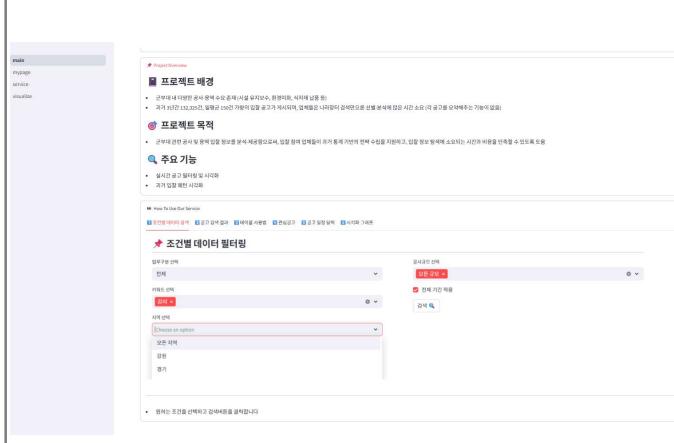


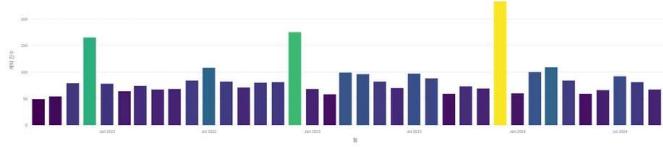
프로젝트명	국방 분야 공사·용역 공고 입찰 전략을 위한 통계 분석 및 정보 제공 서비스	수행기간	4/30 ~ 5/15
팀원	주재현, 김형래, 안보경, 김나영	GitHub url	x
역할 및 참여범위	-역할 : 웹 크롤링 및 대시보드 구현, 분석, DB설계		
프로젝트 개요	<p>배경 : 군부대 내의 다양한 공사 용역에 대한 입찰의 어려움      타겟 : 군부대 시설 공사 및 용역 입찰을 원하는 업체      목적 : 군부대 관련 공사 및 용역 입찰 정보를 분석·제공      입찰 참여 업체들이 과거 통계 기반의 전략 수립 지원      입찰 정보 탐색에 소요되는 시간과 비용을 단축   </p>		
개발 PLATFORM	VScode / Streamlit / MySQL		
각별히 애썼던 부분	<p>웹 크롤링에서 크롤링이 불가능 한 페이지가 우리 조가 완벽하게 필요한 부분이어서 다른 페이지에서 크롤링(데이터 수집)을 하느라 많은 노력을 기울임. 또한 크롤링 데이터들을 전처리 하는거에 있어서 여러 가지 문제들을 해결하느라 노력을 많이 함      대시보드에서 알바몬 / 알바천국 같이 캘린더 및 관심공고등을 표현하여 사용자 친화적인 UI를 제공함</p>		
 <p>Main 페이지로 우리가 제공하는 서비스를 사용하는 방법을 알려주는 페이지입니다.</p>			

사용자가 관심있는 공고를 지역 / 키워드 별로 검색할 수 있습니다.

또한 공고 검색 결과를 요약해서 표로 보여주고  
지도를 표시하여 시각적으로 표현해줍니다.

사용자가 선택한 키워드 / 지역 별 공고 데이터들의 분석  
그래프를 보여주는 페이지입니다.

키워드별 / 예산 추이 분석 / 규모별 분석 그래프를  
제공합니다.



공사 규모별 분석 그래프 예시

마이페이지로 내가 관심가는 목록을 추가하여 필요없는 정보를 제외하고 온전히 필요한 공고들만을 볼 수 있는 기능입니다.

# 계약 데이터 분석 대시보드

※ 내부 관리고  
※ 내부 관리자  
※ 내부 관리자

## 내 관심 공고 목록

관심사를 등록한 공고 목록입니다. 가기마다 관심 공고 목록을 자주 고 싶으면 선택지를 혹은 전체목록 누른채로

번호	제목	작성일	상태
10:74	‘25년 우안면 행정복지센터 기관 소속도면 연구 품목’ ※ 사진   ☎ 02-510-1111   읽어보기	2024-01-10	미답변
10:14	송장동 도도 행정복지센터 기관 품목 ※ 사진   ☎ 02-510-0000   읽어보기	2024-01-10	미답변
00:132	00부문 면허 및 저작권료 및 협회회비(비상금) ※ 사진   ☎ 02-510-0000   읽어보기	2024-01-09	미답변

## 공고 일정 달력

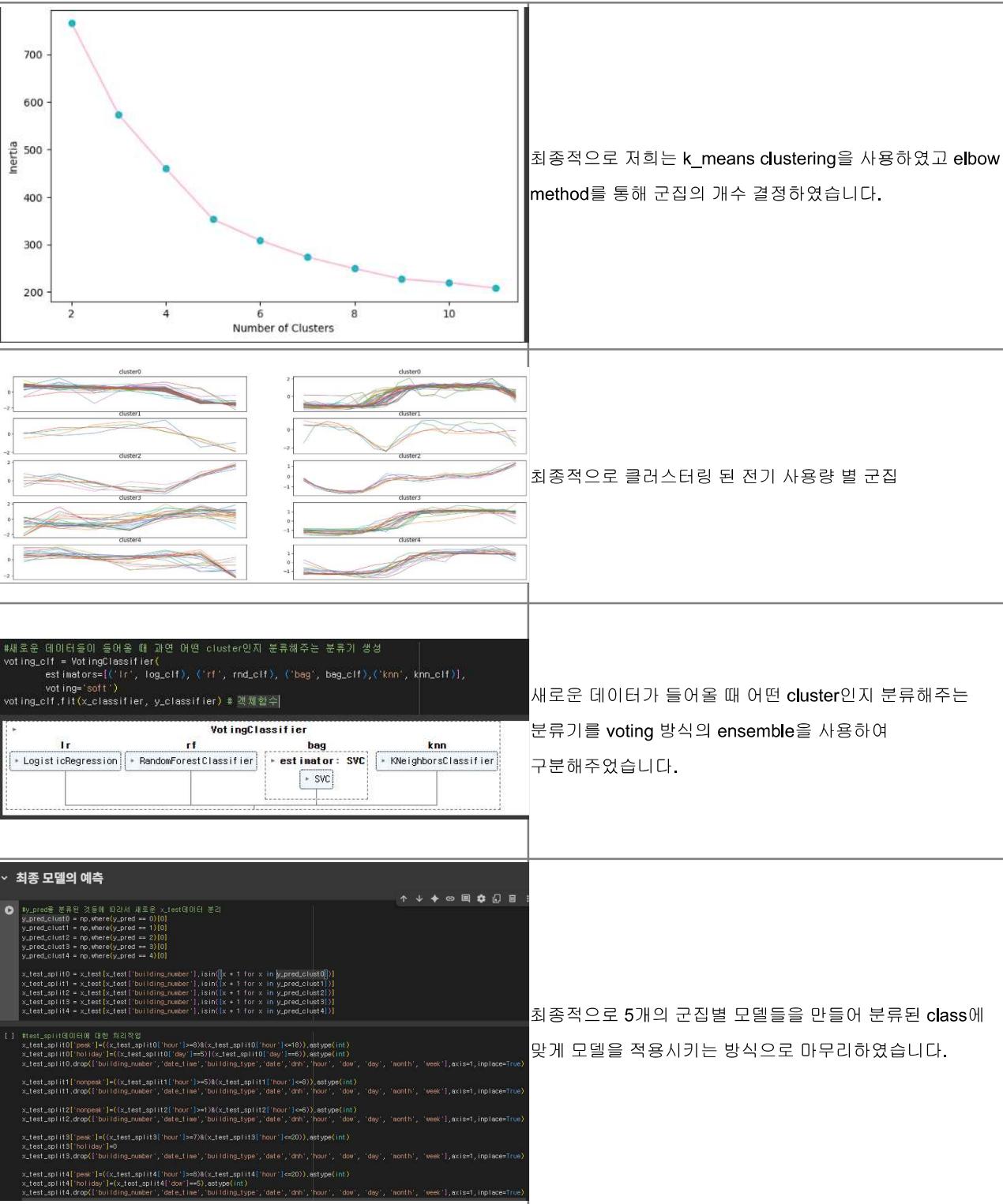
※ 공고 일정 달력

관심 공고의 시작 마감 일자를 달력에 추가하여  
가시성을 높여주고 사용자가 일정을 잊지 않도록 도와주는  
기능을 합니다.

