

CNNs를 통한  
이륜차 사고다발지 예측

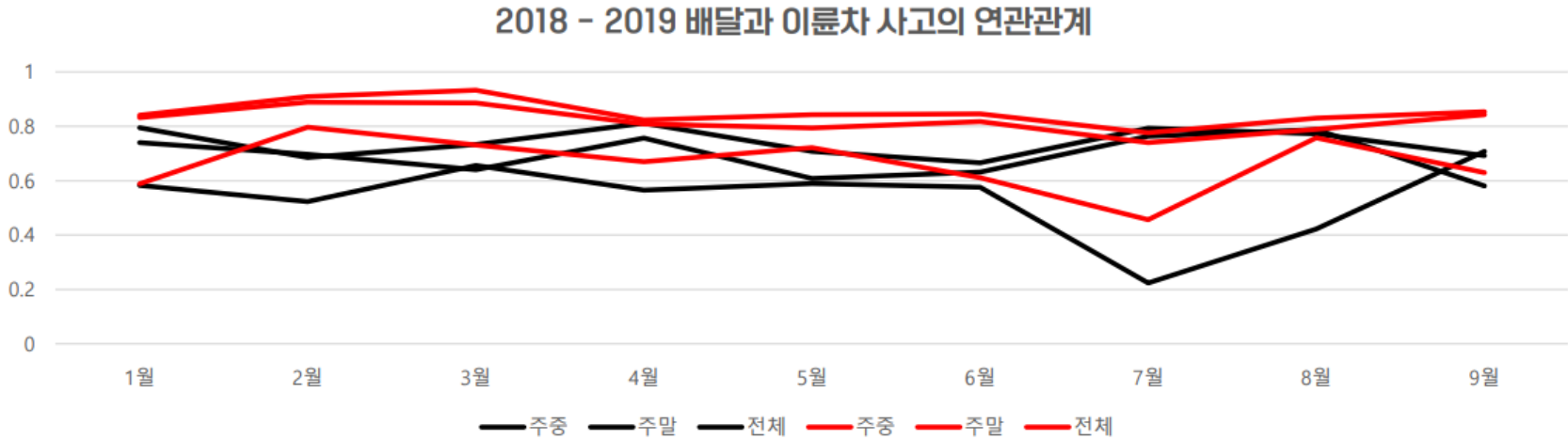
늘어난 배달과 이륜차 사고

구분		주 중	주 말	전 체
2018년 (서 울)	1월	0.740	0.583	0.795
	2월	0.697	0.523	0.685
	3월	0.640	0.656	0.733
	4월	0.757	0.565	0.811
	5월	0.608	0.590	0.708
	6월	0.632	0.576	0.666
	7월	0.764	0.224	0.793
	8월	0.788	0.422	0.772
	9월	0.581	0.708	0.692
전체 (2018년 1~9월)		0.653	0.536	0.699

〈2018년 배달과 이륜차 사고의 연관관계〉

구분		주 중	주 말	전 체
2019년 (서 울)	1월	0.832	0.589	0.841
	2월	0.889	0.797	0.909
	3월	0.886	0.731	0.933
	4월	0.809	0.670	0.823
	5월	0.794	0.722	0.843
	6월	0.817	0.611	0.846
	7월	0.740	0.457	0.776
	8월	0.789	0.758	0.830
	9월	0.843	0.630	0.854
전체 (2019년 1~9월)		0.764	0.610	0.783

〈2019년 배달과 이륜차 사고의 연관관계〉



## 사회적 이슈가 심각하다

시사 > 전체기사

### “오토바이 지상 출입 거부한 아파트, 배달 안 갑니다”

라이더 유니온 인천송도지회, 인천 송도 한 아파트단지 배달 중단키로  
앞서 해당 아파트단지, 배달 지상 출입 막으려 '세대호출' 제한 등

입력 : 2021-09-26 17:45 / 수정 : 2021-09-26 17:58

### 소음에 난폭 운전까지...배달 오토바이, 이대로 괜찮나

| 전문가 "제도 변화"·"이륜차 규제"·"전면 번호판 부착" 등 제언

인터넷 | 입력 : 2021/09/26 08:30 수정: 2021/09/27 08:40



김성현 기자 | ✉

기자 페이지 구독

기자의 다른기사 보기



[이벤트] 삼성SDS Brity RPA 개발자 컨퍼런스 참여 시 갤럭시 Z폴립, 갤럭시 버즈 등 증정!

### 오토바이의 위험성이 증가함에 따른 기사

안전 시설물이 과연 잘 작동하고 있는가?  
오토바이에 대한 사회적 이슈가 심각

# 연구 목적

- 사고는 여러가지 요인에 의해 발생
- 운전자의 부주의가 일어날 수 있는 환경을 제공하는 도로 형태를 분석 및 위험도 예측
- 특히 공간정보가 담긴 이미지를 활용

=> 더 안전한 도로를 만들 수 있을 것

# 이륜차 사고다발지 예측 관련 연구1 차이점

- 회귀 모델 대신 CNN 모델사용



이륜차사고 영향요인 종합분석을 통한 위험지역 도출 - 서울시 강남구를 중심으로

Identification of hazardous areas through comprehensive analysis of factors influencing two-wheeled veh

디지털문화아카이브지

약어 : JDCA

2021, vol.4, no.2, pp. 136-152 (17 pages)

DOI : 10.23089/jdca.2021.4.2.010

발행기관 : 전북대학교 문화융복합아카이빙연구소

연구분야 : 공학 > 교통공학

이한빈 /Hanbin Lee<sup>1</sup>, 김휘연 /HwiEon Kim<sup>2</sup>, 김세영 /Seyoung Kim<sup>3</sup>, 정승환 /Seung Hwan Jeong<sup>4</sup>, 소현기 /Hyeon-Gi So<sup>5</sup>,  
김태영 /Taeyoung Kim<sup>6</sup>, 박광현 /Gwanghun park<sup>7</sup>

<sup>1</sup>공주대학교 도시·교통공학

<sup>2</sup>전북대학교 문화융복합아카이빙연구소

~

# 이륜차 사고다발지 예측 관련 연구2 차이점

- 동일하게 CNN 모델사용
- 하지만, 하나의 이미지
- 하나의 모델



그림1.4 도로가 나무에 의해 가려진 위성 이미지

나) Combining Satellite Imagery and Open Data to Map Road Safety(2017)<sup>4</sup>

