsonify(status="error", message=str(e))

# 자율주행 정지 제어
@onestep.route('/stop-driving', methods=['POST'])
def stop_driving():
    global auto
    try:
        if auto==True:
            auto=False
            urlopen('http://' + ip + '/action?go=stop')
            return jsonify(status="success", message="차량 정지됨")
    except Exception as e:
        return jsonify(status="error", message=str(e))
```

자율 주행을 위한 학습 과정



트랙 촬영



티처블 머신 학습

```
def image_process_thread():
    global img, ip, image_flag, car_state2, motor_model, class_names, auto

    while True:
        if image_flag == 1:
            img = np.asarray(img, dtype=np.float32).reshape(1, 224, 224, 3)
            img = (img / 127.5) - 1

            # Predict the img_model
            prediction = motor_model.predict(img)
            index = np.argmax(prediction)
            class_name = class_names[index]
            confidence_score = prediction[0][index]
            percent = int(str(np.round(confidence_score * 100))[:-2])

            if auto==True:
                if "go" in class_name[2:] and car_state2 == "go" and percent >= 95:
                    print("직진:", str(np.round(confidence_score * 100))[:-2], "%")
                    urlopen('http://' + ip + "/action?go=forward")
                elif "left" in class_name[2:] and car_state2 == "go" and percent >= 90:
                    print("좌회전:", str(np.round(confidence_score * 100))[:-2], "%")
                    urlopen('http://' + ip + "/action?go=left")
                elif "right" in class_name[2:] and car_state2 == "go" and percent >= 100:
                    print("우회전:", str(np.round(confidence_score * 100))[:-2], "%")
                    urlopen('http://' + ip + "/action?go=right")

            image_flag = 0
```

자율 주행 코드 입력

자율 주행



```
<div class="row justify-content-center">
    <button class="image-button btn-primary" style="background:transparent;border:none;color:black" onclick="speedcontrol('speed40');">
        
    </button>
    <button class="image-button btn-primary" style="background:transparent;border:none;color:black" onclick="speedcontrol('speed50');">
        
    </button>
    <button class="image-button btn-primary" style="background:transparent;border:none;color:black" onclick="speedcontrol('speed60');">
        
    </button>
    <button class="image-button btn-primary" style="background:transparent;border:none;color:black" onclick="speedcontrol('speed80');">
        
    </button>
    <button class="image-button btn-primary" style="background:transparent;border:none;color:black" onclick="speedcontrol('speed100');">
        
    </button>
</div>
```

```
@onestep.route('/speed', methods=['POST'])
def speed():
    data = request.get_json()
    direction = data.get('direction')

    if direction == 'speed40':
        urlopen('http://' + ip + "/action?go=speed40")
    elif direction == 'speed50':
        urlopen('http://' + ip + "/action?go=speed50")
    elif direction == 'speed60':
        urlopen('http://' + ip + "/action?go=speed60")
    elif direction == 'speed80':
        urlopen('http://' + ip + "/action?go=speed80")
    elif direction == 'speed100':
        urlopen('http://' + ip + "/action?go=speed100")
```

신호 패널 코드

```
<div class="col my-auto">
    <button class="image-button" style="width:116px;height:116px">
        
    </button>
</div>
<div class="col my-auto">
    <button class="image-button" style="width:116px;height:116px">
        
    </button>
</div>
<div class="col my-auto">
    <button class="image-button" style="width:116px;height:116px">
        