프로젝트명	전력 사용량 예측 AI 모델 개발	수행기간	2023/12/18 ~ 2023/12/24
팀원	주재현 안영록 한성모 장기윤	GitHub url	x
역할 및 참여범위	<p>-역할 : 데이터 전처리, 머신러닝 모델링, EDA, feature engineering</p>		
프로젝트 개요	<p>정적이고 효율적인 에너지 공급을 위해서는 전력 사용량에 대한 정확한 예측이 필요합니다. 따라서 한국에너지공단에서는 전력 사용량 예측 시뮬레이션을 통한 효율적인 인공지능 알고리즘 개발을 목표로 하는 프로젝트입니다.</p>		
개발 PLATFORM	Google Colab		

	<p>1. 전체적인 방향성</p> <ul style="list-style-type: none"> <li>* 건물별 군집화를 통해 클러스터를 나눈 후 나눠진 클러스터별로 다른 피쳐를 추가해 모델을 생성하는 방식 최종적으로 우리는 <b>k_means clustering</b>을 사용하였고 <b>elbow method</b>를 통해 군집의 개수 결정하였습니다. 결정된 군집 수 만큼 모델 생성(각 클러스터마다 건물별로 데이터들이 들어있을것인데 각 건물별로 모델링 한 후 양상을 통해 클러스터별 최종 한개의 모델을 생성하였습니다) 또한 요일별 전력 중앙값, 시간대별 전력 중앙값으로 분류기를 만들어 건물별로 어느 군집에 속하는지 분류를 실시하였습니다. 이후 각 분류에따라 테스트 데이터에 피쳐를 군집에 특성에 맞는 피쳐를 각공해주고 군집에 맞는 최종 모델로 예측을 실시하였습니다.</li> </ul>
각별히 애썼던 부분	<p>2. 이상치 및 결측치 처리</p> <ul style="list-style-type: none"> <li>* 결측치는 아래의 코드에서 설명되어있듯이 풍속과 습도는 선형보간을 통해 처리해주었습니다. 또한 풍속과 습도에 관련된 불쾌지수, 체감온도 피쳐를 공식에따라 작성하여 추가하였고, 냉방면적에 대한 전력사용량 관계성을 위해 냉방도시피쳐를 추가하였습니다.</li> <li>* 이상치는 <b>eda</b>를 통해 각 건물마다 모든 일자의 전력사용량 그래프를 그려 분석하며 어느날 이상치가 있거나 임시휴무등을 했는지 판단했습니다. 원래 주말이 아닌 공휴일(6월1일, 6월6일, 8월15일)은 모델이 공휴일을 예측해야하는 부담이 존재하여 정확성이 떨어진다고 생각하여 그냥 아무것도 없는 날로 대체하였습니다. 즉 공휴일에 대해 영향을 받아 쉬는날이 되어버린 건물에 대해서 다른 모든 월요일, 수요일의 평균을 넣어줌으로써 공휴일이 없던것처럼 만들었습니다.</li> <li>* 또한 그래프 보면서 판단한 공휴일이 아닌 날들에 대해 주기성이 없는 데이터들 혹은 튀는 값들은 비슷한 스케일을 가진 날짜들의 평균으로 치환해줬습니다. 또한, 매우 큰 이상치를 가지는 데이터에 대해서 다른 날의 전력 사용량으로 치환하여 적용하였습니다. 그 다른날의 기준은 전력사용량이 완전 비슷한 정도의 날로 대체한 것입니다.</li> <li>* 주말에 임시적으로 쉬는 건물들에 대해서 주기성을 만들어 주는 작업을 진행하였습니다. 예를들어 7월10일에 쉬고 7월 24일에 쉬는 격주마다 쉬는 건물이였는데 7월31일도 쉬었다고 한다면 이 건물에 대해서 7월 31일 데이터는 안 쉰 데이터로 치환하는 작업을 하였습니다. 이로서 모델은 격주로 쉬는 날임을 예측할 것입니다.</li> </ul>
	<p>3. 후처리</p> <ul style="list-style-type: none"> <li>* 클러스터를 나눈 후 모델을 학습하기위해서 클러스터별 리스트형태로 데이터프레임들을 넣고, 각 데이터프레임에 클러스터에 특징에 맞게 피쳐를 추가하는 후처리를 진행하였습니다.</li> <li>* 또한 테스트 데이터들도 후처리 작업이 필요하므로 분류된 클러스터에 맞게 데이터 프레임을 후처리하는 작업을 진행하였습니다.</li> </ul>

<p>▼ 1. 시간에 따른 전력 사용량 피쳐</p> <pre>[1]: #to_datetime을 통해 일반적인 숫자데이터들을 시간에 따른 데이터로 변환해줌 date = pd.to_datetime(total_df['date'])  #날짜의 날짜와 일의 티스푼 숟가락처럼 갖춰지지 않은 날짜경로 date.not_holiday total_df['hour'] = date.dt.hour total_df['day'] = date.dt.weekday #day of week total_df['day'] = date.dt.day total_df['month'] = date.dt.month total_df['week'] = date.dt.weekofyear</pre> <p>시간의 푸리에 변환</p> <p><a href="https://dacon.io/competitions/official/235757/codeshare/3590">https://dacon.io/competitions/official/235757/codeshare/3590</a></p> <p><a href="https://www.analyticsvidhya.com/blog/2019/12/6-powerful-feature-engineering-techniques-time-series/">https://www.analyticsvidhya.com/blog/2019/12/6-powerful-feature-engineering-techniques-time-series/</a></p> <p>아래는 일반적인 하루 시간에 따른 그래프이다. 아래와 같은 방식으로 모델학습에 시간데이터를 입력하게 된다면 지난날 23시55분과 오늘의 00시 05분은 10분 차이가 아닌 23시간 50분의 차이로 인식하게된다.</p> <p>▶ 이미지</p> <p>따라서 우리는 주기성을 보다 확실하게 모델에 전달하기 위해 시간의 푸리를 변환을 사용해야한다. sin 파형으로 바꾸어 지난날 23시55분과 오늘의 00시 05분은 10분 차이로 인식하게 만들었지만 정오가 저녁이랑 같아지는 오류를 범하게 된다.</p> <p>▶ 이미지</p> <p>그래서 우리는 cos으로의 변환을 추가하여 두 시간상의 데이터를 모두 반영하면 모델은 시간의 주기성을 확실하게 이해할 수 있을것이다.</p> <p>▶ 이미지</p> <p>최종적으로 모델은 이러한 형태로 시간의 주기성을 인식하게 된다.</p> <p>▶ 이미지</p> <pre>[1]: #날짜의 지름을 봄고서 푸리의 변환코드를 작성해주세요 total_df['sin_1sec'] = np.sin(2*np.pi*total_df['hour']/24) total_df['cos_1sec'] = np.cos(2*np.pi*total_df['hour']/24)</pre>	<p>전력 소비량을 결정하는 가장 큰 요인이 될 수 있는 점을 생각해보면 사람이라고 생각했습니다. 사람이 사용하지 않으면 전력은 소비량은 현저히 줄 것입니다.</p> <p>그렇다면 사람들의 전력 소모의 양상을 결정하는 큰 요인에 대해 고민해본 결과 크게 두가지로 나뉘었습니다.</p> <ol style="list-style-type: none"> <li>1. 시간에 따른 전력 사용량</li> <li>2. 기후정보에 따른 전력 사용량</li> </ol> <p>세부적으로 다른 전력 소비량의 양상이 나타난 이유는 그 건물의 쓰임새가 다르기 때문이라 생각했습니다.</p> <p>이에따라 건물별 vs 건물유형별로 모델을 만들 수 있었고, 기본적으로 나뉘어져 있는 12개의 건물유형은 제대로된 분류라고 생각되지 않았으므로 clustering을 통해 추가적으로 건물 유형을 나눌 수 있다고 생각했습니다.</p> <p>군집화된 건물마다 다른 feature engineering을 통해 군집화된 수 만큼 예측모델을 만들어 새로운 데이터들이 들어온다면 classifier를 통해 어느 군집에 속하는지 판단한 뒤 건물 목적에 맞게 모델을 사용해서 예측 할 예정입니다. 따라서 기본적인 시간에 따른 피쳐와 기후정보에 따른피쳐, 전력사용량 통계는 공통적으로 들어가야 모델의 성능이 향상 되므로 공통적으로 수행해 주었습니다.</p>
<p>▼ 2. 기후에 따른 전력 사용량 피쳐</p> <p>기후에 따른 전력사용량 및 링크를 통해 기후에 따른 전력사용량의 변수들을 더 생각해본 결과 냉방도시에 관한 아이디어를 생각했다.</p> <p>냉방도시 코드 차용 및 링크를 통해 냉방도시 변수를 추가하기 위한 함수를 사용하였다.</p> <pre>[1]: #불편지수 및 확장온도 total_df['Th'] = pd.Series(0.01*total_df['temperature']+0.01+total_df['humidity']*(0.99+total_df['temperature']-14.3)+46.3) total_df['Tct'] = 19.2408215*total_df['temperature']-11.37*total_df['windspeed']+0.16403965*total_df['temperature']+total_df['windspeed']+0.0001*total_df['windspeed']**2  #냉방도시 코드 및 링크에서 차용 def CDR(temp):     cuseus = np.cuseus((x - 26))     return np.concatenate((cuseus[11], cuseus[11:-11]))</pre> <p>def calculate_and_add_cdr(dataframe):     cdrs = []     for i in range(0, 10):         temp = datafram[building_number] == i][['temperature']].values         cdr = CDR(temp)         cdrs.append(cdr)     return np.concatenate(cdrs)  total_df['CDH'] = calculate_and_add_cdr(total_df) </p>	<p>기후에 따른 전력사용량 및 링크를 통해 기후에 따른 전력사용량의 변수들을 더 생각해본 결과 냉방도시에 관한 아이디어를 생각했습니다.</p> <p><a href="https://dacon.io/competitions/official/235736/codeshare/2743?page=1&amp;dtype=recent">https://dacon.io/competitions/official/235736/codeshare/2743?page=1&amp;dtype=recent</a></p> <p>냉방도시 코드 차용 및 링크를 통해 냉방도시 변수를 추가하기 위한 함수를 사용하였습니다.</p> <p><a href="https://dacon.io/competitions/official/236125/codeshare/8782?page=1&amp;dtype=recent">https://dacon.io/competitions/official/236125/codeshare/8782?page=1&amp;dtype=recent</a></p>
<pre>#전력사용량이 아닌 '요일과 시간대에 따른 전력 사용량의 경향성'에 따라서만 궁금한 것이라면, 평균화 scaling이 필요함 for i in range(len(df)):     #요일 평 전력 증양값에 대해 scaling     df[i].loc[1, 18] = (df[i].loc[1, 18] - df[i].loc[1, 18].mean()) / df[i].loc[1, 18].std()     #시간대 평 평균값에 대해 scaling     df[i].loc[1, 8] = (df[i].loc[1, 8] - df[i].loc[1, 8].mean()) / df[i].loc[1, 8].std()  #그려보는 일자별 평균 즉 일자별로 전력 사용량의 경향성 아래그래프는 하루중 시간에 따른 전력 사용량의 경향성 그래프임 fig = plt.figure(figsize=(10,3)) for i in range(len(df)):     plt.plot(df[i].loc[1:, 18], alpha = 0.5, linewidth = 0.5) fig = plt.figure(figsize=(20,3)) for i in range(len(df)):     plt.plot(df[i].loc[1:, 8], alpha = 0.5, linewidth = 0.5)</pre>	<p>건물별군집화를 통해 클러스터를 나눈 후 나눠진 클러스터별로 다른 피쳐를 추가해 모델을 생성하는 방식</p> <p>그래프를 보면 전기량 사용의 경향성이 나타났습니다.</p>