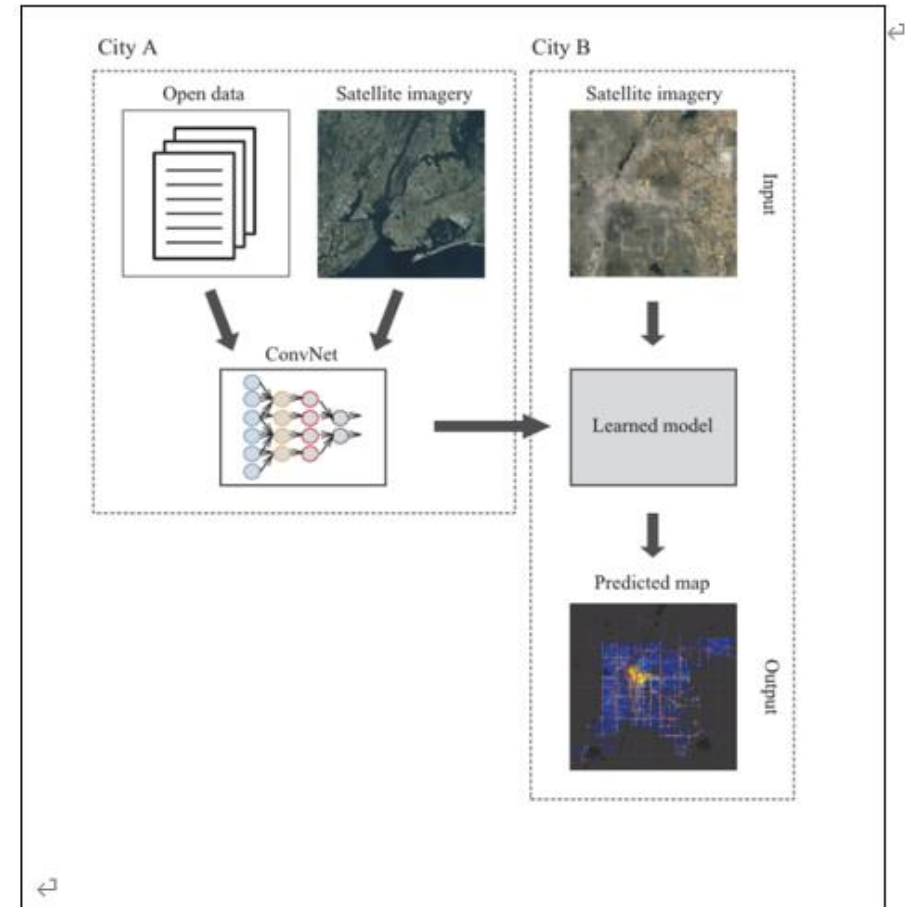
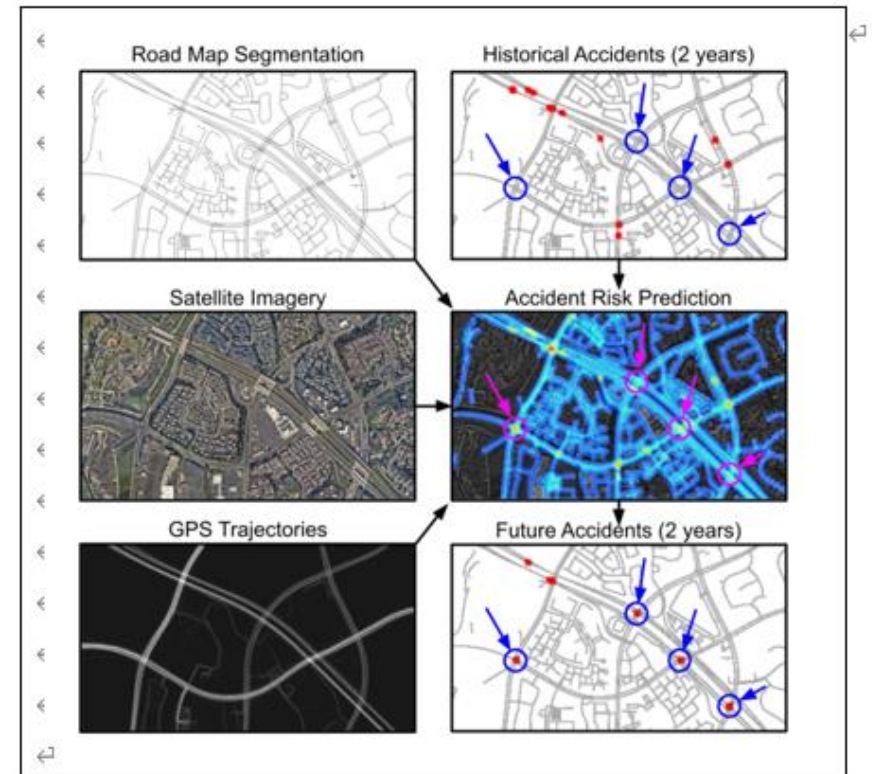


그림1.1 관련연구 나)의 시스템

# 이륜차 사고다발지 예측 관련 연구3 차이점

- 동일하게 CNN 알고리즘 사용
- 하지만, 본 프로젝트는 분류문제
- 관련연구는 회귀 문제
- 국내에서 얻기 힘든 데이터 사용

라) Inferring high-resolution traffic accident risk maps based on satellite imagery and GPS trajectories (2021)<sup>6</sup>



