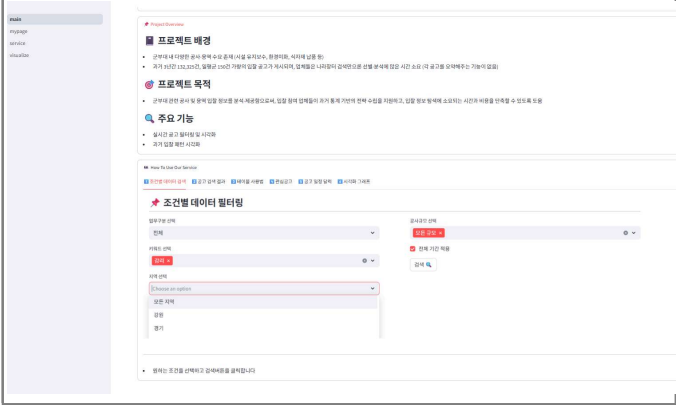
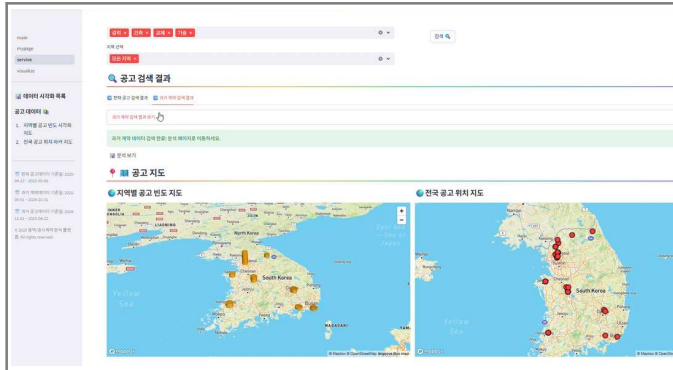
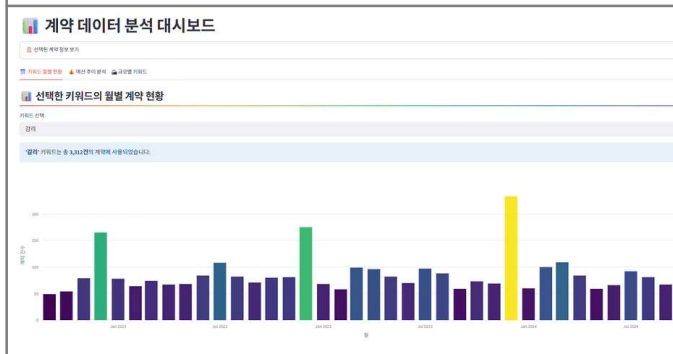


프로젝트명	국방 분야 공사·용역 공고 입찰 전략을 위한 통계 분석 및 정보 제공 서비스	수행기간	4/30 ~ 5/15
팀원	주재현, 김형래, 안보경, 김나영	GitHub url	x
역할 및 참여범위	-역할 : 웹 크롤링 및 대시보드 구현, 분석, DB설계		
프로젝트 개요	배경 : 군부대 내의 다양한 공사 용역에 대한 입찰의 어려움 타겟 : 군부대 시설 공사 및 용역 입찰을 원하는 업체 목적 : 군부대 관련 공사 및 용역 입찰 정보를 분석·제공 입찰 참여 업체들이 과거 통계 기반의 전략 수립 지원 입찰 정보 탐색에 소요되는 시간과 비용을 단축		
개발 PLATFORM	VScode / Streamlit / MySQL		
각별히 애했던 부분	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 여러가지 문제들을 해결하느라 노력을 많이 함 대시보드에서 알바몬 / 알바천국 같이 캘린더 및 관심공고등을 표현하여 사용자 친화적인 UI를 제공함		
		Main 페이지로 우리가 제공하는 서비스를 사용하는 방법을 알려주는 페이지입니다.	



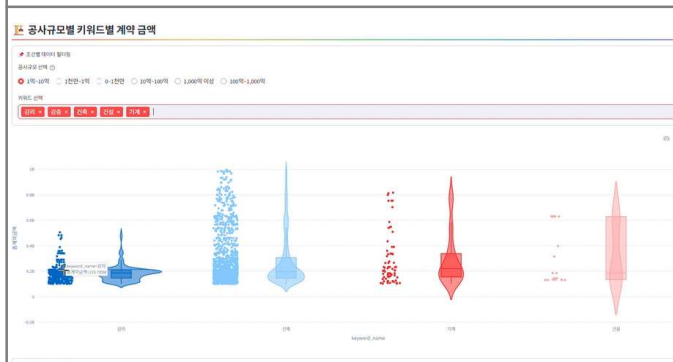
사용자가 관심있는 공고를 지역 / 키워드 별로 검색할 수 있습니다.

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



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

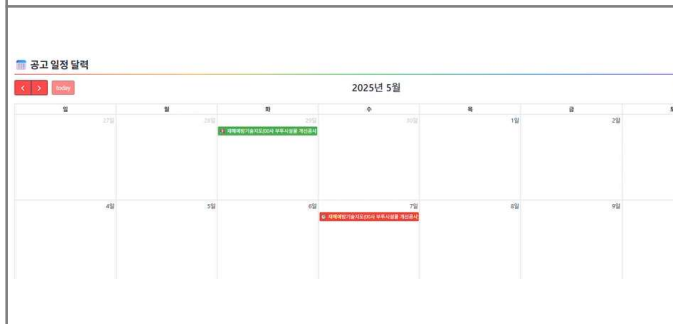
키워드별 / 예산 추이 분석 / 규모별 분석 그래프를 제공합니다.



공사 규모별 분석 그래프 예시



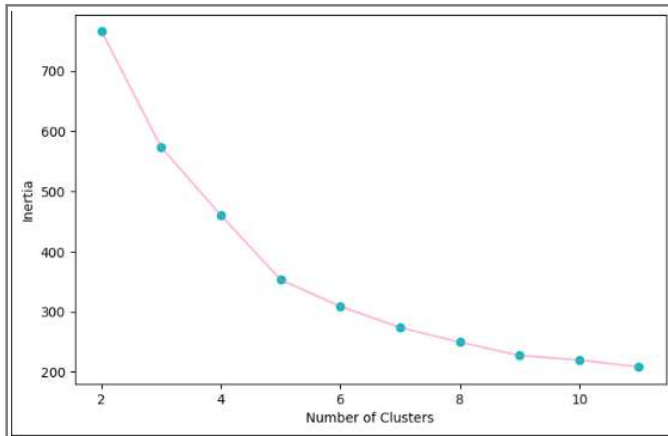
마이페이지로 내가 관심가는 목록을 추가하여 필요없는 정보를 제외하고 온전히 필요한 공고들만을 볼 수 있는 기능입니다.