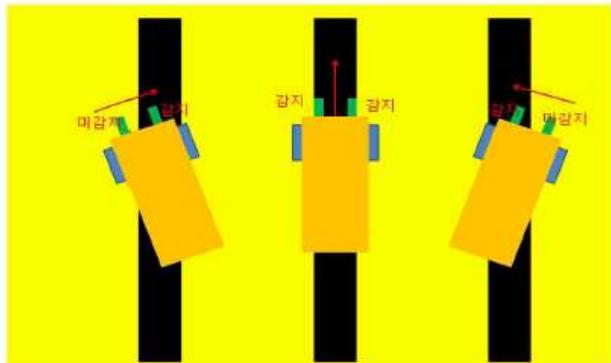
프로젝트명	스마트 농업을 위한 자율주행 레이저 제초 로봇	수행기간	2024/03/16 ~ 2024/06/16
팀원	주재현 안영록 한성모 이종연연	GitHub url	x
역할 및 참여범위	-역할 : YOLO 모델, 아두이노 코딩, 회로 설계, 라즈베리파이 연동, 차체 제작		
프로젝트 개요	<p>현재 농업은 여러 문제에 직면해있습니다. 기후변화, 고령화, 환경파괴, 농촌 소멸, 식량 위기 등이 그 예입니다. 농업은 높은 노동 강도, 부족한 인력, 기후변화로 인한 농작물 피해 등 많은 어려움을 겪고 있습니다. 따라서 현세대와 미래 세대의 먹거리를 책임지는 농업은 '성장'과 '지속성' 측면에서 보호하고 발전시켜야 합니다.</p> <p>고령화와 저출산으로 인한 농촌의 생산 가능 인구 감소는 현대 사회에서 중요한 문제로 대두되고 있습니다. 이에 따라 농촌진흥청은 스마트 농업기술을 현장에 구현하기 위한 10대 핵심 추진 과제를 선정하는 등 스마트 농업에 대한 연구를 활발히 진행하고 있으며, 노동력 대체를 위한 무인화의 필요성을 강조하고 있습니다.</p> <p>따라서 본 프로젝트는 적외선 센서를 이용한 라인 트레이싱(Line Tracing)과 YOLO_V8 알고리즘을 이용한 객체 인식을 통해 잡초를 탐지하고, 레이저를 사용하여 잡초를 제초하는 로봇을 제작합니다. 기존의 화학 제초제 방식 대신 레이저를 이용한 제초 방식을 도입함으로써, 농촌의 노동력 부족 문제를 해결하고 친환경적인 농업을 촉진하고자 합니다.</p>		
개발 PLATFORM	VScode / arduino / Raspberry Pi		
각별히 애썼던 부분	<p>라즈베리 파이 연동에 있어서 우리가 설계했던 성능보다 많이 떨어지는 성능에 맞추어 자율주행을 개선하다보니 최적화를 하는 점에 있어 가장 많은 신경을 썼습니다. ( 카메라로 인식하는게 delay가 발생하여 어떻게 하면 delay를 줄이고 실시간으로 객체를 인식시킬지)</p> <p>무선으로 로봇이 동작해야 하므로 UPS를 선정하는데에 있어서도 많은 고민을 하고 또한 모터들의 정격, 다른 센서들의 정격을 생각하여 시행착오를 겪으면서 선정하였습니다.</p>		
<p><b>2.1.2 OpenCV, 적외선 센서를 이용한 Line Tracing</b></p> <p>이 절에서는 자율주행 레이저 제초로봇의 가장 중요한 과제인 자율주행에 사용된 알고리즘에 대해 설명한다.</p> <p>자율주행을 할 수 있는 방법에는 크게 두가지가 존재한다. 그 중 첫번 째는 OpenCV를 활용해 실시간 이미지를 받아와 픽셀 값이 한 쪽으로 급격히 변하는 부분이나 사물과 사물, 사물과 배경의 경계선인 엣지를 검출하여 검출된 선 성분에서 평균직선을 그려 월 스티어링 각도를 얻는 방식이다.</p>  <p>[그림 2-3] Edge 검출</p> <p>Canny 알고리즘을 이용하여 Edge를 검출하고 Region Of Interest 방식을 사용하여 원하는 범위의 프레임을 받아 실제 차선이 구부러진 각도를 파악하였습니다.</p>			



[그림 2-5] 적외선 센서

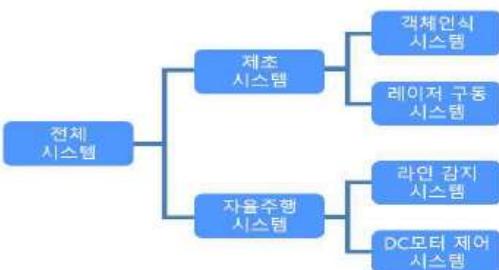
적외선 센서는 발광부와 수광부가 있다. 발광부에서 적외선을 쏘게 되면 물체에 반사되어서 수광부로 들어오게 된다. 하지만 검은색은 빛을 흡수하여 수광부에 적외선이 도달하지 못하게 된다. 이 원리를 이용하여 검은 선과 흰 선을 구분해 준다.

적외선 센서를 이용한 Line Tracing 방식을 채택하여 자율주행을 진행하였습니다.



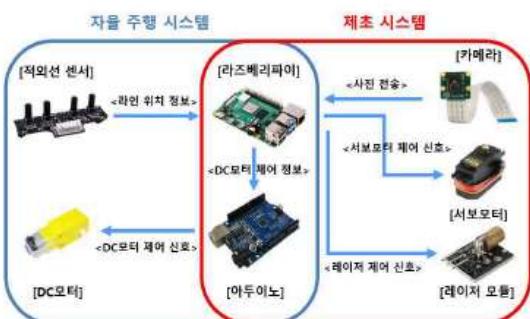
[그림 2-6] 적외선 센서를 이용한 Line Tracing

**레이저 구동 시스템:** 자체 인식 시스템에서 제공된 잡초의 위치 정보를 기반으로 서보 모터를 제어한다. 서보 모터는 레이저의 방향을 정밀하게 조정하여 잡초에 정확히 맞추도록 한다. 이후 레이저를 조사하여 잡초를 제거한다. 이 시스템은 잡초를 효과적으로 제거할 수 있도록 하여, 농작물의 건강을 유지하고 생산성을 높이는 데 기여한다.



[그림 2-7] 시스템 구성

시스템 구성 방식 및 구성요소입니다.



#### - 레이저 구동 시스템 동작 원리

객체 인식 시스템에서 잡초가 인식되면, 레이저 구동 시스템이 동작한다. 레이저 구동 시스템은 서보 모터를 조절하여 화면상의 레이저 위치와 잡초의 위치가 일치하도록 한다. 이를 위해 레이저 위치와 잡초 위치 간의 각도 차이를 계산한다.

화면 상의 각도를 추정하여 계산한 것이기 때문에 일부 오차가 발생할 수 있다. 이러한 오차를 해결하기 위해 비례 제어기(P제어)를 사용하여 레이저가 정확하게 잡초를 향하게 한다. P제어는 실제 각도와 목표 각도 간의 차이를 기반으로 서보 모터의 각도를 미세하게 조정하여, 레이저의 정확성과 안정성을 높인다.