    </button>
</div>
```

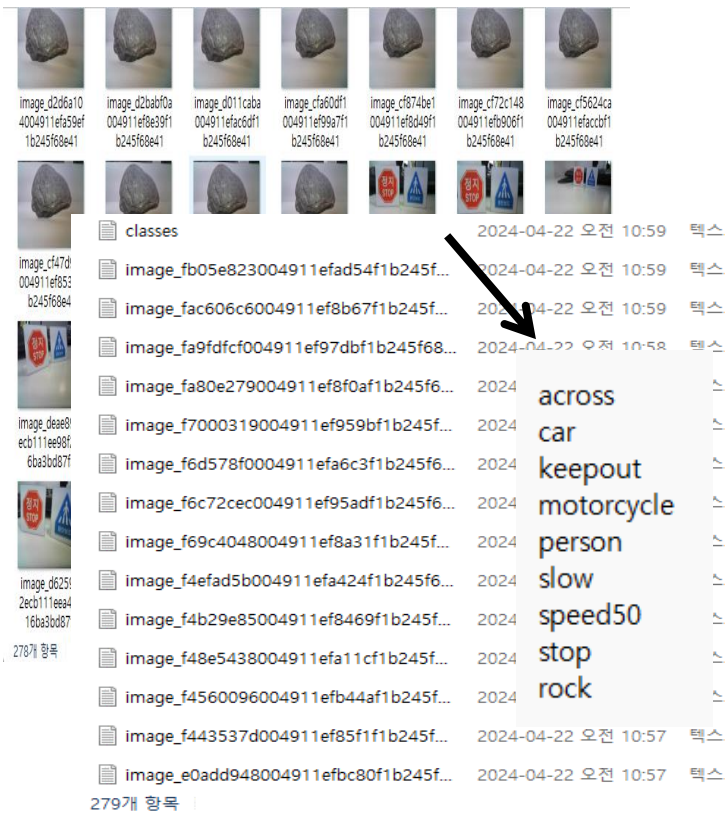
```
if not detections.empty:
    # 결과를 반복하며 객체 표시
    for _, detection in detections.iterrows():
        x1, y1, x2, y2 = detection[['xmin', 'ymin', 'xmax', 'ymax']].astype(int).values
        label = detection['name']
        conf = detection['confidence']

        if "stop" in label and conf > 0.8:
            print("stop")
            car_state2 = "stop"
            urlopen('http://' + ip + '/action?go=stop')
        elif "speed50" in label and conf > 0.6:
            print("go")
            car_state2 = "go"
            urlopen('http://' + ip + '/action?go=speed50')
        elif "rock" in label and conf > 0.85:
            print("rock")
            car_state2 = "stop"
            urlopen('http://' + ip + '/action?go=stop')

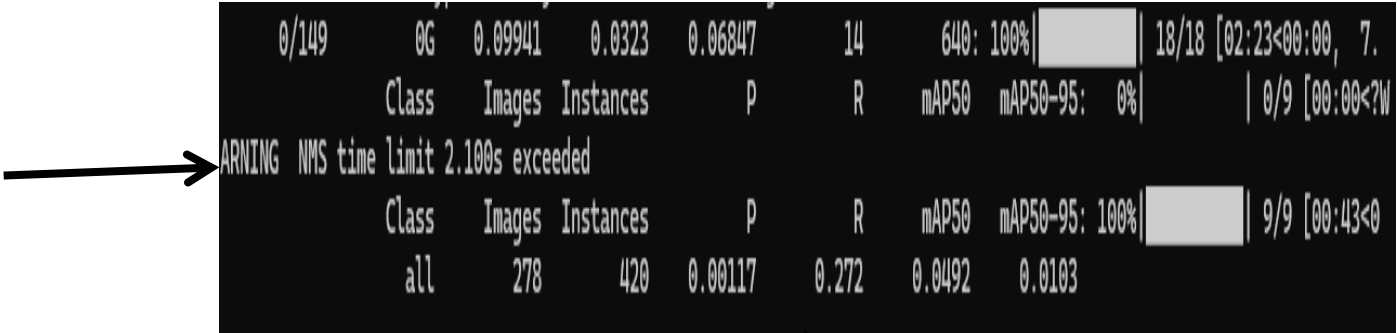
    # 박스와 라벨 표시
    color = [int(c) for c in random.choice(range(256), size=3)]
    cv2.rectangle(thread_frame, (x1, y1), (x2, y2), color, 2)
    cv2.putText(thread_frame, f'{label} {conf:.2f}', (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)