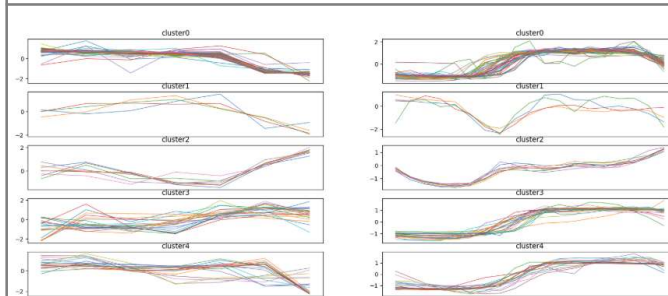
관심 공고의 시작 마감 일자를 달력에 추가하여 가시성을 높여주고 사용자가 일정을 잊지 않도록 도와주는 기능을 합니다.

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

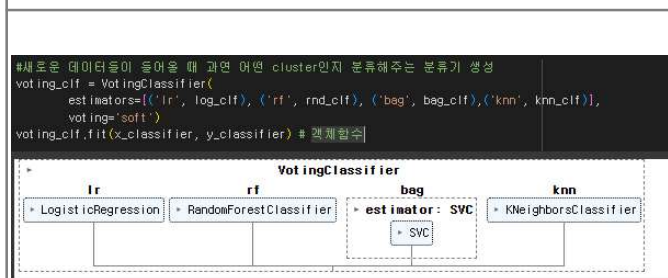
<p>각별히 애썼던 부분</p>	<p>1. 전체적인 방향성</p> <p>* 건물별군집화를 통해 클러스터를 나눈 후 나뉜 클러스터별로 다른 피처를 추가해 모델을 생성하는 방식 최종적으로 우리는 k_means clustering을 사용하였고 elbow method를 통해 군집의 개수 결정하였습니다. 결정된 군집 수 만큼 모델 생성(각 클러스터마다 건물별로 데이터들이 들어있을것인데 각 건물별로 모델링 한 후 앙상블을 통해 클러스터별 최종 한개의 모델을 생성하였습니다)</p> <p>또한 요일별 전력 중앙값, 시간대별 전력 중앙값으로 분류기를 만들어 건물별로 어느 군집에 속하는지 분류를 실시하였습니다.</p> <p>이후 각 분류에따라 테스트 데이터에 피처를 군집에 특성에 맞는 피처를 각공해주고 군집에 맞는 최종 모델로 예측을 실시하였습니다.</p> <p>2. 이상치 및 결측치 처리</p> <p>* 결측치는 아래의 코드에서 설명되어있듯이 풍속과 습도는 선형보간을 통해 처리해주었습니다. 또한 풍속과 습도에 관련된 불쾌지수, 체감온도 피처를 공식에따라 작성하여 추가하였고, 냉방면적에 대한 전력사용량 관계성을위해 냉방도시피처를 추가하였습니다.</p> <p>* 이상치는 eda를 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 건물에대해서 다른 모든 월요일 , 수요일의 평균을 넣어줌으로써 공휴일이 없던것처럼 만들었습니다.</p> <p>* 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.</p> <p>* 주말에 임시적으로 쉬는 건물들에 대해서 주기성을 만들어 주는 작업을 진행하였습니다. 예를들어 7월10일에쉬고 7월 24일에 쉬는 격주마다 쉬는 건물이었는데 7월31일도 쉬었다고 한다면 이 건물에 대해서 7월 31일 데이터는 안 쉰 데이터로 치환하는 작업을 하였습니다. 이로서 모델은 격주로 쉬는 날임을 예측할 것입니다.</p> <p>3. 후처리</p> <p>* 클러스터를 나눈 후 모델을 학습하기위해서 클러스터별 리스트형태로 데이터프레임들을 넣고, 각 데이터프레임에 클러스터에 특징에 맞게 피처를 추가하는 후처리를 진행하였습니다.</p> <p>* 또한 테스트 데이터들도 후처리 작업이 필요하므로 분류된 클러스터에 맞게 데이터 프레임을 후처리하는 작업을 진행하였습니다.</p>
---------------------------	---



최종적으로 저희는 k_means clustering을 사용하였고 elbow method를 통해 군집의 개수 결정하였습니다.