이 과정을 통해 레이저 구동 시스템은 잡초를 효과적으로 겨냥하고 제거할 수 있다.



[그림 2-20] 서보모터 구동



[그림 2-21] 레이저 구동

잡초를 인식하고 목표를 조준하는 알고리즘을 촬영한 장면입니다.



실제 시연 장면

프로젝트명	전시상황에서 군수품 수송로 확보를 위한 실시간 경로 탐색 시스템 개발	수행기간	2024/07/14 ~ 2024/08/24
팀원	주재현 김지민 김나영 김태영 김태훈 신경민 하은지	Miricanvas url	<a href="https://www.miricanvas.com/v/152ltya">https://www.miricanvas.com/v/152ltya</a>
역할 및 참여범위	-역할 : [백엔드 메인] YOLO 모델 설계 및 분석 XGBoost모델 설계, D*Lite, RTAA* 알고리즘 구현, 분석보고서 작성, DB연동, UI 설계, 서비스 배포		

## 1. 배경

전쟁의 향방을 바꾼 보급로 확보 중요

- 2022년 러·우 전쟁에서 우크라이나는 드론과 위성 영상을 활용해 도로 파괴 현황과 적 위치를 파악하고, 보급망을 유지하여 방어에 성공함.

장애물/지형적 요소를 반영한 경로탐색 시스템 필요

- 현재 국내에서 개발된 대부분의 전시상황 대응 시스템들은 유·무인 형태로 사전에 설정된 경로를 따라 물자 수송, 환자 이동, 장애물 제거 등의 기능을 제공하고 있음. (예. K-CEV)
- 일부 시스템은 장애물 발생 시 국지적인 우회 경로 탐색 기능만 제공하며, 전체 경로를 전략적으로 재설계하는 기능은 부재함. (예. Arion-SMET)

## 2. 분석 지역 선정 – 은평구·서대문구·마포구

공간·정책적 연계성

- 프로젝트  
개요
- 서울시 생활권 계획상 동일한 서북권에 속하는 권역임.
  - 공동 개발·재생 계획 및 교통망 설계가 추진되어 정책적 연속성과 비교 가능성이 높음.
- 교통축선의 전략성
- 통일로·자유로·제2자유로를 통한 파주–서울 연결축 형성됨.
  - 평시 물류·통근 중심축, 유사 시 병력·물자 수송 및 보급로로 기능함.

지휘·통제 거점과의 연계성

- 용산 국가위기관리센터와 전방 지역을 연결하는 전략적 접근 통로임.
- 유사 시 병참선 확보와 최종 방어선 역할 수행 가능성이 높음.

## 3. 목적

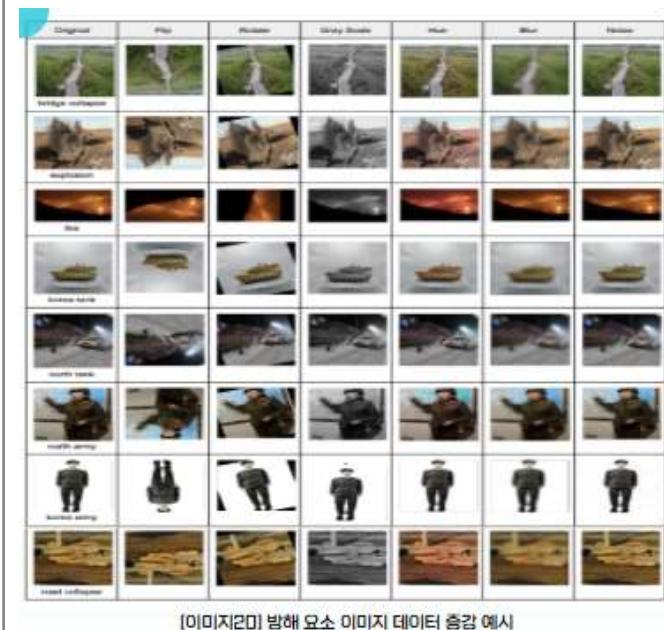
전시 상황에서 군수품 보급의 효율성을 확보하기 위해, 드론을 활용해 경로상 방해 요소를 실시간으로 감지하고, 이를 회피하는 최적 경로를 신속히 제공하는 시스템을 구축하는 것을 목표로 함.

## 4. 주요 적용 기관

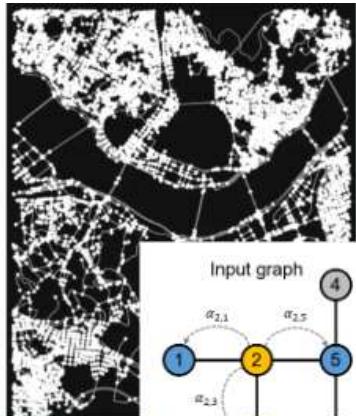
- 국군수송사령부, 군수사령부, 육·해·공군 본부, 연합사령부

개발 PLATFORM	VScode / Flask / jupyter notebook / mysql / render
각별히 애썼던 부분	알고리즘을 구현하는데에 있어서 코드를 시뮬레이션 형태로 변환하는데 많은 신경을 들였고, 웹 서버에서 시뮬레이션이 구현되도록 변형하는 것이 가장 어려웠습니다. 또한 osmnx의 node 수가 굉장히 많아서 로딩 속도나 반응속도가 늦어 웹 서버에서 구현이 버벅거리는 것을 해결하는 것에 신경을 많이 썼습니다.

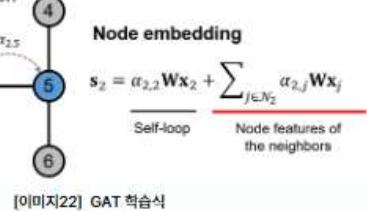
<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>acdnt_no</th><th>x_crdnt</th><th>y_crdnt</th></tr> </thead> <tbody> <tr><td>2024081200100100</td><td>950141</td><td>1953681</td></tr> <tr><td>2022010200100020</td><td>950223</td><td>1953517</td></tr> <tr><td>2019052800100040</td><td>950092</td><td>1951215</td></tr> <tr><td>2016092400100160</td><td>951527</td><td>1953700</td></tr> <tr><td>2014112200100080</td><td>948865</td><td>1953120</td></tr> <tr><td>2013051400100290</td><td>952135</td><td>1952928</td></tr> <tr><td>2010031600100060</td><td>952884</td><td>1951944</td></tr> <tr><td>2007010200100040</td><td>952099</td><td>1952567</td></tr> </tbody> </table> <p style="text-align: center;">[표2] 교통사고 좌표 데이터 수집 (총 8,187개)</p>													acdnt_no	x_crdnt	y_crdnt	2024081200100100	950141	1953681	2022010200100020	950223	1953517	2019052800100040	950092	1951215	2016092400100160	951527	1953700	2014112200100080	948865	1953120	2013051400100290	952135	1952928	2010031600100060	952884	1951944	2007010200100040	952099	1952567
acdnt_no	x_crdnt	y_crdnt																																					
2024081200100100	950141	1953681																																					
2022010200100020	950223	1953517																																					
2019052800100040	950092	1951215																																					
2016092400100160	951527	1953700																																					
2014112200100080	948865	1953120																																					
2013051400100290	952135	1952928																																					
2010031600100060	952884	1951944																																					
2007010200100040	952099	1952567																																					



- 데이터셋 규모: 총 9,389개 이미지
- 이미지 리사이징: 모든 입력 이미지를 640×640 픽셀 크기로 통일
- 정규화: 픽셀 값을 [0, 1] 범위로 스케일링하여 학습 안정성 향상
- 데이터 증강
  - 기하학적 변환: 좌우/상하 반전, 90° 회전 → 다양한 시점·방향 대응
  - 색상 변환: 일부 이미지를 그레이스케일 변환, 색조(Hue) -20°~+20° 조정
  - 화질 변환: 최대 4px 블러, 최대 1.05% 노이즈 추가→ 화질 저하 상황에서도 견고한 성능 유지
- 라벨링 작업: 오픈 데이터셋이 아닌 경우, **Labelling** 등 어노테이션 툴을 활용해 **Bounding Box** 및 클래스 라벨 `wlwd`

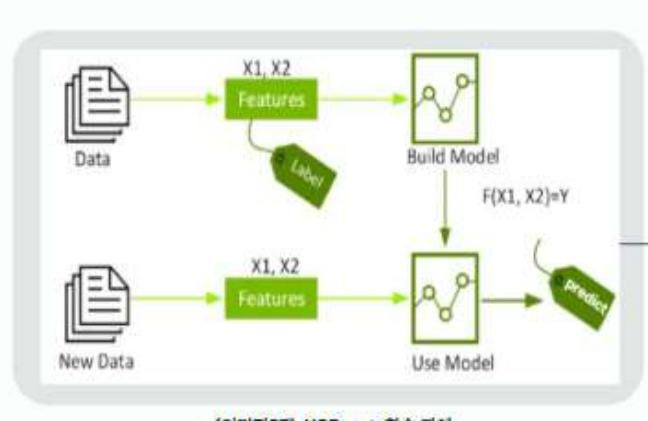


[이미지21] OSMNX 예시 이미지



#### GAT(graph attention networks)모델 학습

- 분석 지역: 은평구·서대문구·마포구 도로망 활용
- 분석 알고리즘: GAT
- 피처(X): 도로 속성 및 지형 정보 17가지 (차선 수, 길이, 경사도, 고도, 표면 상태 등)
- 타겟(y): XGBoost로 산출한 '엣지별 예측 사고율'
- 데이터셋 분리: 훈련 B : 검증 1: 테스트 1
  - Attention 메커니즘으로 중요 도로에 더 큰 가중치 반영
- 출력: 도로별 위험도 점수 산출
- 활용: 최종 비용 계산식에 도로 기본 위험도로 반영됨