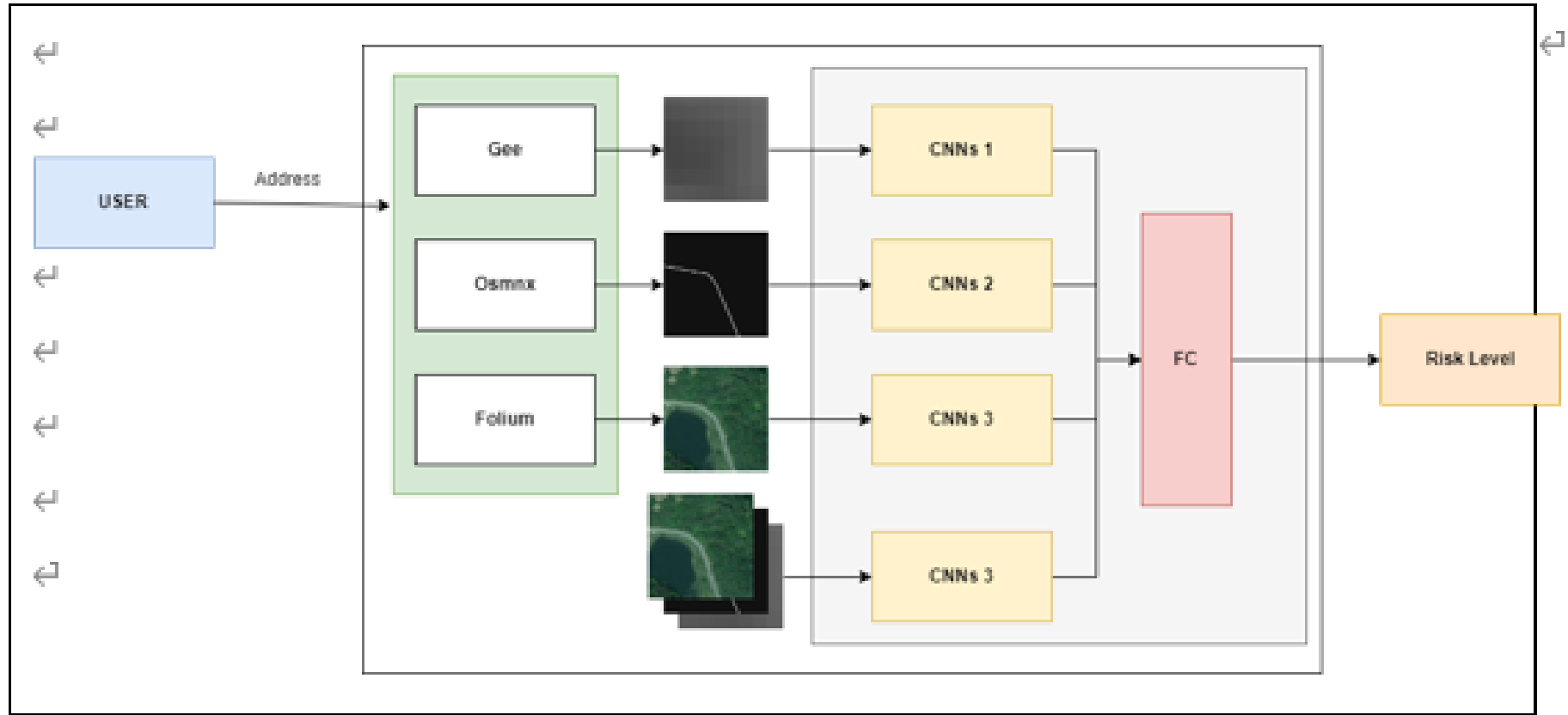
# 구현 환경

환경↵	이름↵	↵
운영체제↵	Windows 11 (버전: 22H2)↵	↵
개발언어↵	Python 3.8.11↵	↵
인공지능 프레임워크↵	<u>Tensorflow</u> 2.9.1 ( <u>cuda</u> 11.2)↵	↵
GPU↵	GTX 1660 super 6.0g↵	↵

주 사용 라이브러리 및 API↵	설명↵	↵
<u>Osmnx</u> 1.2.2↵	도로 이미지 수집 (도로 네트워크)↵	↵
Folium 0.12.1↵	도로 이미지 수집 및 시각화(위성)↵	↵
Google Earth Engine API↵	도로 이미지 수집(높이)↵	↵
Pandas 1.4.3↵	CSV 파일 처리↵	↵
<u>Opencv-python</u> 4.5.3.56↵	이미지 읽기↵	↵
Matplotlib↵	데이터 시각화↵	↵
<u>geopandas</u> 0.11.0↵	도로 <u>geojson</u> 처리↵	↵
<u>Clean-fid</u> 0.1.32↵	상관성 평가를 위한 FID 계산↵	↵



# 최종 시스템 구성도



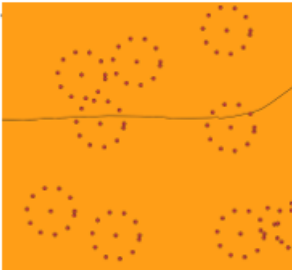
구현 내용

# 데이터 수집

## 가) CSV 데이터 수집

데이터 이름	제공 기관	URL
이륜차사고다발지 2016~2021	도로교통공단	<a href="https://www.data.go.kr/data/15105286/openapi.do">https://www.data.go.kr/data/15105286/openapi.do</a>
서울시 CCTV	Local Data	<a href="https://www.localdata.go.kr/lif/lifeCtacDataView.do?opnEtcSvcId=12_04_08_E">https://www.localdata.go.kr/lif/lifeCtacDataView.do?opnEtcSvcId=12_04_08_E</a>
서울시 과속방지 턱	Local Data	<a href="https://www.localdata.go.kr/lif/lifeCtacDataView.do?opnEtcSvcId=12_04_06_E">https://www.localdata.go.kr/lif/lifeCtacDataView.do?opnEtcSvcId=12_04_06_E</a>
서울시 횡단보도	서울 열린데이터 광장	<a href="http://data.seoul.go.kr/dataList/OA-15554/S/1/datasetView.do">http://data.seoul.go.kr/dataList/OA-15554/S/1/datasetView.do</a>
서울시 신호등	서울 열린데이터 광장	<a href="http://data.seoul.go.kr/dataList/OA-15554/S/1/datasetView.do;jsessionid=4B75D6E2529334747BAE40CC7FB013D5.new_portal-svr-11">http://data.seoul.go.kr/dataList/OA-15554/S/1/datasetView.do;jsessionid=4B75D6E2529334747BAE40CC7FB013D5.new_portal-svr-11</a>

# 데이터 수집: CSV



좌표 데이터 약 1,000개  
오버 샘플링 적용하여 10,000개로

## 가) CSV 데이터 수집

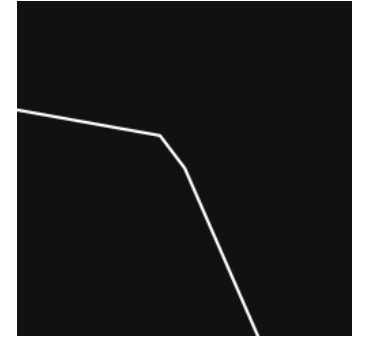
데이터 이름	제공 기관	URL
이륜차사고다발지 2016~2021	도로교통공단	<a href="https://www.data.go.kr/data/15105286/openapi.do">https://www.data.go.kr/data/15105286/openapi.do</a>
서울시 CCTV	Local Data	<a href="https://www.localdata.go.kr/lif/lifeCt acDataView.do?opnEtcSvcId=12_04_08_E">https://www.localdata.go.kr/lif/lifeCt acDataView.do?opnEtcSvcId=12_04_08_E</a>
서울시 과속방지 턱	Local Data	<a href="https://www.localdata.go.kr/lif/lifeCt acDataView.do?opnEtcSvcId=12_04_06_E">https://www.localdata.go.kr/lif/lifeCt acDataView.do?opnEtcSvcId=12_04_06_E</a>
서울시 횡단보도	서울 열린데이터 광장	<a href="http://data.seoul.go.kr/dataList/OA-15554/S/1/datasetView.do">http://data.seoul.go.kr/dataList/OA-15554/S/1/datasetView.do</a>
서울시 신호등	서울 열린데이터 광장	<a href="http://data.seoul.go.kr/dataList/OA-15554/S/1/datasetView.do?jsessionId=4B75D6E2529334747BAE40CC7FB013D5.new_portal-svr-11">http://data.seoul.go.kr/dataList/OA-15554/S/1/datasetView.do?jsessionId=4B75D6E2529334747BAE40CC7FB013D5.new_portal-svr-11</a>