최종적으로 클러스터링 된 전기 사용량 별 군집



새로운 데이터가 들어올 때 어떤 cluster인지 분류해주는 분류기를 voting 방식의 ensemble을 사용하여 구분해주었습니다.

```

# 새로운 데이터들이 들어올 때 어떤 cluster인지 분류해주는 분류기 생성
voting_clf = VotingClassifier(
    estimators=[('lr', log_clf), ('rf', rnd_clf), ('bag', bag_clf), ('knn', knn_clf)],
    voting='soft')
voting_clf.fit(x_classifier, y_classifier) # 모델학습

# 최종 모델의 예측
# y_pred를 분류한 것들에 따라서 새로운 x_test데이터 분리
y_pred_clust0 = np.where(y_pred == 0)[0]
y_pred_clust1 = np.where(y_pred == 1)[0]
y_pred_clust2 = np.where(y_pred == 2)[0]
y_pred_clust3 = np.where(y_pred == 3)[0]
y_pred_clust4 = np.where(y_pred == 4)[0]

x_test_spl10 = x_test[x_test['building_number'].isin([x + 1 for x in y_pred_clust0])]
x_test_spl11 = x_test[x_test['building_number'].isin([x + 1 for x in y_pred_clust1])]
x_test_spl12 = x_test[x_test['building_number'].isin([x + 1 for x in y_pred_clust2])]
x_test_spl13 = x_test[x_test['building_number'].isin([x + 1 for x in y_pred_clust3])]
x_test_spl14 = x_test[x_test['building_number'].isin([x + 1 for x in y_pred_clust4])]

# Htest_spl10데이터에 대한 처리작업
x_test_spl10['peak'] = (x_test_spl10['hour'] >= 0 & (x_test_spl10['hour'] <= 10)).astype(int)
x_test_spl10['holiday'] = (x_test_spl10['day'] == 5) | (x_test_spl10['day'] == 6).astype(int)
x_test_spl10.drop(['building_number', 'date_time', 'building_type', 'date', 'dnh', 'hour', 'dow', 'day', 'month', 'week'], axis=1, inplace=True)

x_test_spl11['nonpeak'] = (x_test_spl11['hour'] >= 5 & (x_test_spl11['hour'] <= 0)).astype(int)
x_test_spl11.drop(['building_number', 'date_time', 'building_type', 'date', 'dnh', 'hour', 'dow', 'day', 'month', 'week'], axis=1, inplace=True)


x_test_spl12['nonpeak'] = (x_test_spl12['hour'] >= 1) & (x_test_spl12['hour'] <= 6).astype(int)
x_test_spl12.drop(['building_number', 'date_time', 'building_type', 'date', 'dnh', 'hour', 'dow', 'day', 'month', 'week'], axis=1, inplace=True)

x_test_spl13['peak'] = (x_test_spl13['hour'] >= 7 & (x_test_spl13['hour'] <= 20)).astype(int)
x_test_spl13['holiday'] = 0
x_test_spl13.drop(['building_number', 'date_time', 'building_type', 'date', 'dnh', 'hour', 'dow', 'day', 'month', 'week'], axis=1, inplace=True)

x_test_spl14['peak'] = (x_test_spl14['hour'] >= 0 & (x_test_spl14['hour'] <= 20)).astype(int)
x_test_spl14['holiday'] = (x_test_spl14['dow'] == 5).astype(int)
x_test_spl14.drop(['building_number', 'date_time', 'building_type', 'date', 'dnh', 'hour', 'dow', 'day', 'month', 'week'], axis=1, inplace=True)

```

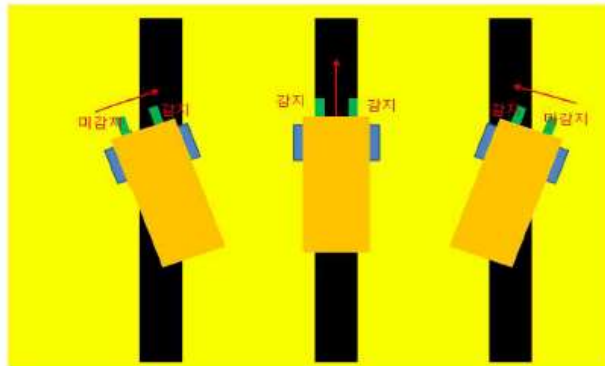
최종적으로 5개의 군집별 모델들을 만들어 분류된 class에 맞게 모델을 적용시키는 방식으로 마무리하였습니다.

프로젝트명	스마트 농업을 위한 자율주행 레이저 제초 로봇	수행기간	2024/03/16 ~ 2024/06/16
팀원	주재현 안영록 한성모 이종연	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 현재세대와 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(LineTracing)과 YOLO_V8 알고리즘을 이용한 객체 인식을 통해 잡초를 탐지하고, 레이저를 사용하여 잡초를 제초하는 로봇을 제안합니다. 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 선정하는데 있어서도 많은 고민을 하고 또한 모터들의 정격, 다른 센서들의 정격을 생각하여 시행착오를 겪으면서 선정하였습니다.</p>		
<p>2.1.2 OpenCV, 적외선 센서를 이용한 Line Tracing</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 첫번째는 OpenCV를 활용해 실시간 이미지를 받아와 픽셀 값이 한 쪽으로 급격히 변하는 부분이나 사물과 사물, 사물과 배경의 경계선인 엣지를 검출하여 검출된 선 성분에서 평균직선을 그려 휠 스티어링 각도를 얻는 방식이다.</p>  <p>[그림 2-3] Edge 검출</p>		<p>Canny 알고리즘을 이용하여 Edge를 검출하고 Region On Interest방식을 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.



[그림 2-6] 적외선 센서를 이용한 Line Tracing

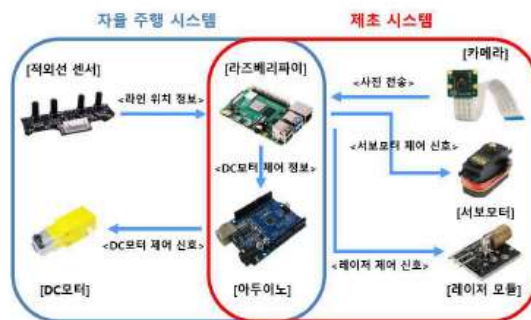
적외선 센서를 이용한 Line Tracing 방식을 채택하여 자율주행을 진행하였습니다.

레이저 구동 시스템: 객체 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.



<p>- 레이저 구동 시스템 동작 원리</p> <p>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.</p> <p>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.</p> <p>이 과정을 통해 레이저 구동 시스템은 잡초를 효과적으로 겨냥하고 제거할 수 있다.</p> <div data-bbox="235 640 457 835"> </div> <p>[그림 2-20] 서보모터 구동</p> <div data-bbox="587 636 810 835"> </div> <p>[그림 2-21] 레이저 구동</p>	<p>잡초를 인식하고 목표를 조준하는 알고리즘을 촬영한 장면입니다.</p>
<div data-bbox="186 905 854 1325"> </div>	<p>실제 시연 장면</p>

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	https://www.miricanvas.com/v/152ltya
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		