#### XGBoost 모델 학습

##### [실험 설정]

- 분석 지역: 은평구·서대문구·마포구 도로망 활용
- 분석 알고리즘 : XGBoost
- 피처(X): 도로 속성 및 지형 정보 17가지 (차선 수, 길이, 경사도, 고도, 표면 상태 등)
- 타겟(y): 엣지별 사고율 (0~1 연속형)
- 데이터셋 분리: 훈련 B : 검증 1: 테스트 1
- 성능 평가 지표: RMSE, MAE

##### [배경 설정]

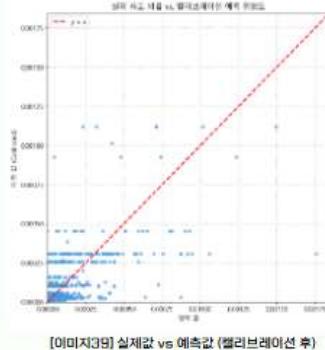
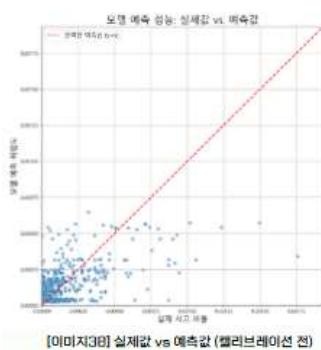
- 실제 전시 상황의 도로 위험도 데이터 부재
- 교통사고 데이터 활용하여 도로 구조적 요인 기반

	<p>사고율을 전시 상황 위험도로 추정 • GAT 모델 학습의 레이블(y 값)로 사용</p>																																																														
<table border="1"> <thead> <tr> <th>방해 유형별</th> <th>가중치</th> <th>간단설명</th> <th>출처</th> <th>반경(m)</th> <th>간단설명</th> <th>출처</th> </tr> </thead> <tbody> <tr> <td>교량 파괴 (bridge collapse)</td> <td>100,000,000</td> <td>여기 시 차량 통행 불가 → 경로에서 안전히 통과</td> <td>IMBC 보도, CMU 논문</td> <td>10</td> <td>차량 통행 불가 → 경로에서 안전히 통과</td> <td>없음</td> </tr> <tr> <td>도로 붕괴 (road collapse)</td> <td>100,000,000</td> <td>도로 가는 산성 → 경로/소방이 즉시 통제</td> <td>IMBC 보도</td> <td>10</td> <td>차량 통행 불가 → 경로에서 안전히 통과</td> <td>없음</td> </tr> <tr> <td>탱크 (tank)</td> <td>10,000</td> <td>주제전차(M1A1) 2.3km 교전 가능, 탱크원 2.9km 사거리 → 차량 적 위협</td> <td>M1 Abrams 위키, FGM-148 Javelin 위키</td> <td>300</td> <td>북한군 전차 사거리 참고</td> <td>[나무위키] 조선민족군 북군 참조</td> </tr> <tr> <td>적군 (enemies)</td> <td>5,000</td> <td>보병 소총(M16: 460m, AK-47) 및 RPG-7(300m) 위협 → 중간 위협, 인디렉트 대화</td> <td>M16 위키, RPG-7 위키</td> <td>200</td> <td>북한군 보병 전차 사거리 참고</td> <td>[나무위키] 조선민족군 북군 참조</td> </tr> <tr> <td>화재 (fire)</td> <td>3,000</td> <td>시내 저지 및 열 폭발, 도로 폭발 가능 → 맹그-폭발물(200m 반경)</td> <td>Wildfire Preparedness</td> <td>200</td> <td>증발물 기준의 최소값</td> <td>참고문헌 참고산림화 산불 예방 보상(2024)</td> </tr> <tr> <td>폭발IED (explosion)</td> <td>10,000</td> <td>도로 폭발 폭발물로 차량 구조물 파괴</td> <td>DHS IED Fact Sheet</td> <td>300</td> <td>차량 폭발 기준</td> <td>논문 참고 CNS, North Korea's Nuclear Threats: Post-Test Analysis (2019)</td> </tr> <tr> <td>철조망(barbed_wire)</td> <td>100,000,000</td> <td>경로에서 안전히 통과</td> <td>없음</td> <td>10</td> <td>차량 통행 불가 → 경로에서 안전히 통과</td> <td>없음</td> </tr> <tr> <td>낙석(rocksfall)</td> <td>100,000,000</td> <td>경로에서 안전히 통과</td> <td>없음</td> <td>10</td> <td>차량 통행 불가 → 경로에서 안전히 통과</td> <td>없음 27</td> </tr> </tbody> </table>	방해 유형별	가중치	간단설명	출처	반경(m)	간단설명	출처	교량 파괴 (bridge collapse)	100,000,000	여기 시 차량 통행 불가 → 경로에서 안전히 통과	IMBC 보도, CMU 논문	10	차량 통행 불가 → 경로에서 안전히 통과	없음	도로 붕괴 (road collapse)	100,000,000	도로 가는 산성 → 경로/소방이 즉시 통제	IMBC 보도	10	차량 통행 불가 → 경로에서 안전히 통과	없음	탱크 (tank)	10,000	주제전차(M1A1) 2.3km 교전 가능, 탱크원 2.9km 사거리 → 차량 적 위협	M1 Abrams 위키, FGM-148 Javelin 위키	300	북한군 전차 사거리 참고	[나무위키] 조선민족군 북군 참조	적군 (enemies)	5,000	보병 소총(M16: 460m, AK-47) 및 RPG-7(300m) 위협 → 중간 위협, 인디렉트 대화	M16 위키, RPG-7 위키	200	북한군 보병 전차 사거리 참고	[나무위키] 조선민족군 북군 참조	화재 (fire)	3,000	시내 저지 및 열 폭발, 도로 폭발 가능 → 맹그-폭발물(200m 반경)	Wildfire Preparedness	200	증발물 기준의 최소값	참고문헌 참고산림화 산불 예방 보상(2024)	폭발IED (explosion)	10,000	도로 폭발 폭발물로 차량 구조물 파괴	DHS IED Fact Sheet	300	차량 폭발 기준	논문 참고 CNS, North Korea's Nuclear Threats: Post-Test Analysis (2019)	철조망(barbed_wire)	100,000,000	경로에서 안전히 통과	없음	10	차량 통행 불가 → 경로에서 안전히 통과	없음	낙석(rocksfall)	100,000,000	경로에서 안전히 통과	없음	10	차량 통행 불가 → 경로에서 안전히 통과	없음 27
방해 유형별	가중치	간단설명	출처	반경(m)	간단설명	출처																																																									
교량 파괴 (bridge collapse)	100,000,000	여기 시 차량 통행 불가 → 경로에서 안전히 통과	IMBC 보도, CMU 논문	10	차량 통행 불가 → 경로에서 안전히 통과	없음																																																									
도로 붕괴 (road collapse)	100,000,000	도로 가는 산성 → 경로/소방이 즉시 통제	IMBC 보도	10	차량 통행 불가 → 경로에서 안전히 통과	없음																																																									
탱크 (tank)	10,000	주제전차(M1A1) 2.3km 교전 가능, 탱크원 2.9km 사거리 → 차량 적 위협	M1 Abrams 위키, FGM-148 Javelin 위키	300	북한군 전차 사거리 참고	[나무위키] 조선민족군 북군 참조																																																									
적군 (enemies)	5,000	보병 소총(M16: 460m, AK-47) 및 RPG-7(300m) 위협 → 중간 위협, 인디렉트 대화	M16 위키, RPG-7 위키	200	북한군 보병 전차 사거리 참고	[나무위키] 조선민족군 북군 참조																																																									
화재 (fire)	3,000	시내 저지 및 열 폭발, 도로 폭발 가능 → 맹그-폭발물(200m 반경)	Wildfire Preparedness	200	증발물 기준의 최소값	참고문헌 참고산림화 산불 예방 보상(2024)																																																									
폭발IED (explosion)	10,000	도로 폭발 폭발물로 차량 구조물 파괴	DHS IED Fact Sheet	300	차량 폭발 기준	논문 참고 CNS, North Korea's Nuclear Threats: Post-Test Analysis (2019)																																																									
철조망(barbed_wire)	100,000,000	경로에서 안전히 통과	없음	10	차량 통행 불가 → 경로에서 안전히 통과	없음																																																									
낙석(rocksfall)	100,000,000	경로에서 안전히 통과	없음	10	차량 통행 불가 → 경로에서 안전히 통과	없음 27																																																									
<p>[이미지30] YOLO 모델 학습 구조</p>	<ul style="list-style-type: none"> <li>입력(X): 8 개 클래스 (화재, 도로 붕괴, 다리 붕괴, 폭발, 한국군, 북한군, 한국 전차, 북한 전차)</li> <li>타겟(y): 각 이미지 내 방해 요소 위치 (바운딩 박스 좌표) + 클래스 라벨</li> <li>학습: 훈련 9: 검증 0.5 : 테스트 0.5 세트 분리, 하이퍼파라미터 조정하며 성능 최적화</li> <li>출력:       <ul style="list-style-type: none"> <li>방해 요소 위치 (Bounding Box)</li> <li>종류 (Class Label)</li> <li>개수 (Object Count)</li> <li>신뢰도 점수 (예: 화재 92%)</li> </ul> </li> <li>활용:       <ul style="list-style-type: none"> <li>드론 영상에서 실시간 객체 탐지 수행</li> <li>위치·종류·개수에 따라 방해 요소 점수(가중치)를 계산 → 최종 비용 계산식에 반영</li> </ul> </li> </ul>																																																														
<h2>YOLO탐지 기반 방해요소 점수 (Riskscore) 계산식</h2> <p>RiskScore(e) = <math>d(\text{road\_collapse}) \times 100,000,000</math>  <math>+ d(\text{bridge\_collapse}) \times 100,000,000</math>  <math>+ d(\text{tanks}) \times n1 \times 10,000</math>  <math>+ d(\text{enemies}) \times n2 \times 5,000</math>  <math>+ d(\text{fire}) \times n3 \times 2,000</math>  <math>+ d(\text{explosion}) \times n4 \times 10,000</math>  <math>+ d(\text{barbed\_wire}) \times 100,000,000</math>  <math>+ d(\text{rockfall}) \times 100,000,000</math></p> <p>[이미지31] Riskscore 계산식 이미지</p>	<ul style="list-style-type: none"> <li><math>d()</math>: 해당 위험 요소 탐지 여부</li> <li><math>n</math>: YOLO에서 탐지된 객체 개수</li> <li>여러 위험이 동시에 탐지되면, 가중치 합산 최종 RiskScore에 반영</li> <li><math>C_{\text{base}}(e)</math>: 도로 길이 비용</li> <li><math>\text{RiskScore}(e)</math>: YOLO 탐지 기반 방해요소 점수 (위치·종류·개수 반영)</li> <li><math>C_{\text{GAT}}(e)</math>: GAT 모델이 예측한 도로 기본 위험도</li> </ul>																																																														