# 데이터 수집: 이미지

OSMnx 1.2.2



도로 네트워크 이미지



위성 이미지



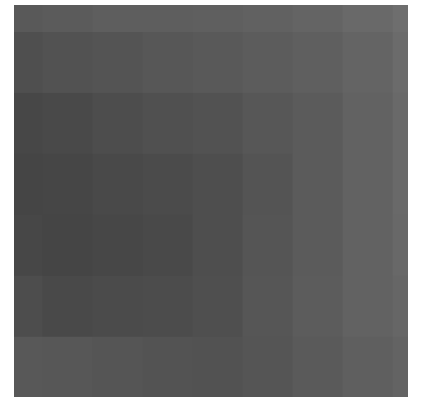
Folium



Google Earth Engine



높이 이미지



# 데이터 수집: CSV 데이터 처리

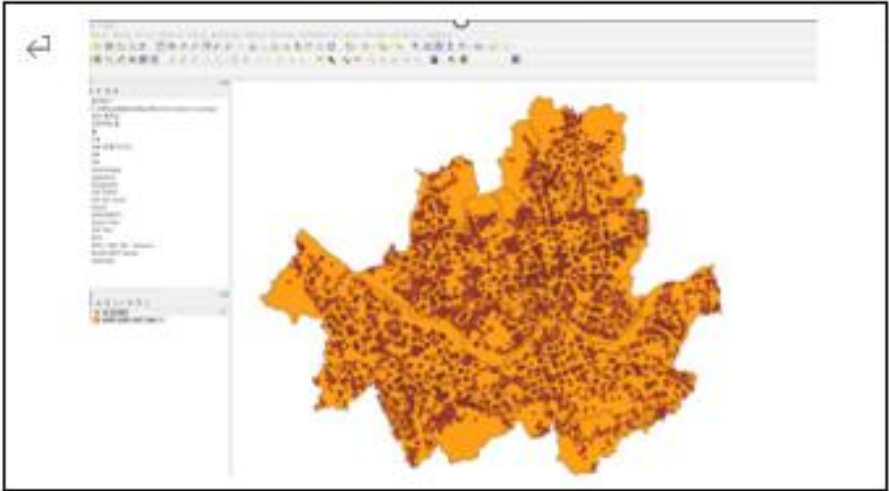


그림2.3 QGIS 화면, 사고다발지 데이터



그림2.4 QGIS 화면, 사고다발지 데이터, cctv 과속방지턱, 횡단보도, 신호등 데이터

# 데이터 수집: CSV 데이터 처리

←

	사망자수	중상자수	경상자수	부상신고자수	bumps	cctvs	walk	light
사고다발지FID								
104688	0	19	27	7	0	0	7	69
104683	0	6	16	11	0	2	8	35
104684	0	6	13	9	0	0	12	47
104573	0	9	12	3	2	1	7	62
104773	0	12	16	3	2	2	7	46

←

그림2.5 합친 CSV 결과←

# 데이터 탐색: CSV 데이터 상관분석

```
1 Import seaborn as sb
2 sb.heatmap(df2.corr(),
3           annot = True,
4           cmap = 'Greens', #색상
5           vmin = -1, vmax=1
6           )
```

코드 2.4 df2의 상관분석 결과를 히트맵으로 시각화

Matplotlib기반 패키지인 Seaborn을 사용, Pandas의 dataframe의 corr 멤버 함수를 사용하여 상관계수를 시각화 하였다.