<p>프로젝트 개요</p>	<p>1. 배경</p> <p>전쟁의 향방을 바꾼 보급로 확보 중요</p> <ul style="list-style-type: none"> 2022년 러·우 전쟁에서 우크라이나는 드론과 위성 영상을 활용해 도로 파괴 현황과 적 위치를 파악하고, 보급망을 유지하여 방어에 성공함. <p>장애물/지형적 요소를 반영한 경로탐색 시스템 필요</p> <ul style="list-style-type: none"> 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) <p>2. 분석 지역 선정 - 은평구·서대문구·마포구</p> <p>공간·정책적 연계성</p> <ul style="list-style-type: none"> 서울시 생활권 계획상 동일한 서북권에 속하는 권역임. 공동 개발·재생 계획 및 교통망 설계가 추진되어 정책적 연속성과 비교 가능성이 높음. <p>교통축선의 전략성</p> <ul style="list-style-type: none"> 통일로·자유로·제2자유로를 통한 파주-서울 연결축 형성됨. 평시 물류·통근 중심축, 유사 시 병력·물자 수송 및 보급로로 기능함. <p>지휘·통제 거점과의 연계성</p> <ul style="list-style-type: none"> 용산 국가위기관리센터와 전방 지역을 연결하는 전략적 접근 통로임. 유사 시 병참선 확보와 최종 방어선 역할 수행 가능성이 높음. <p>3. 목적</p> <p>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.</p> <p>4. 주요 적용 기관</p> <ul style="list-style-type: none"> 국군수송사령부, 군수사령부, 육·해·공군 본부, 연합사령부
<p>개발 PLATFORM</p>	<p>VScode / Flask / jupyter notebook / mysql / render</p>
<p>각별히 애했던 부분</p>	<p>알고리즘을 구현하는데 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.</p>

u	v	highway	oneway	length	bridge	tunnel	avg_height	avg_slope	lanes	up_lanes	down_lanes	surface
287630237	2451940610	residential	0	55.09174701		yes	8.25	0.362278218	4	2	2	
287630237	2139941906	residential	1	62.1456816			8.58500004	0.141539025	2	1	1	
355129397	355129299	tertiary	1	47.04975344			8.65999889	0.141539025	4	2	2	paving_stones
307924211	2930525549	residential	0	82.42929389	yes		30.35000038	0.389096727	4	2	2	sand
307924211	2960308646	secondary	1	9.439797278			30.60000038	0.389096727	4	2	2	asphalt
3020786325	3820786328	residential	0	80.6103812	yes		30.35000038	0.389096727	2	1	1	dirt
3829310959	3829311033	secondary	1	132.5750315		yes	43.78499865	7.612212222	4	2	2	earth
307924222	3823586881	residential	0	49.75974644			27.80999947	0.0252765522	3	2	2	
307924223	307924224	residential	0	20.54590260		yes	28.97999954	0.0549937188	4	2	2	
387154518	438718632	tunnel	1	340.0158334			14.17500019	6.365841721	1	1	0	concrete

[표1] 도로 속성 데이터 수집 (총 27,034개)

- 출처 OSMnx 및 Overpass API
- 도로구간(edge)별 주요 속성 포함(총 27,134개행)

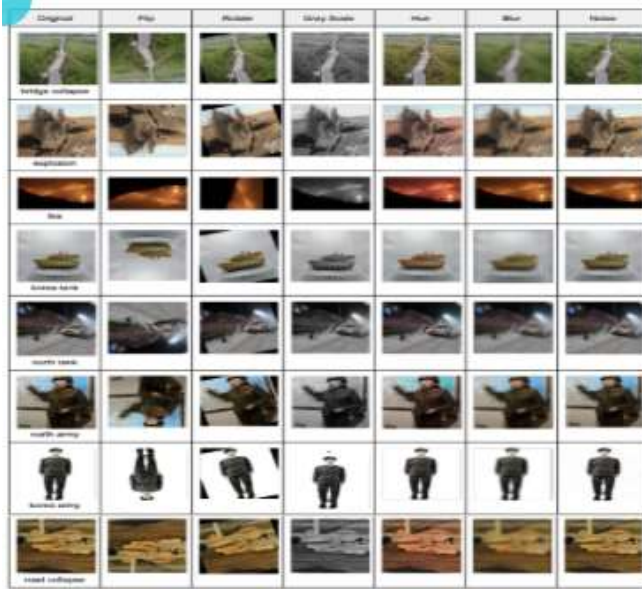
o u,v: 시작/끝노드
o highway: 도로유형
o length: 길이
o oneway: 일방통행여부
o bridge, tunnel: 교량/터널여부
o lanes, up/downlanes: 차로수
o avg_height: 평균 교도
o avg_slope: 평균 경사도
o surface: 포장재질

- 이후 전처리를 통해 결측치보완 및 파생변수 생성
- 원-핫인코딩 및 타겟인코딩을 수행하여 모델입력데이터로 활용

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	952864	1951944
2007010200100040	952099	1952567

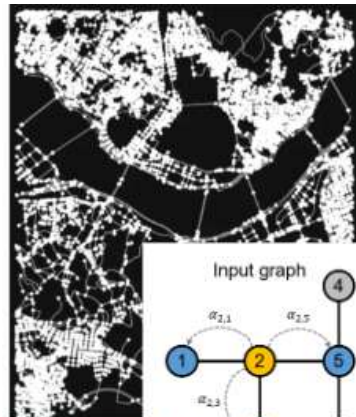
[표2] 교통사고 좌표 데이터 수집 (총 8,187개)

- 출처: 한국교통안전공단 교통사고분석시스템(TAAS)
- 분석지역: 서울특별시은평구·서대문구·마포구 도로망 수집방식:
 - TAAS는 웹기반 지도 서비스에서 사고 지점을 원형마커로 시각화
 - 이과정에서 브라우저에 WPS(Web Feature Service) 요청 결과(GeoJSON)가 공개적으로 전송
 - 해당응답에서 좌표(x_crdnt, y_crdnt) 및 사고 식별번호(acdnt_no) 등을 추출
- 좌표체계: EPSG:5179 (중부원점 TM, 한국 표준 좌표계)
- 수집데이터(총 8,187개행):
 - acdnt_no: 사고식별번호
 - x_crdnt, y_crdnt: 사고발생좌표

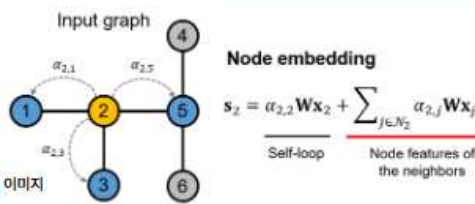


[이미지20] 방해 요소 이미지 데이터 증강 예시

- 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 라벨을 추가



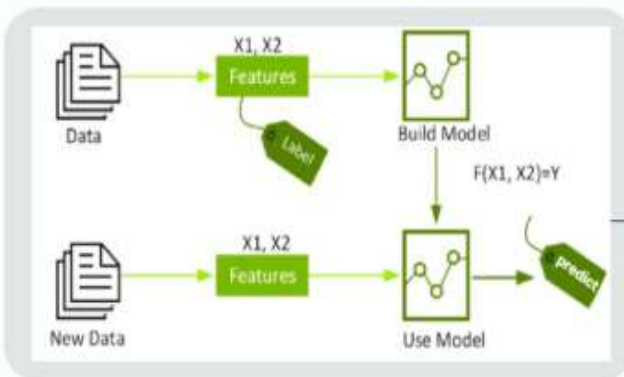
[이미지21] OSMNX 예시 이미지



[이미지22] GAT 학습식

GAT(graph attention networks)모델 학습

- 분석 지역: 은평구·서대문구·마포구 도로망 활용
- 분석 알고리즘: GAT
- 피쳐(X): 도로 속성 및 지형 정보 17가지 (차선 수, 길이, 경사도, 고도, 표면 상태 등)
- 타겟(y): XGBoost로 산출한 '옛지별 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