## 각 도로(엣지)의 최종 비용( $C_{total}$ ) 계산식

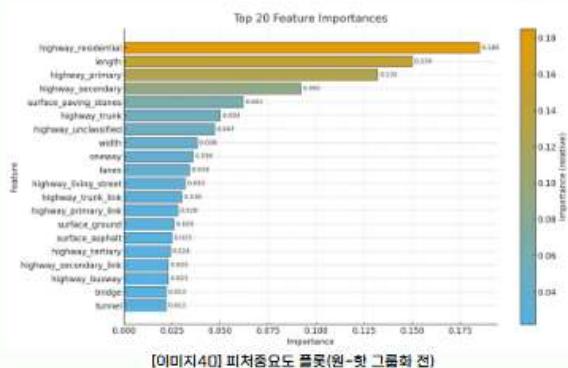
$$C_{total}(e) = C_{base}(e) + RiskScore(e) + C_{GAT}(e)$$

### ① 실제값-예측값 산점도 캘리브레이션 전 / 후

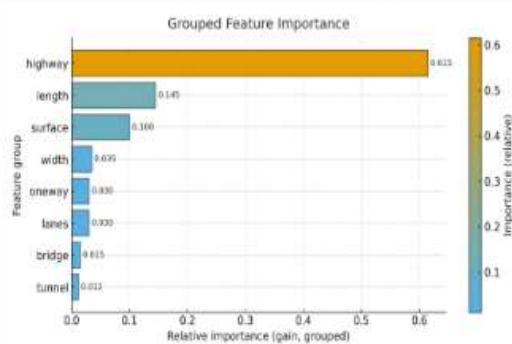


- XGBoost 산점도 개념
- 예측된 위험도와 실제 위험도를 비교
- XGBoost 전/후 비교
  - 캘리브레이션 전: 저위험 과대, 고위험 과소
  - 캘리브레이션 후: 고위험 구간 보정으로 기준선 근접
- 해석
  - 캘리브레이션으로 예측 균형성 전반적 향상
  - 데이터 희소성으로 인해 여전히 불안정한 구간 존재
- 의미 및 향후 과제
  - 고위험 구간 상향 보정 → 캘리브레이션 효과 있음
  - 고위험 데이터 보강 필요

### ② 피처 중요도(Feature Importance) 플롯

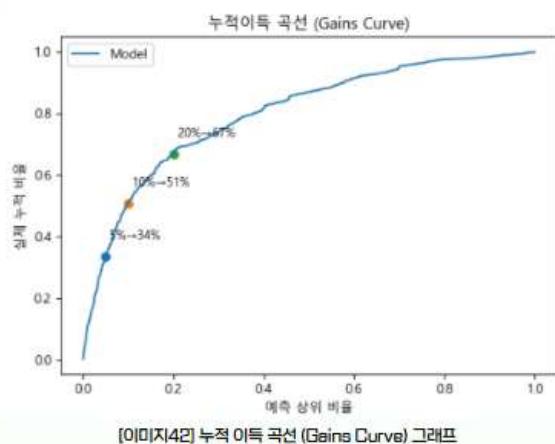


- 피처중요도 플롯 (원-핫 그룹화 전)
  - 모델이 실제값에 더 가까운 예측을 하도록 가장 많이 도움을 준 정보 순으로 상위 20개를 보여줌
  - 특히 도로 등급(highway), 길이(length), 차로·일방통행(lanes/oneway)의 기여가 큼



- 피처중요도 플롯 (원-핫 그룹화 후)
  - 원-핫 인코딩된 highway\_\*, surface\_\*를 접두사 별로 그룹화하여 피처 변수 '자체'의 기여를 추정
    - highway 전체 기여가 ≈60%로 가장 큼
    - 2위: length, 3위: surface

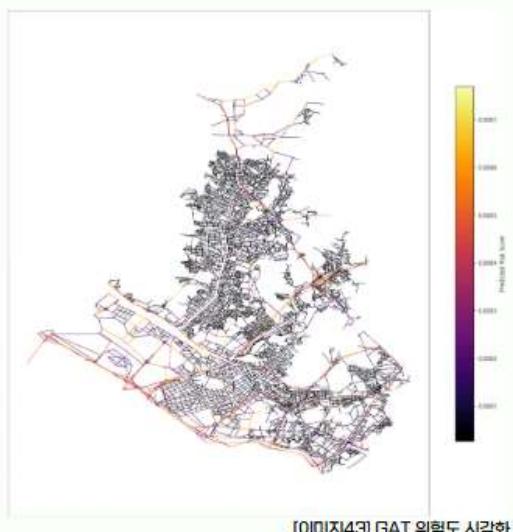
### ③ 누적 이득 곡선(Gains Curve)



[이미지42] 누적 이득 곡선 (Gains Curve) 그래프

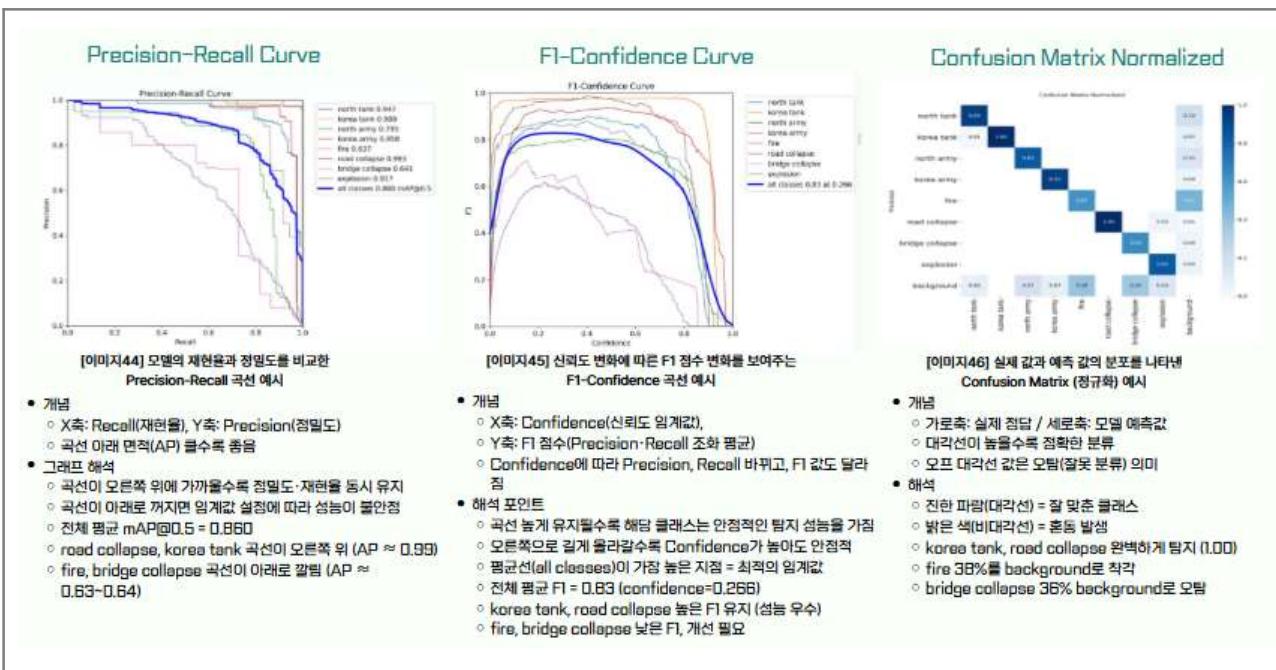
- 누적 이득 곡선(Gains Curve)
- 가로축: 예측 상위 k% 도로, 세로축: 실제 사고 비율
  - 모델 선이 기준선보다 높으면 효과적으로 고위험 도로를 식별한 것
- 핵심 숫자 정리
  - 상위 5% → 실제 34%
  - 상위 10% → 실제 51%
  - 상위 20% → 실제 67%
- 모델 강점 강조
  - 상위 10% 대상으로만 관리해도 전체 사고의 절반 이상을 커버 가능
- 제한점 및 향후 과제
  - 상위 이후 커브 평탄 → 희소 고위험 구간 미반영
  - 더 다양한 데이터 확보 및 희소 구간 보강 필요