thread_image_flag = 1
image_flag = 0
```

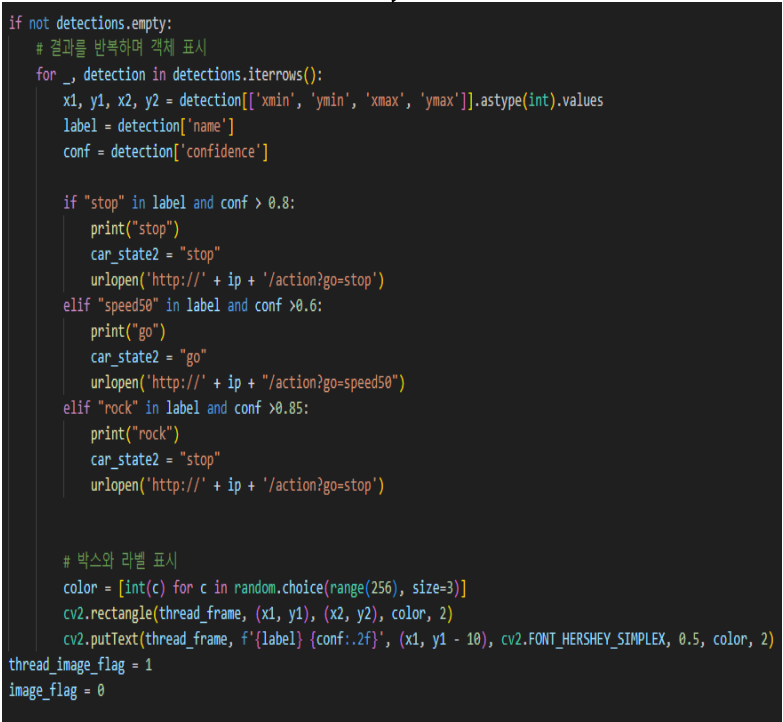

이미지 감지를 위한 학습 과정



이미지 촬영



PowerShell
학습

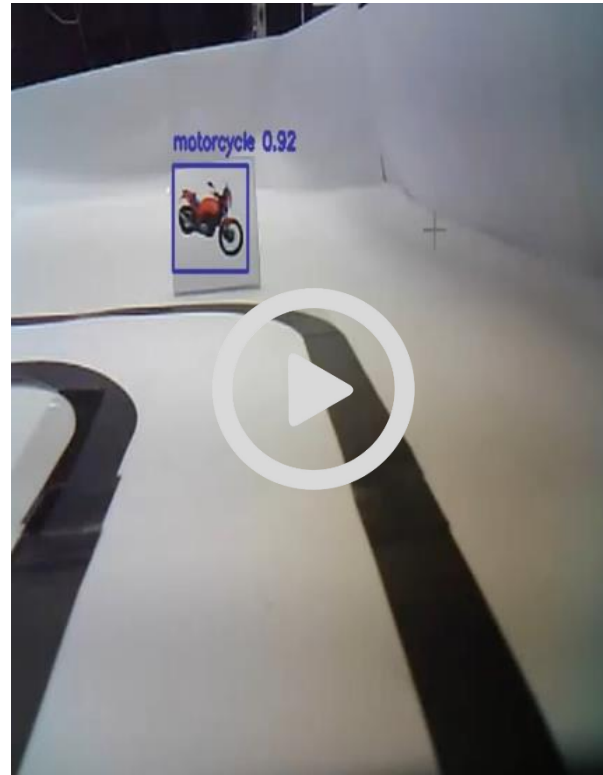


학습된 이미지
감지 코드 입력

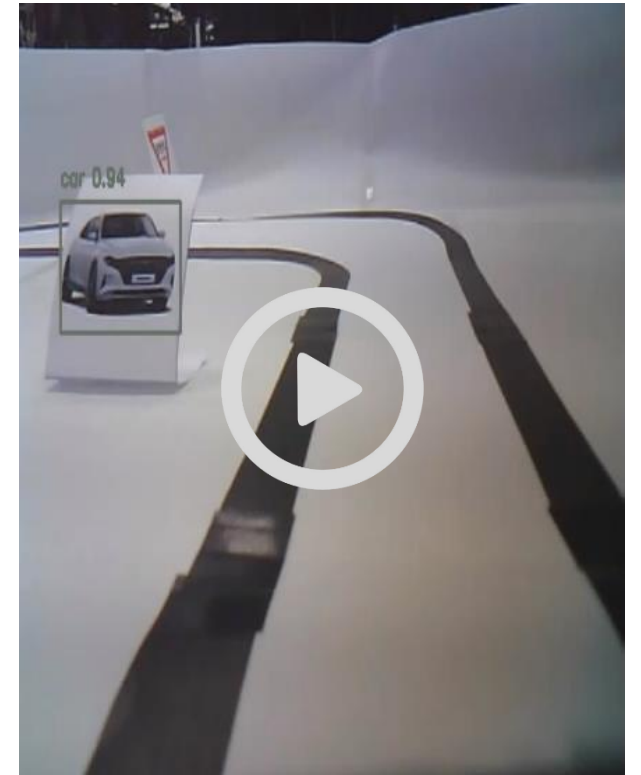
이미지 감지(영상)



across



motorcycle