[이미지23] XGBoost 학습 과정

XGBoost 모델 학습

[실험 설정]

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

[배경 선정]

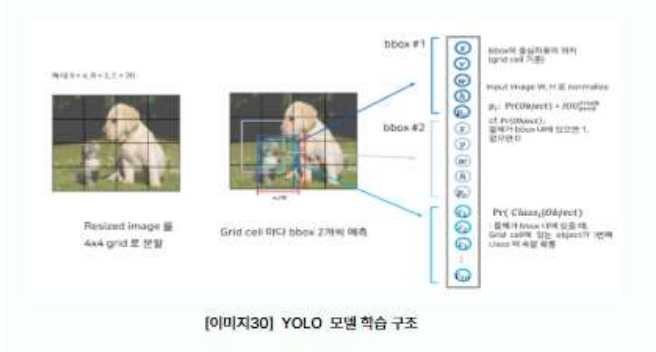
- 실제 전시 상황의 도로 위험도 데이터 부재
- 교통사고 데이터 활용하여 도로 구조적 요인 기반

사고율을 전시 상황 위험도로 추정

- GAT 모델 학습의 레이블(y 값)로 사용

방해 유형별	가중치	간단설명	출처	반경(m)	간단설명	출처
교량 붕괴 (Bridge collapse)	100,000,000	붕괴 시 차량 동행 불가 → 경로에서 안전의 배제	IMBC 보도, CMU 논문	10	차량 동행 불가 → 경로에서 안전의 배제	없음
도로 붕괴 (Road collapse)	100,000,000	도로 기능 상실 → 경합/소행이 즉시 발생	IMBC 보도	10	차량 동행 불가 → 경로에서 안전의 배제	없음
탱크 (Tank)	10,000	주박전차(M1A1) 2-3km 교전 가능, 최후탄 2.5km 시거리 → 치명적 위협	M1 Abrams 위키, FGM-148 Javelin 위키	300	북한군 전차 시거리 참고	나무위키, 조선인민군 복고[참고]
적군 (enemies)	5,000	상업 소총(M16, 480m, AK-47) 및 RPG-7 (300m) 위협 → 공간 위협, 인원수 배 배제	M16 위키, RPG-7 위키	200	북한군 교전 시거리 참고	나무위키, 조선인민군 복고[참고]
화재 (fire)	3,000	이러 저하 및 연 피해, 도로 통행 가능 → 탱크 폭발보다는 낮음	Wildfire Preparedness	200	중행불 기준의 최소값	참고할만한 참고사항(한글) 공개 (연보 2024)
폭발/IED (explosion)	10,000	도로 폭의 폭발물로 차량-구조물 파괴	DHS IED Fact Sheet	300	차량의 폭발 기준	논문 참고(CNSG, North Korean Short-Range Systems Analysis [2019])
철조망(barbed_wire)	100,000,000	경로에서 안전의 배제	없음	10	차량 동행 불가 → 경로에서 안전의 배제	없음
낙석(rockfall)	100,000,000	경로에서 안전의 배제	없음	10	차량 동행 불가 → 경로에서 안전의 배제	없음

전시상황 시 발생할 수 있는 방해요소들별 각 도로에 위험도를 반영하여 우회할 수 있도록 가중치를 설정해주었습니다.



- 입력(X): 8 개 클래스 (화재, 도로 붕괴, 다리 붕괴, 폭발, 한국군, 북한군, 한국 전차, 북한 전차)
- 타겟(y): 각 이미지 내 방해 요소 위치 (바운딩 박스 좌표) + 클래스 라벨
- 학습: 훈련 9: 검증 0.5 : 테스트 0.5 세트 분리, 하이퍼파라미터 조정하며 성능 최적화
- 출력:
 - 방해 요소 위치 (Bounding Box)
 - 종류 (Class Label)
 - 개수 (Object Count)
 - 신뢰도 점수 (예: 화재 92%)
- 활용:
 - 드론 영상에서 실시간 객체 탐지 수행
 - 위치·종류·개수에 따라 방해 요소 점수(가중치)를 계산 -> 최종 비용 계산식에 반영

YOLO탐지 기반 방해요소 점수 (RiskScore) 계산식

$$\begin{aligned}
 \text{RiskScore}(e) = & d(\text{road_collapse}) \times 100,000,000 \\
 & + d(\text{bridge_collapse}) \times 100,000,000 \\
 & + d(\text{tanks}) \times n1 \times 10,000 \\
 & + d(\text{enemies}) \times n2 \times 5,000 \\
 & + d(\text{fire}) \times n3 \times 2,000 \\
 & + d(\text{explosion}) \times n4 \times 10,000 \\
 & + d(\text{barbed_wire}) \times 100,000,000 \\
 & + d(\text{rockfall}) \times 100,000,000
 \end{aligned}$$

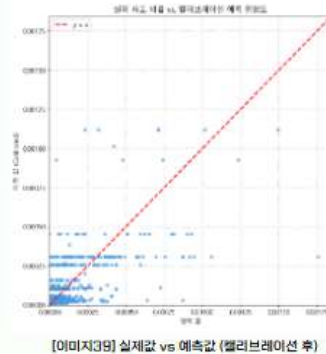
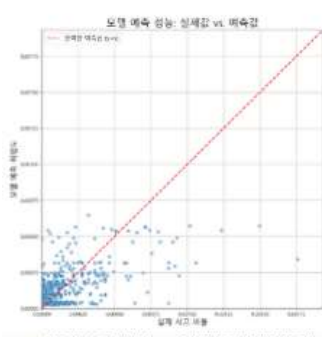
[이미지31] Riskscore 계산식 이미지