### ④ GAT 위험도 시각화



[이미지43] GAT 위험도 시각화

- 시각화
  - 보라색: 낮은 예측 위험도
  - 노란색: 높은 예측 위험도
- 주요 결과
  - 학점 교통축(간선도로, 순환로)은 노란색 → 높은 위험 표시
  - 주거 이면도로는 보라색 → 낮은 위험 표시
- 의미 있는 해석
  - 모델이 도로 유형과 통행량을 학습해 위험 구간을 구분
  - 실제 도로 특성과 예측 일치
- 모델 성능 인사이트
  - 상대적 위험 구분은 유의미함
  - 주의구간 후보를 좁히는데 도움됨



## 하이퍼파라미터 튜닝

파라미터	튜닝 version 1	튜닝 version 2
epoch	50	50
imagez	640	640
batch	8, 16, 32	32
optimizer	SGD, AdamW	SGD
lr0	0.0001, 0.01 사이	0.0025, 0.0065 사이
lrf	0.01	0.01
patience	20	20
close_moving	5	5
momentum	0.8, 0.99 사이	0.85, 0.94 사이
weight decay	0.0, 0.001 사이	0.0005, 0.00095 사이
method	bayes	bayes

[표10] 1차/2차 하이퍼파라미터 튜닝 설정값

- 총 2단계 하이퍼파라미터 튜닝
  - 1차 튜닝: Optimizer, Batch Size (큰 범위 조정)
  - 2차 튜닝: lr0, Momentum, Weight Decay (세부 조정)

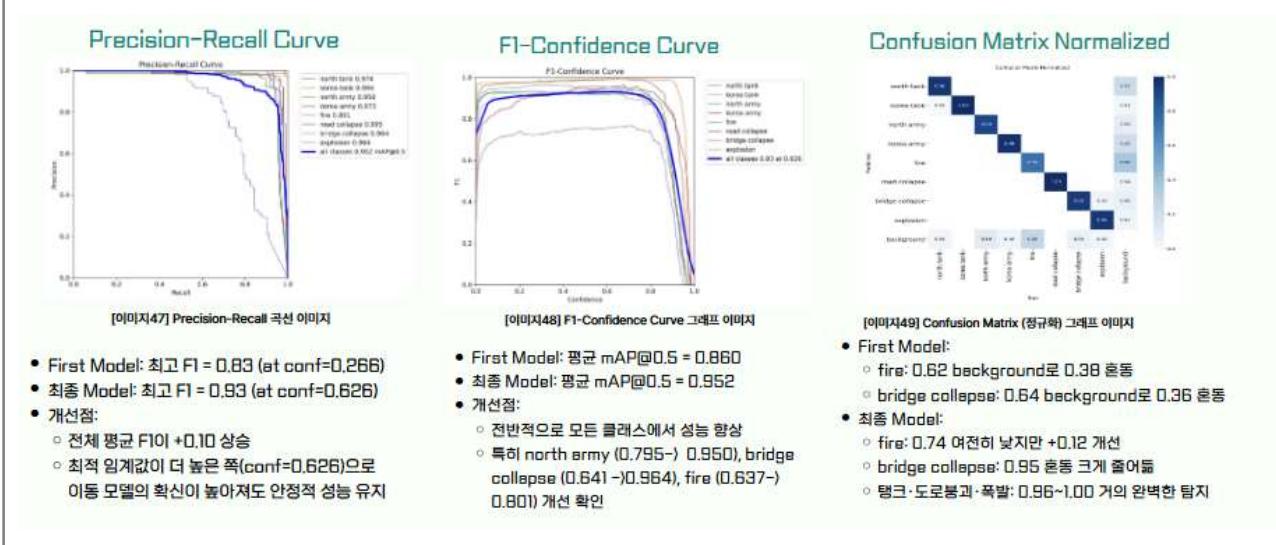
## 튜닝 결과: 최종 모델 성능

	batch	epochs	lr0	momentum	optimizer	weight_decay	mAP50	mAP50-95
1	32	60	0.004901	0.95105	SGD	0.0000240	0.89	0.653
2	32	50	0.003052	0.92099	SGD	0.00007431	0.891	0.649
3	8	60	0.0056642	0.9158	AdamW	0.00002511	0.871	0.603
4	8	50	0.0042499	0.970	SGD	0.0000902	0.883	0.636
5	16	50	0.0051795	0.92448	SGD	0.00006985	0.899	0.652
6	32	50	0.0040217	0.89518	SGD	0.00006948	0.911	0.732
7	32	60	0.0061958	0.9563	SGD	0.00007256	0.901	0.723
8	32	50	0.00151	0.87873	-	0.00006174	0.912	0.798
9	32	60	0.0027800	0.85082	SGD	0.00006320	0.932	0.766
10	32	50	0.0053359	0.91621	SGD	0.00005426	0.943	0.772

[표11] 하이퍼파라미터 튜닝 결과 비교

### 총 10개 모델 실험 → 모델 8 최종 선택

- mAP50 = 0.952
- mAP50-95 = 0.798



## 알고리즘 분석 표준 기준표

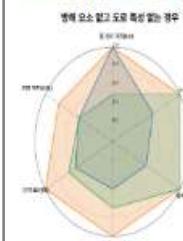
평가 항목	평가 설명
총 경로 거리(Km)	차량이 실제로 이동한 경로의 총 길이
소요 시간(ms)	출발지→도착지까지 실행 전체 경과시간
탐색 시간(ms)	알고리즘이 경로 계획에 실제로 쓴 누적 연산 시간
탐색 노드 수	알고리즘이 탐색/갱신 과정에서 확장한 총 노드 수
탐색 효율(%)	실제 지나간 노드 수 대비 탐색에서 확장한 노드 수 비율
위험 회피율(%)	실제 이동한 엣지 중 위험 가중치가 걸린 엣지를 피한 비율

[표2] 알고리즘 분석 표준 기준표

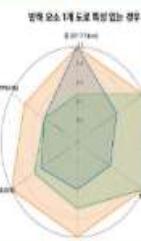
- 표준 기준 참고 논문 리스트
- [Q-learning based on strategic artificial potential field for path planning enabling concealment and cover in ground battlefield environments]
- A multi-algorithm pathfinding method\_Exploiting

## 도로 특성 없음

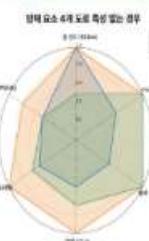
### 방해요소 0개



### 방해요소 1개

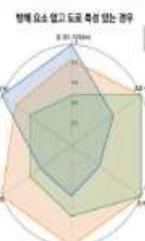


### 방해요소 4개

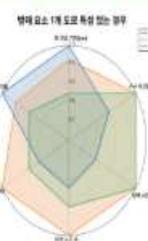


## 도로 특성 있음

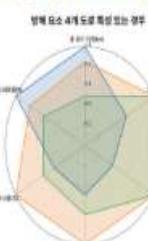
### 방해요소 0개



### 방해요소 1개



### 방해요소 4개



#### 특성

- CCH+A\*: 0.5 수준으로 보통, 총 경로 거리 1.0로 최단 거리 탐색에 가장 강점
- D Lite\*: 회피율도 0.8로 우수, 이 외 모두 1.0으로 전반적으로 최적 성능
- RTAA\*: 탐색 시간 1.0으로 속도  
위험 회피율 0.6, 경로거리 0.6, 탐색 효율 0.7은 상대적으로 낮음
- 종합평가
  - 안전(회피율): D Lite\* > RTAA\* ≥ CCH+A\*
  - 종합 성능: D Lite\*

#### 특성

- CCH+A\*: 위험 회피율 약 1.0수준으, 총 경로 거리 1.0외로 높은 수치지만, 이 외 수치는 다 낮음
- D Lite\*: 회피율도 0.8로 이 외 다 모두 1.0으로 전반적으로 최적 성능
- RTAA\*: 탐색 시간 1.0으로 속도  
위험 회피율 0.6, 경로거리 0.6, 탐색 효율 0.7은 상대적으로 낮음
- 종합 평가
  - 안전(회피율): CCH+A\* ≥ D Lite\* > RTAA\*
  - 안전, 최단거리 확보 목적: CCH+A\*
  - 종합 성능: D Lite\*

© 2019 KAIST. All Rights Reserved.