그림2.6 상관분석 결과



# 데이터 탐색: FID 로 상관관계 판단

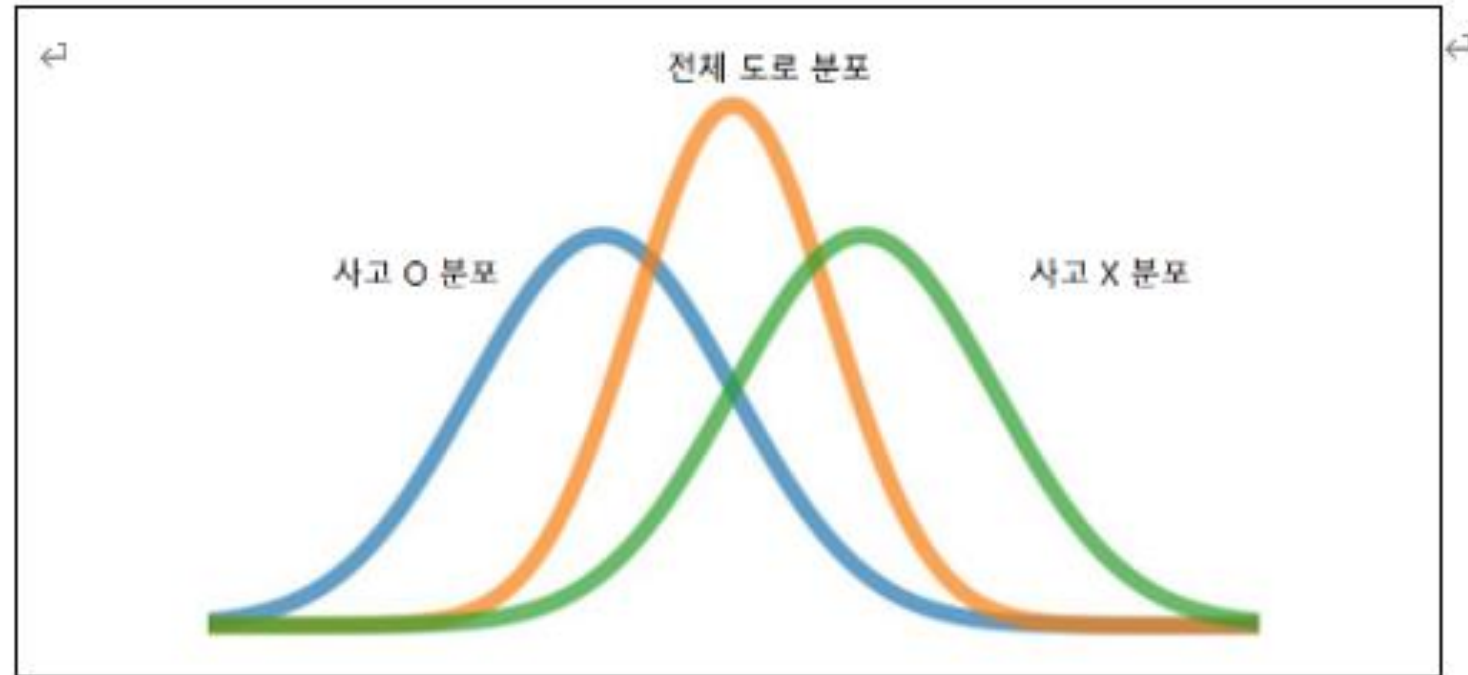


그림 2.9 FID

# 1차 모델 학습

## 나) 1차 모델 학습

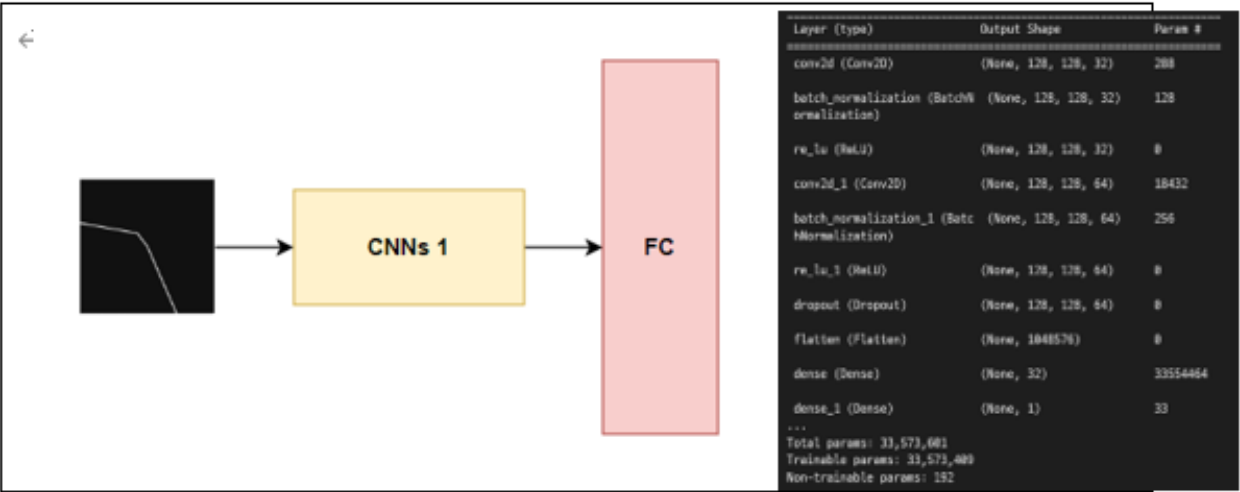


그림2.10 클래스 2개, 이미지 한 개가 인풋으로 들어가는 모델

## 다) 결과 (30 epoch)

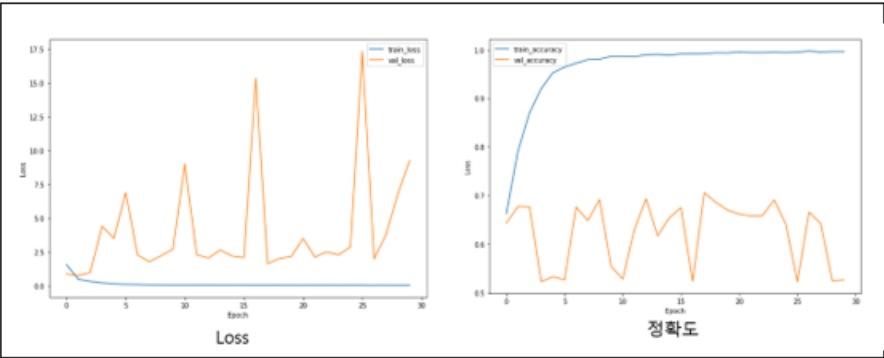


그림2.11 1차 모델학습 결과

	Train	Validation	Test
Accuracy	99.2%	70.6%	69.8%

99%의 정확도를 보이는 Train 데이터에 과적합 문제가 발생하였다.

간단한 CNN 모델 => 과적합 문제 발생

# 2차 모델 학습

## 나) 2차 모델 학습

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 32)	288
batch_normalization (Batch Normalization)	(None, 128, 128, 32)	128
re_lu (ReLU)	(None, 128, 128, 32)	0
conv2d_1 (Conv2D)	(None, 128, 128, 64)	18432
batch_normalization_1 (Batch Normalization)	(None, 128, 128, 64)	256
re_lu_1 (ReLU)	(None, 128, 128, 64)	0
dropout (Dropout)	(None, 128, 128, 64)	0
flatten (Flatten)	(None, 1640576)	0
dense (Dense)	(None, 32)	33554464
dense_1 (Dense)	(None, 1)	33
...		
Total params: 33,573,601		
Trainable params: 33,573,489		
Non-trainable params: 192		

→

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 128, 128, 32)	288
batch_normalization_2 (Batch Normalization)	(None, 128, 128, 32)	128
re_lu_2 (ReLU)	(None, 128, 128, 32)	0
max_pooling2d (MaxPooling2D)	(None, 64, 64, 32)	0
conv2d_3 (Conv2D)	(None, 64, 64, 64)	18432
batch_normalization_3 (Batch Normalization)	(None, 64, 64, 64)	256
re_lu_3 (ReLU)	(None, 64, 64, 64)	0
dropout_1 (Dropout)	(None, 64, 64, 64)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 64)	0
...		
Total params: 21,217		
Trainable params: 21,825		
Non-trainable params: 192		

그림.2.12 33,573,601에서 21,217 로 파라미터의 수를 줄임

## 다) 결과 (30 epoch)

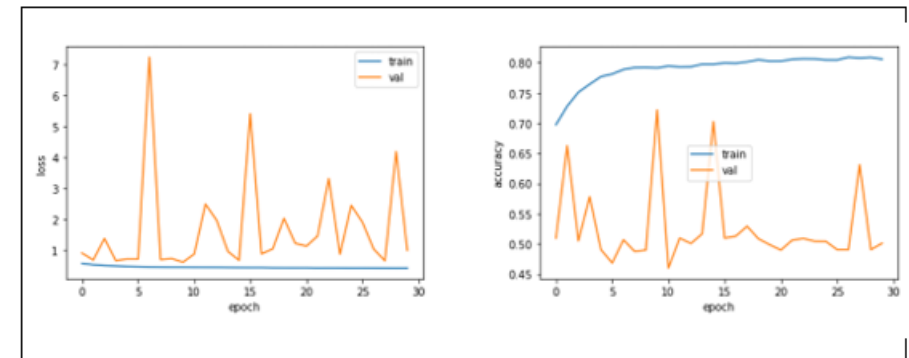


그림.2.13 2차 모델 학습 결과

데이터 증강, Max Pooling, GAP => 여전히 과적합 문제 발생

# 3차 모델 학습

가) 3차 모델 학습

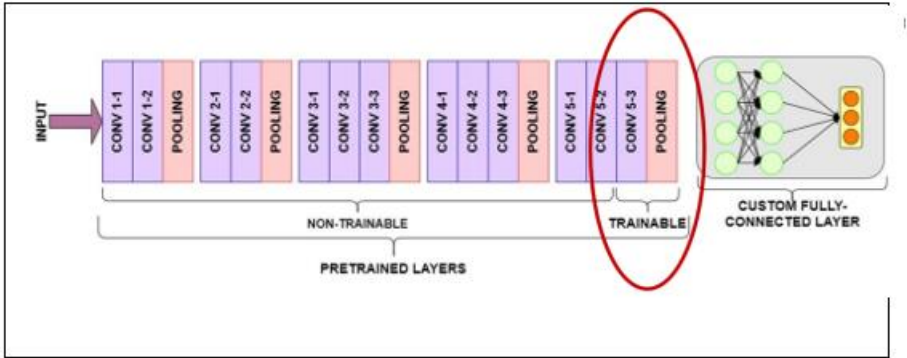


그림2.14 3차 모델

CNNs 모델을 VGG16 사전 모델로 교체한 후 모델을 학습시켰다.