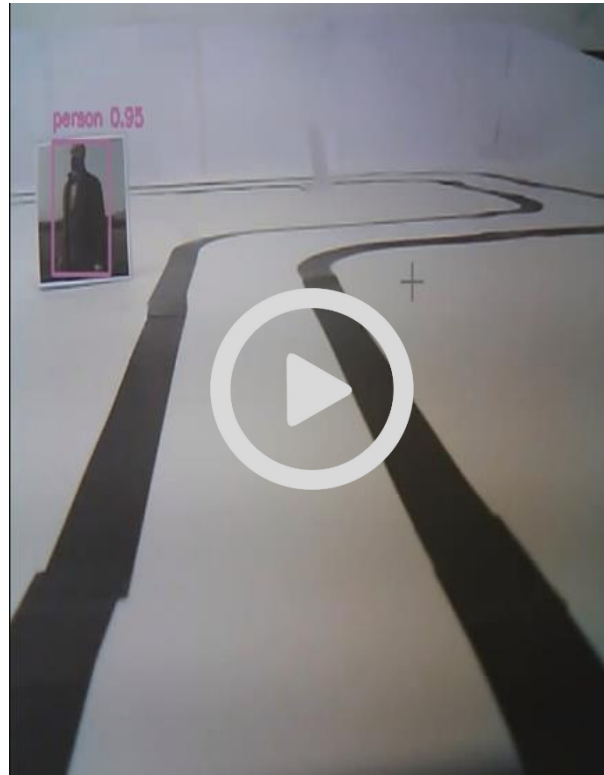


car

이미지 감지(영상)



keepout



person



rock

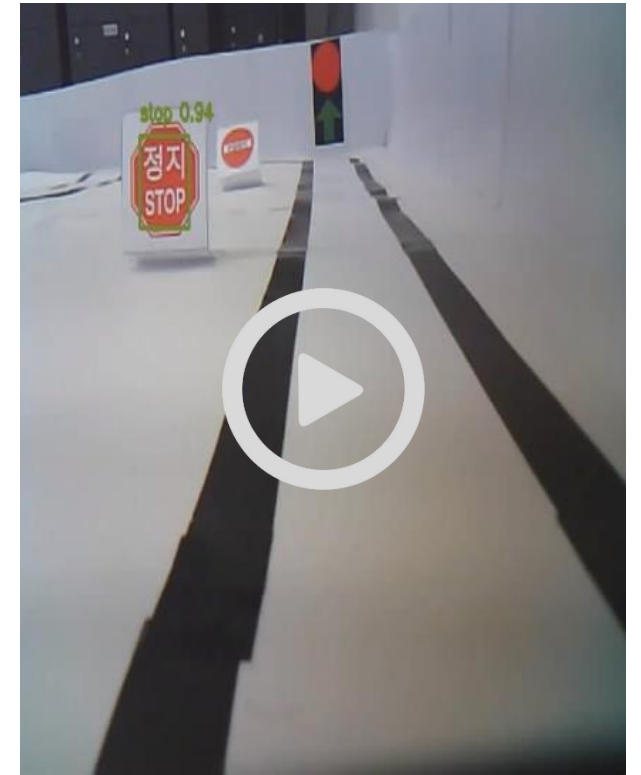
이미지 감지(영상)



slow



speed50



stop

Thank you

김호권

Tel: 010-3935-8229

Email: gisf456@gmail.com

Github: <https://github.com/KimHoGwon>

kakaoID: gisf456