지표	F값	p-value	결과
총 경로 거리 (km)	13.52	0.03	유의한 차이
소요 시간 (ms)	206.16	0.00	유의한 차이
탐색 시간 (ms)	24.08	0.01	유의한 차이
탐색 노드 수	12.61	0.03	유의한 차이
탐색 효율(%)	11.05	0.04	유의한 차이

[표14] ANOVA 분석 결과

- 방해 요소가 4개, 도로 특성(GAT)을 반영하지 않은 경우
- 가설 설정:
  - $H_0$  (귀무가설): 모든 그룹 평균이 동일
  - $H_1$  (대립가설): 적어도 하나 평균이 다름
- 전제 조건:
  - 독립성, 정규성, 등분산성 만족 여부 확인
- 계산: SS\_between, SS\_within MS, F-통계량  
→ P-값 산출
- 해석:
  - 모든 지표에서 p-value ( $<0.05$ 이므로, 귀무가설( $H_0$ )을 기각)
  - 모든 지표에서 알고리즘 간 평균 차이가 통계적으로 유의미함을 확인
  - 소요 시간(ms)\*\*은  $F=206.16$ ,  $p<0.001$ 로 가장 큰 차이를 보여, 알고리즘별 속도 성능 차이가 확연히 존재함을 의미

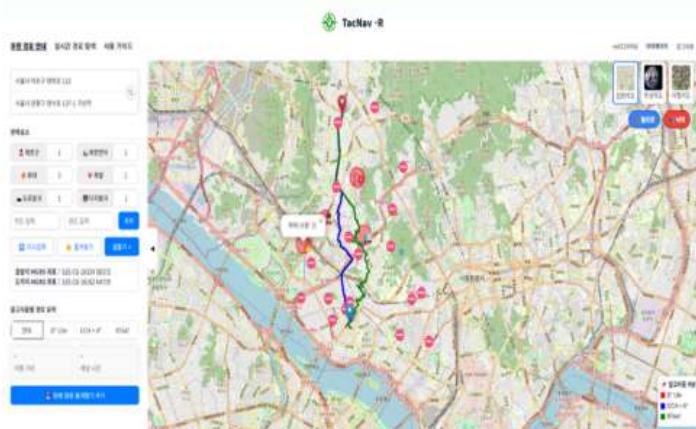


[이미지56, 57] 시스템 첫 화면

## 회원가입 및 로그인

- 좌측 화면: 기존 사용자가 아이디와 비밀번호를 입력하여 로그인할 수 있는 화면
- 우측 화면: 신규 사용자가 이름, 이메일, 비밀번호, 연락처 등의 정보를 입력하고 계정을 생성할 수 있는 회원가입 화면

## ① 추천 경로 안내

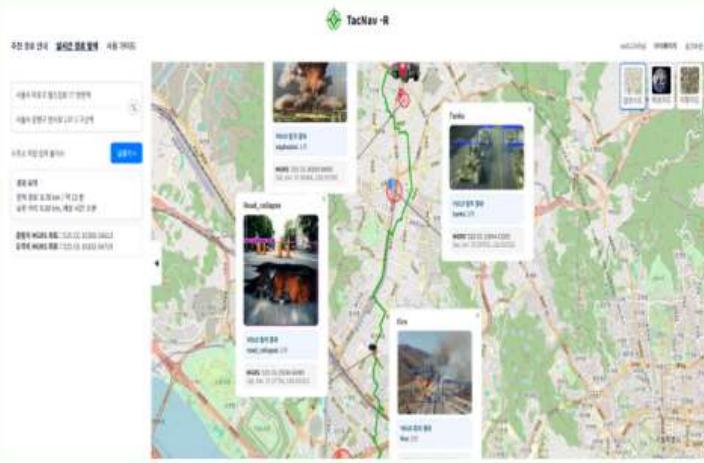


[이미지58] 사용자 맞춤형 경로 탐색 및 추천

## 사용자 맞춤형 경로 탐색 및 추천

- 페이지 설명: 방해 요소 회피 기반 알고리즘별 최적 경로 비교 기능 시스템
- 1. 사용자가 출발지/목적지 입력
- 2. 방해 요소를 지도 클릭으로 실시간 추가 가능
- 3. 좌표를 통해 방해 요소 추가 가능
- 4. [경로찾기] 버튼으로 탐색 실행
- 5. D\*Lite / CCH / ACO 알고리즘을 활용해 경로 검색 후 성능 비교 분석
- 6. 탐색된 결과를 지도에서 직관적으로 비교하여 가장 적합한 경로 선택 가능

## ② 실시간 경로 탐색



[이미지59] 실시간 장애물 탐색 및 경로 탐색

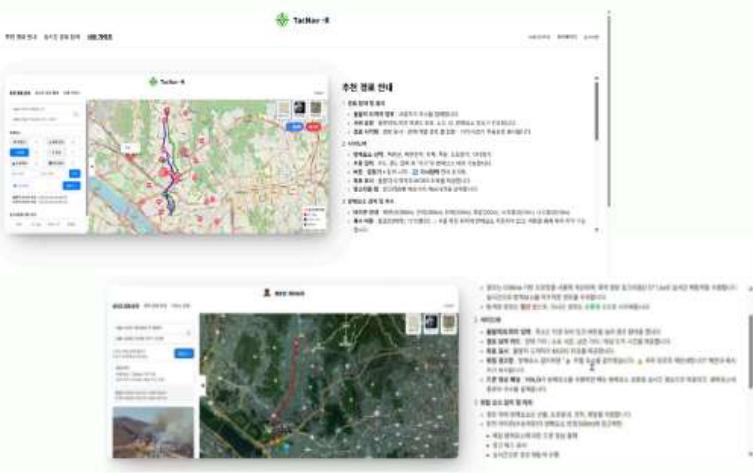
## 실시간 장애물 탐색

### 및 경로 탐색

- 페이지 설명 : 실시간 위험 감지 시 팝업 영상으로 경고 및 즉시 안전 경로로 자동 재탐색

1. 지정된 출발지/목적지 최적 경로 안내
2. 방해 요소 탐지 시, 드론이 실시간 영상 제공 및 팝업창 알림
3. 위험 지역 진입 시, 팝업창에 방해 요소명·군사 좌표·위경도·갯수 표시

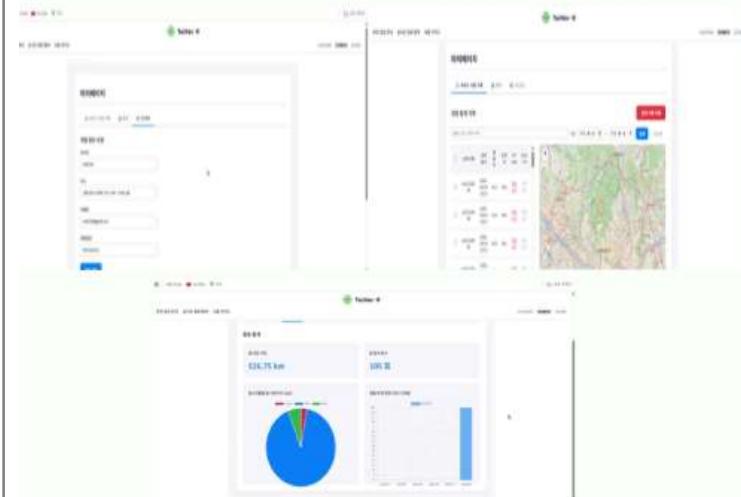
## ① 사용가이드



## 사용가이드 설명서(1)

- 추천 경로 안내 페이지 설명서
  - 해당 페이지 안내
  - 사용방법 A-Z까지 설명
- 실시간 경로 탐색 페이지 설명서
  - 해당 페이지 안내
  - 사용방법 A-Z까지 설명

## ① 사용가이드



[이미지62, 63, 64] 사용 가이드( 회원 정보 관리, 활동 내역 기록, 활동 통계)

## 사용가이드 설명서(2)

- 마이페이지 - 회원 정보 관리
  - 사용자 이름, 이메일, 비밀번호 변경, 연락처 관리 기능 제공
  - 사용자는 자신의 기본 정보를 수정·저장 가능
- 마이페이지 - 활동 내역 기록
  - 과거 탐색 경로 기록 확인 가능
  - 날짜, 출발/도착지, 탐색 거리 등 상세 내역 조회
  - 지도에 당시 탐색 경로를 다시 시각화 가능
- 마이페이지 - 활동 통계
  - 총 이동 거리 (예: 526.75 km)
  - 총 탐색 횟수 (예: 105회)
  - 차트/그래프 제공

끝까지 읽어 주셔서 감사합니다.