나) 결과 (30 epoch)

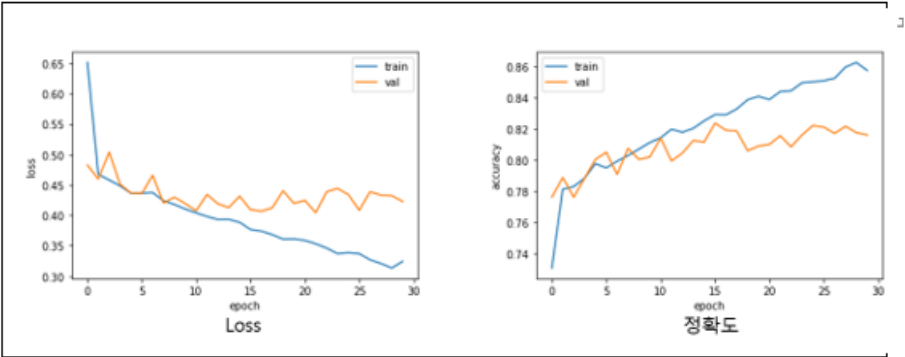


그림2.15 3차 모델 학습 결과

	Train	Validation	Test
Accuracy	82.91%	82.36%	83.66%

모델 학습 결과 2개의 클래스를 분류 정확도 83.66%로 유의미한 정확도가 나왔다. 또한 이전에 있었던 과적합 문제가 해결되어 Train 데이터와 함께 학습이 진행된 결과를 확인할 수 있었다.

사전학습 모델 VGG 16=> 과적합되지 않고 학습이 진행됨, 유의미한 결과

# 4차 모델 학습

가) 라벨링: 3개의 클래스

- 1: 비 사고 다발지역
- 2: 사고 다발지역 약 ~ 중
- 3: 사고 다발지역 강

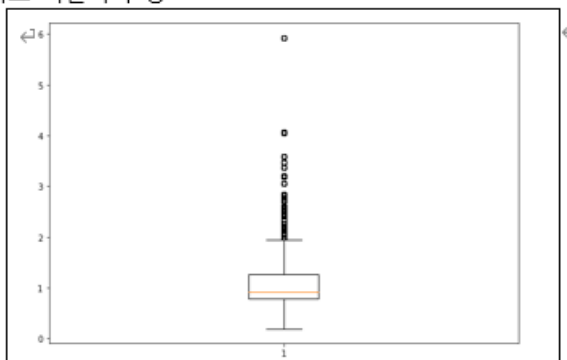


그림2.16 사고 다발지 박스 플롯

나) 결과 (100 epoch)

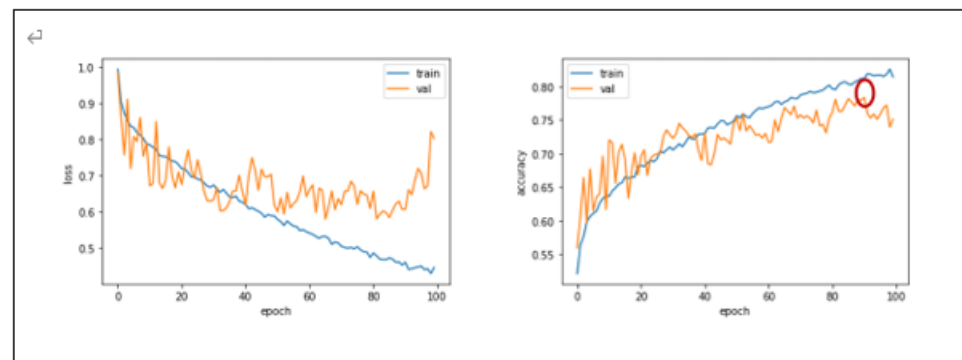


그림2.17 4차 모델 학습 결과

	Train	Validation	Test
Accuracy	81.13%	78.38%	77.17%

2개의 클래스를 분류하는 3차 모델보다 정확도가 떨어진 것을 확인할 수 있었다.

클래스를 3개로 증가 => 2개 클래스 보다 떨어진 정확도

# 최종 모델 학습

가) 최종 모델 학습

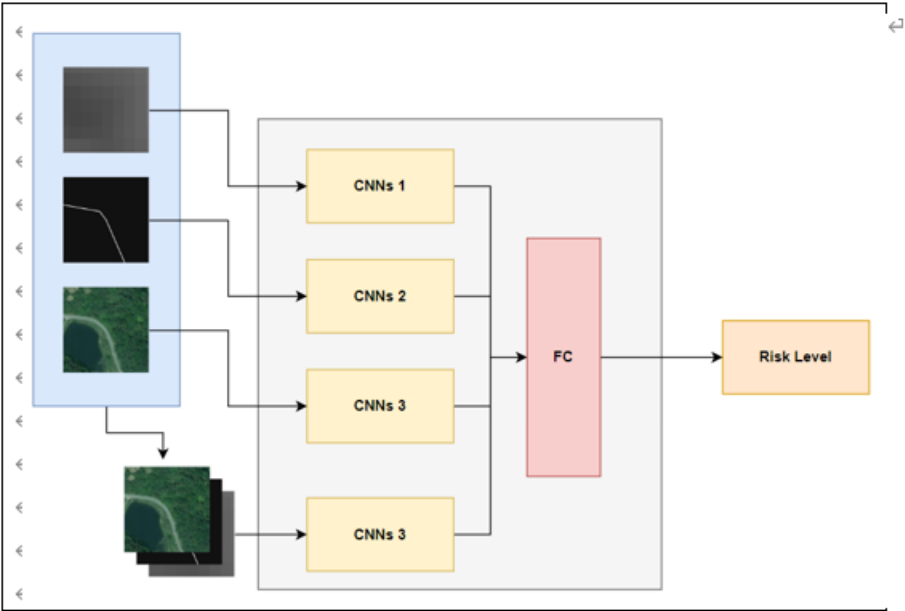


그림2.18 최종 모델 구조 4개의 CNN을 통해 위험 정도(클래스 3개)를 예측한다.