- d(): 해당 위험 요소 탐지 여부
- n: YOLO에서 탐지된 객체 개수
- 여러 위험이 동시에 탐지되면, 가중치 합산 최종 RiskScore에 반영
- C_base(e): 도로 길이 비용
- RiskScore(e): YOLO 탐지 기반 방해요소 점수 (위치·종류·개수 반영)
- C_GAT(e): GAT 모델이 예측한 도로 기본 위험도

각 도로(엣지)의 최종 비용(C_total) 계산식

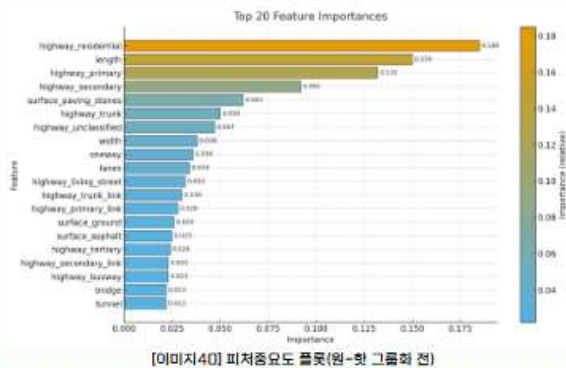
$$C_{total}(e) = C_{base}(e) + RiskScore(e) + C_{GAT}(e)$$

① 실제값-예측값 산점도 캘리브레이션 전 / 후

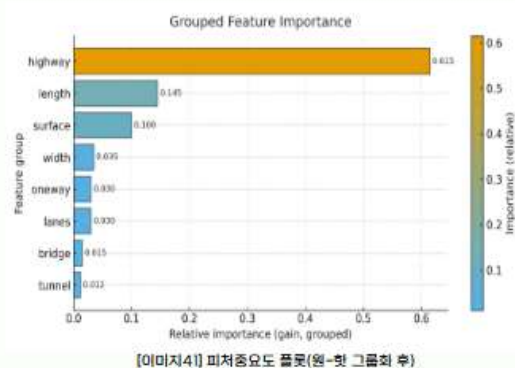


- XGBoost 산점도 개념
- 예측된 위험도와 실제 위험도를 비교
- XGBoost 전/후 비교
 - 캘리브레이션 전: 저위험 과대, 고위험 과소
 - 캘리브레이션 후: 고위험 구간 보정으로 기준선 근접
- 해석
 - 캘리브레이션으로 예측 균형성 전반적 향상
 - 데이터 희소성으로 인해 여전히 불안정한 구간 존재
- 의미 및 향후 과제
 - 고위험 구간 상향 보정 → 캘리브레이션 효과 있음
 - 고위험 데이터 보강 필요

② 피처 중요도(Feature Importance) 플롯

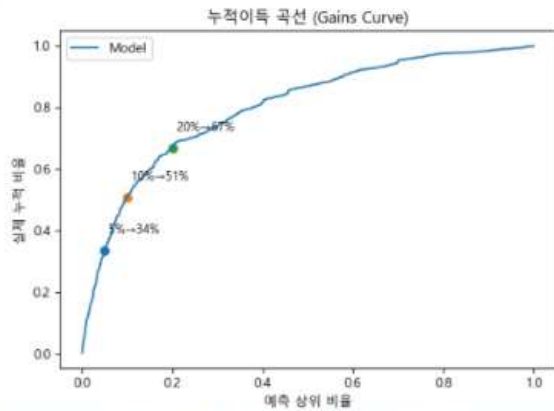


- 피처중요도 플롯 (원-핫 그룹화 전)
 - 모델이 실제값에 더 가까운 예측을 하도록 가장 많이 도움을 준 정보 순으로 상위 20개를 보여줌
 - 특히 도로 등급(highway), 길이(length), 차로·일방통행(lanes/oneaway)의 기여가 큼



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

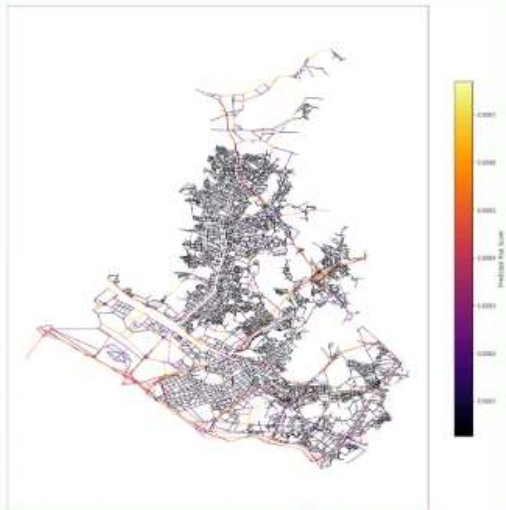
③ 누적 이득 곡선(Gains Curve)



[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

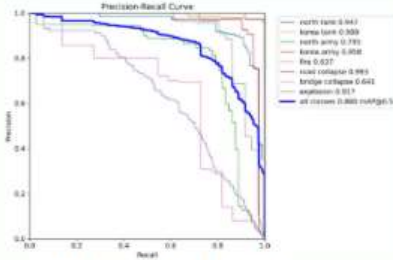
④ GAT 위험도 시각화



[이미지43] GAT 위험도 시각화

- 시각화
 - 보라색: 낮은 예측 위험도
 - 노란색: 높은 예측 위험도
- 주요 결과
 - 핵심 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

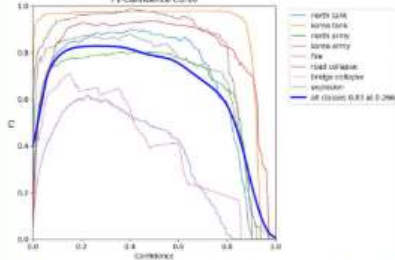
Precision-Recall Curve



[이미지44] 모델의 재현율과 정밀도를 비교한 Precision-Recall 곡선 예시

- 개념
 - X축: Recall(재현율), Y축: Precision(정밀도)
 - 곡선 아래 면적(AP) 클수록 좋음
- 그래프 해석
 - 곡선이 오른쪽 위에 가까울수록 정밀도·재현율 동시 유지
 - 곡선이 아래로 가지면 임계값 설정에 따라 성능이 불안정
 - 전체 평균 mAP@0.5 = 0.860
 - road collapse, korea tank 곡선이 오른쪽 위 (AP ≈ 0.99)
 - fire, bridge collapse 곡선이 아래로 갈수록 (AP ≈ 0.63~0.64)