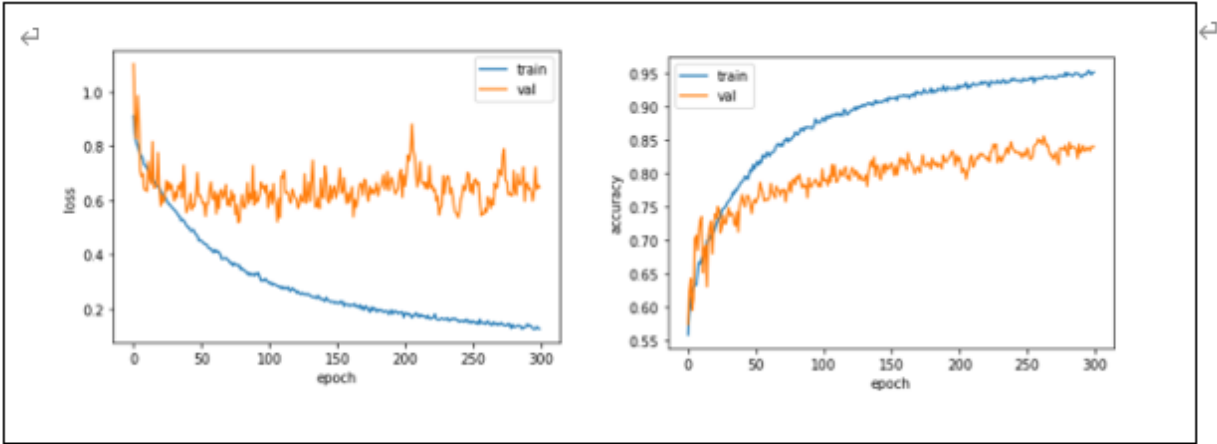


그림2.20 4차 모델 학습 결과

	Train	Validation	Test
Accuracy	94.52%	85.58%	84.37%

클래스를 3개로 증가, 3개의 이미지 사용=> 정확도 84%

# 시각화

## 2.4.8 시각화: 광주광역시 이륜차 사고다발지 예측

84%의 정확도의 모델을 사용하여 광주광역시 도로의 이륜차 사고다발지 예측을 QGIS, Folium 라이브러리, GeoPandas를 사용하여 해보았다.

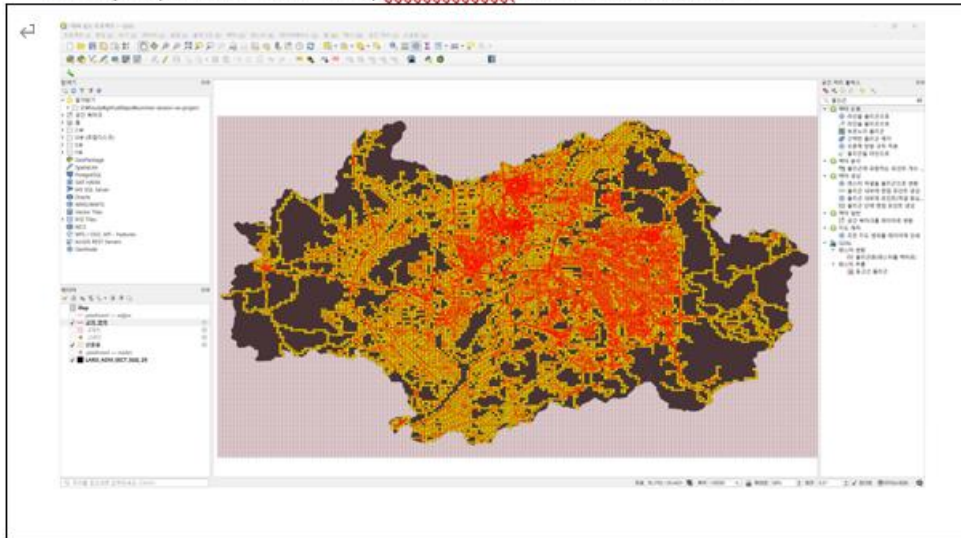


그림 2.21 QGIS를 사용하여 도로 데이터 추출

```
df.head()
```

id	left	top	right	bottom	geometry	center	lon	lat
22847.0	1.412851e+07	4.189621e+06	1.412871e+07	4.189821e+06	MULTIPOLYGON (((126.91951 35.18993, 126.91951 ...	[[126.91860894594058, 35.18919202300463]]	126.918609	35.189192
22846.0	1.412851e+07	4.189821e+06	1.412871e+07	4.190021e+06	MULTIPOLYGON (((126.91951 35.19139, 126.91951 ...	[[126.91860894594059, 35.19066031257187]]	126.918609	35.190660
22845.0	1.412851e+07	4.190021e+06	1.412871e+07	4.190221e+06	MULTIPOLYGON (((126.91951 35.19286, 126.91951 ...	[[126.91860894594056, 35.19212857560577]]	126.918609	35.192129
22819.0	1.412851e+07	4.195221e+06	1.412871e+07	4.195421e+06	MULTIPOLYGON (((126.91951 35.23103, 126.91951 ...	[[126.91860894594058, 35.23029409969932]]	126.918609	35.230294
22818.0	1.412851e+07	4.195421e+06	1.412871e+07	4.195621e+06	MULTIPOLYGON (((126.91951 35.23250, 126.91951 ...	[[126.91860894594059, 35.23176164615013]]	126.918609	35.231762

그림 2.22 추출한 도로 데이터

QGIS로 도로 데이터 추출 => GeoJson

# 시각화

```
1 for idx, row in df3.iterrows():  
2     lat, lon = row['lat'], row['lon']  
3     filename = f'{lat}_{lon}'  
4     imgs = get_image(filename)  
5     if len(imgs) != 3:  
6         print(-1)  
7         latlon_dict[filename] = 0  
8     else:  
9         pred = model.predict(imgs)  
10        latlon_dict[filename] = np.argmax(pred, axis=1)[0]  
11        print('Predicted: ', latlon_dict[filename])
```

코드 2.7 모델을 사용하여 해당 좌표의 위험 정도 (사고다발지)를 예측

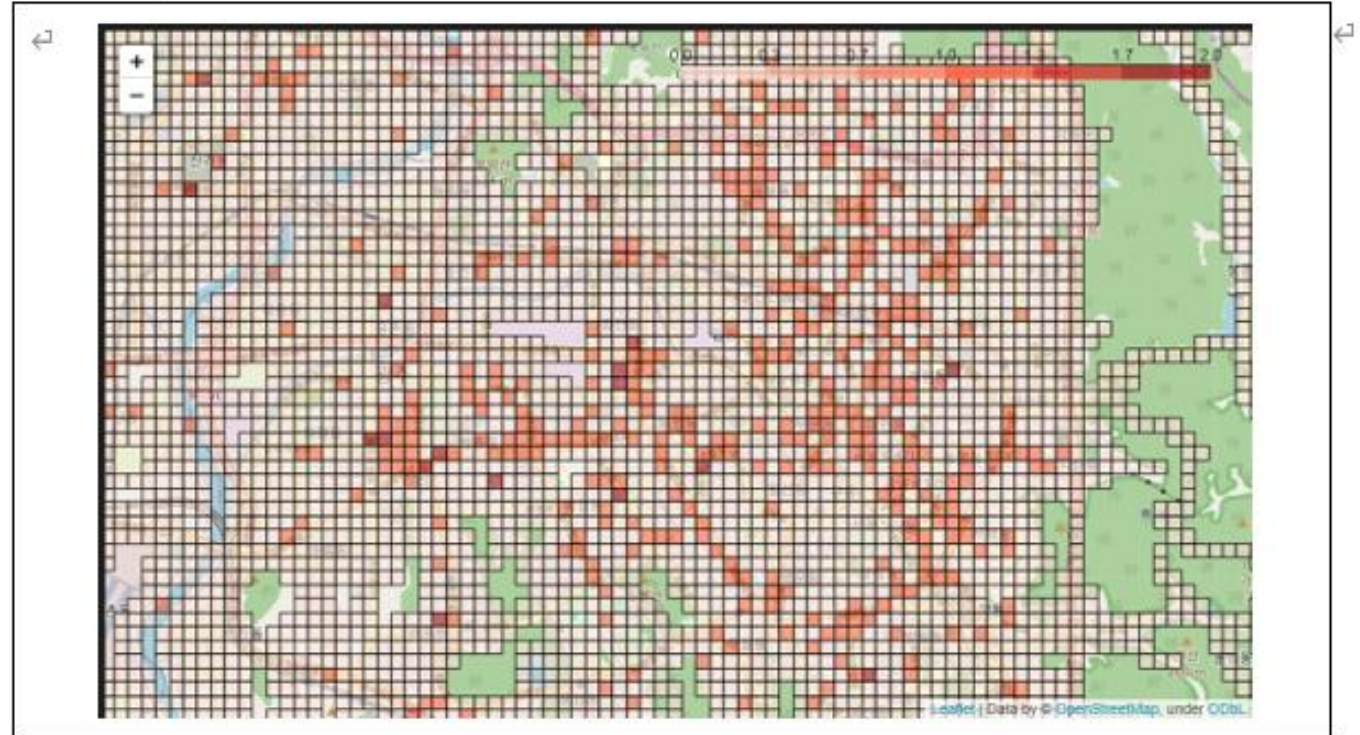
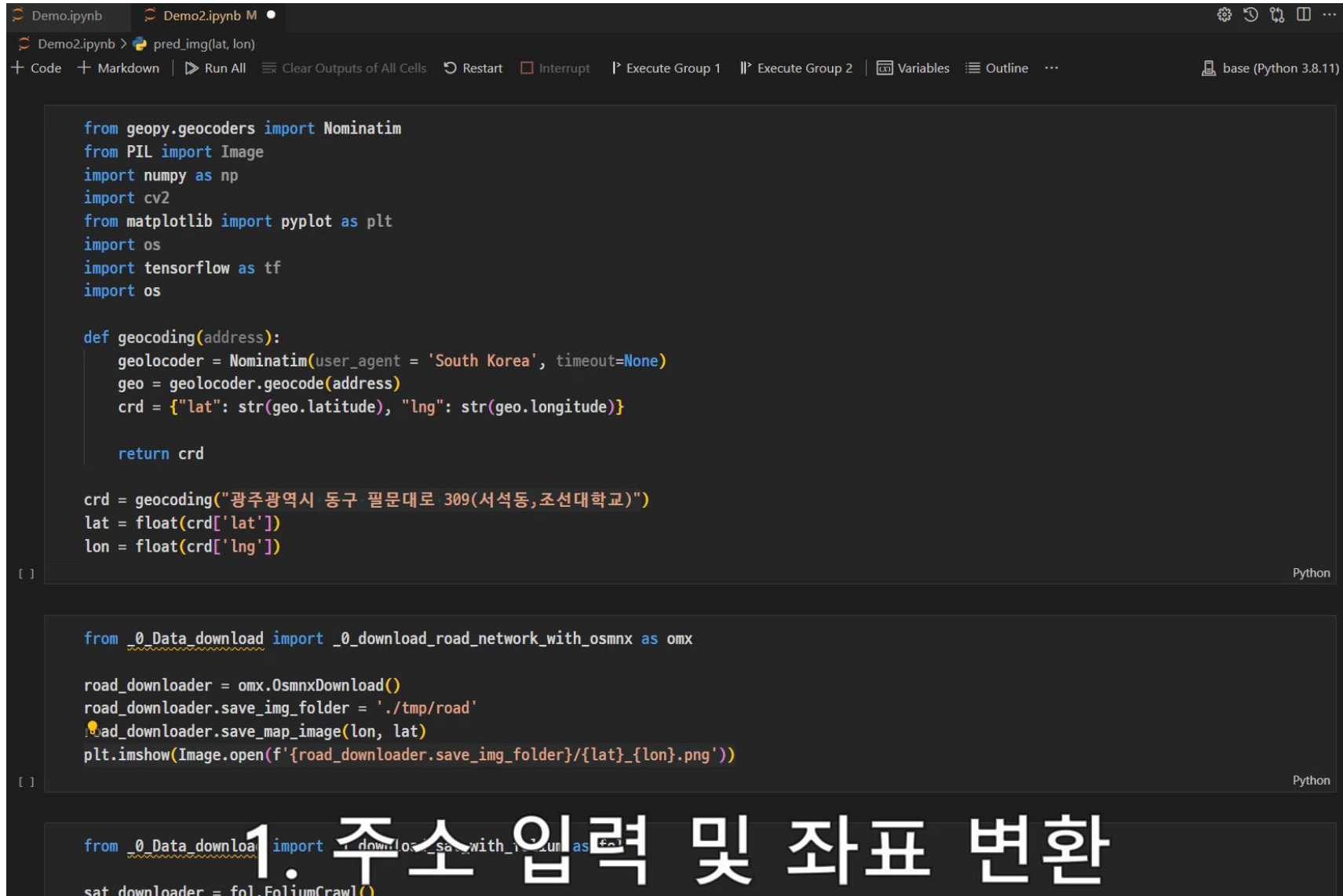


그림2.23 광주 광역시 이륜차 사고 다발지 예측 결과 시각화

모델을 사용하여 광주 광역시 이륜차 사고다발지 예측





```
from geopy.geocoders import Nominatim
from PIL import Image
import numpy as np
import cv2
from matplotlib import pyplot as plt
import os
import tensorflow as tf
import os

def geocoding(address):
    geolocoder = Nominatim(user_agent = 'South Korea', timeout=None)
    geo = geolocoder.geocode(address)
    crd = {"lat": str(geo.latitude), "lng": str(geo.longitude)}

    return crd

crd = geocoding("광주광역시 동구 필문대로 309(서석동,조선대학교)")
lat = float(crd['lat'])
lon = float(crd['lng'])
```

```
from _0_Data_download import _0_download_road_network_with_osmnx as omx

road_downloader = omx.OsmnxDownload()
road_downloader.save_img_folder = './tmp/road'
road_downloader.save_map_image(lon, lat)
plt.imshow(Image.open(f'{road_downloader.save_img_folder}/{lat}_{lon}.png'))
```

```
from _0_Data_download import _0_download_satellite_image as fol
sat_downloader = fol.FoliumCrawl()
```

# 1. 주소 입력 및 좌표 변환

# 결론

프로젝트	정확도	클래스 수	해상도
Combining Satellite Imagery and Open Data to Map Road Safety	78%	3	Google Static Maps API 256×256 pixels, zoom levels (18, 19, and 20).
RoadAccidents	78%	2	200m x 200m
NYC_Traffic_Safety_Project	74.29%	5	?
Traffic-Accidents	82%	2	?

CNN을 사용하여 이륜차 사고 다발지역을 예측할 수 있었음.

정확도 84% 로 다른 프로젝트보다 성능이 더 좋게 나타남

# 기대효과

- 위험한 도로를 피할 수 있게 해주는 응용 프로그램 개발 가능
- 도로 공사, 도로 정비시 도로의 형태를 보고 사전에 위험한지 예측 가능