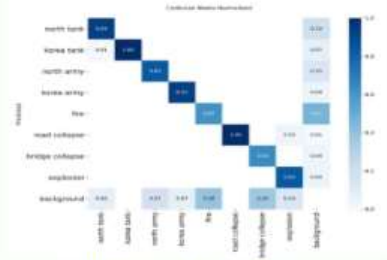
F1-Confidence Curve



[이미지45] 신뢰도 변화에 따른 F1 점수 변화를 보여주는 F1-Confidence 곡선 예시

- 개념
 - X축: Confidence(신뢰도 임계값), Y축: F1 점수(Precision·Recall 조화 평균)
 - Confidence에 따라 Precision, Recall 바뀌고, F1 값도 달라짐
- 해석 포인트
 - 곡선 높게 유지될수록 해당 클래스는 안정적인 탐지 성능을 가짐
 - 오른쪽으로 갈수록 Confidence가 높아도 안정적
 - 평균선(all classes)이 가장 높은 지점 = 최적의 임계값
 - 전체 평균 F1 = 0.83 (confidence=0.266)
 - korea tank, road collapse 높은 F1 유지 (성능 우수)
 - fire, bridge collapse 낮은 F1, 개선 필요

Confusion Matrix Normalized



[이미지46] 실제 값과 예측 값의 분포를 나타낸 Confusion Matrix (정규화) 예시

- 개념
 - 가로축: 실제 정답 / 세로축: 모델 예측값
 - 대각선이 높을수록 정확한 분류
 - 오프 대각선 값은 오탐(잘못 분류) 의미
- 해석
 - 진한 파란(대각선) = 잘 맞춘 클래스
 - 밝은 색(비대각선) = 혼동 발생
 - korea tank, road collapse 완벽하게 탐지 (1.00)
 - fire 38%를 background로 착각
 - bridge collapse 35% background로 오탐

하이퍼파라미터 튜닝

파라미터	튜닝 version 1	튜닝 version 2
epochs	50	50
imgsz	640	640
batch	8, 16, 32	32
optimizer	SGD, AdamW	SGD
lr0	0.0001, 0.01 사이	0.0025, 0.0055 사이
lrf	0.01	0.01
patience	20	20
close_mosaic	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

튜닝 결과: 최종 모델 성능

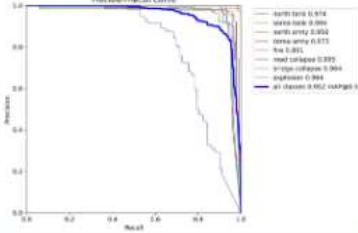
	batch	epochs	lr0	momentum	optimizer	weight_decay	mAP50	mAP50-95
1	32	50	0.004501	0.85105	SGD	0.0006240	0.89	0.653
2	32	50	0.003052	0.92099	SGD	0.0007431	0.891	0.649
3	8	50	0.000642	0.9158	AdamW	0.0002511	0.871	0.605
4	8	50	0.004299	0.9758	SGD	0.000962	0.883	0.636
5	16	50	0.0031795	0.90448	SGD	0.0008985	0.899	0.652
6	32	50	0.0040217	0.89518	SGD	0.0006048	0.911	0.732
7	32	50	0.0061958	0.75363	SGD	0.0007256	0.901	0.723
8	32	50	0.005175	0.81873	SGD	0.0005175	0.932	0.755
9	32	50	0.0027809	0.85082	SGD	0.0005320	0.932	0.755
10	32	50	0.0053359	0.91621	SGD	0.0005426	0.943	0.772

[표11] 하이퍼파라미터 튜닝 결과 비교

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

- 총 10개 모델 실험 ⇒ 모델 8 최종 선택
 - mAP50 = 0.952
 - mAP50-95 = 0.798

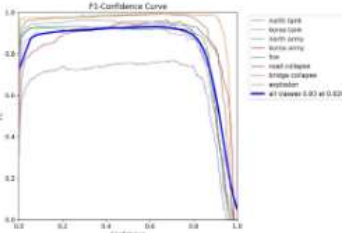
Precision-Recall Curve



[이미지47] Precision-Recall 곡선 이미지

- First Model: 최고 F1 = 0.83 (at conf=0.266)
- 최종 Model: 최고 F1 = 0.93 (at conf=0.626)
- 개선점:
 - 전체 평균 F1이 +0.10 상승
 - 최적 임계값이 더 높은 쪽(conf=0.626)으로 이동 모델의 확신이 높아져도 안정적 성능 유지

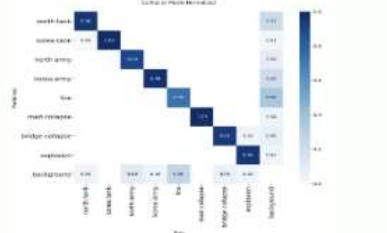
F1-Confidence Curve



[이미지48] F1-Confidence Curve 그래프 이미지

- First Model: 평균 mAP@0.5 = 0.860
- 최종 Model: 평균 mAP@0.5 = 0.952
- 개선점:
 - 전반적으로 모든 클래스에서 성능 향상
 - 특히 north army (0.795→0.950), bridge collapse (0.641→0.964), fire (0.637→0.801) 개선 확인

Confusion Matrix Normalized



[이미지49] Confusion Matrix (정규화) 그래프 이미지

- First Model:
 - fire: 0.62 background로 0.38 혼동
 - bridge collapse: 0.64 background로 0.36 혼동
- 최종 Model:
 - fire: 0.74 여전히 낮지만 +0.12 개선
 - bridge collapse: 0.95 혼동 크게 줄어듦
 - 탱크·도로붕괴·폭발: 0.96~1.00 거의 완벽한 탐지

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/경신 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

[표 12] 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

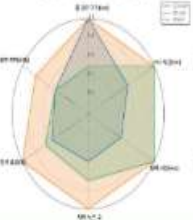
방해요소 0개

방해 요소 없이 도로 특성 없는 경우



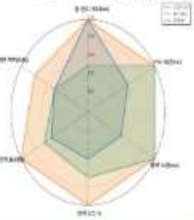
방해요소 1개

방해 요소 1개 도로 특성 없는 경우



방해요소 4개

방해 요소 4개 도로 특성 없는 경우

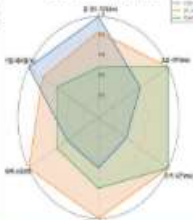


- 특성
 - CCH+A*: 0.5 수준으로 보통, 총 경로 거리 1.0로 최단 거리 탐색에 가장 강함
 - D Lite*: 회피율도 0.8로 우수, 이 외 모두 1.0으로 전반적으로 최적 성능
 - RTAA*: 탐색 시간 1.0으로 속도
- 위험 회피율 0.6, 경로거리 0.6, 탐색 효율 0.7은 상대적으로 낮음
- 종합 평가
 - 안전(회피율): D Lite* > RTAA* > CCH+A*
 - 종합 성능: D Lite*

도로 특성 있음

방해요소 0개

방해 요소 없이 도로 특성 있는 경우



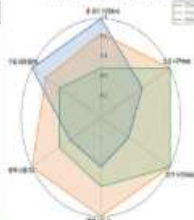
방해요소 1개

방해 요소 1개 도로 특성 있는 경우



방해요소 4개

방해 요소 4개 도로 특성 있는 경우



- 특성
 - CCH+A*: 위험 회피율 약 1.0수준으로, 총 경로 거리 1.0으로 높은 수치이지만, 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*

version 1.0.1 2023.12.15

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 < 0.05이므로, 귀무가설(H_0 : 알고리즘 간 평균 차이가 없다)을 기각
 - 모든 지표에서 알고리즘 간 평균 차이가 통계적으로 유의미함을 확인
 - 소요 시간(ms)**은 F=206.16, p(0.001로 가장 큰 차이를 보여, 알고리즘별 속도 성능 차이가 확연히 존재함을 의미

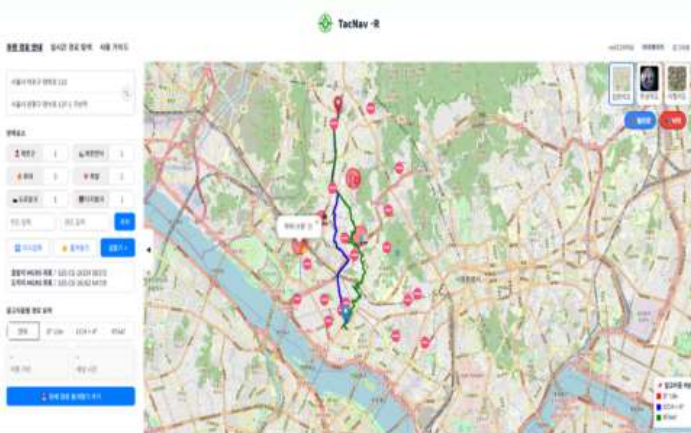


[이미지56, 57] 시스템 첫 화면

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

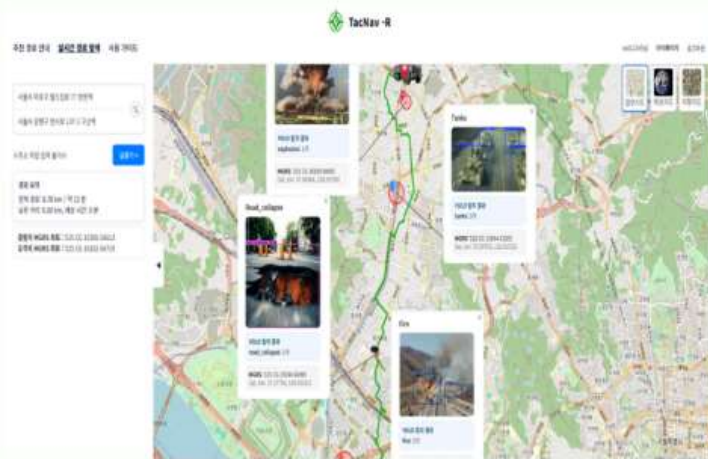


[이미지58] 사용자 맞춤형 경로 탐색 및 추천

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



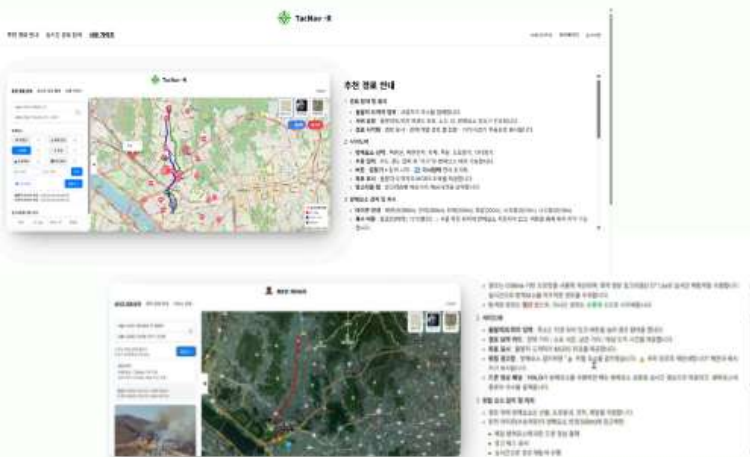
[이미지59] 실시간 장애물 탐색 및 경로 탐색

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

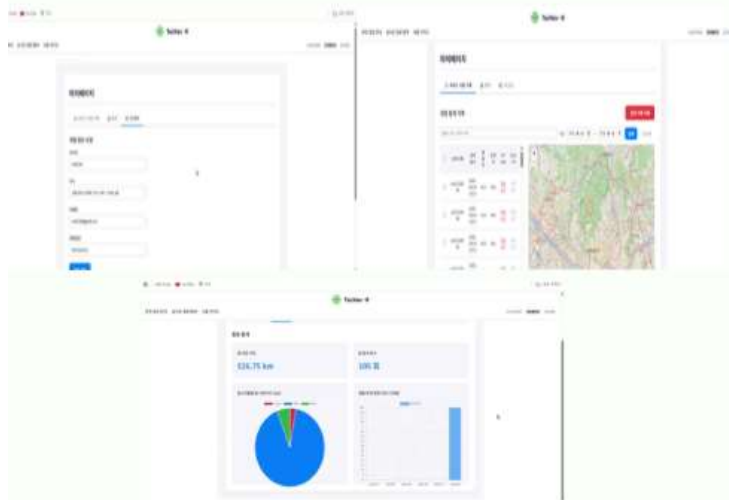
① 사용가이드



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.