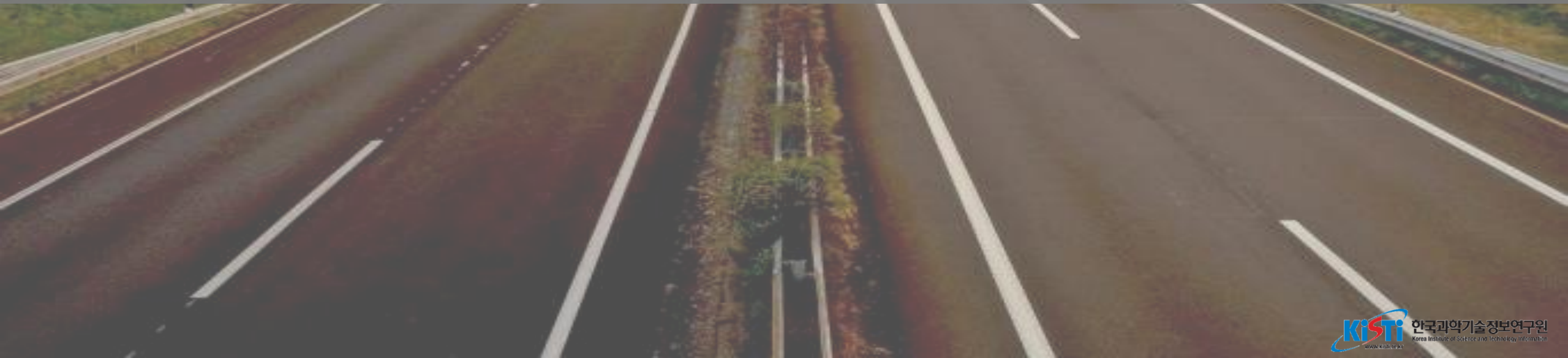
대전광역시 도로 파손 인식 모델 개발과 활용

팀명 : 종균쇠
윤종찬, 소찬균

2021.11.16

KiSTi 한국과학기술정보연구원
www.kisti.re.kr
Korea Institute of Science and Technology Information

CONTENTS



01 프로젝트 개요

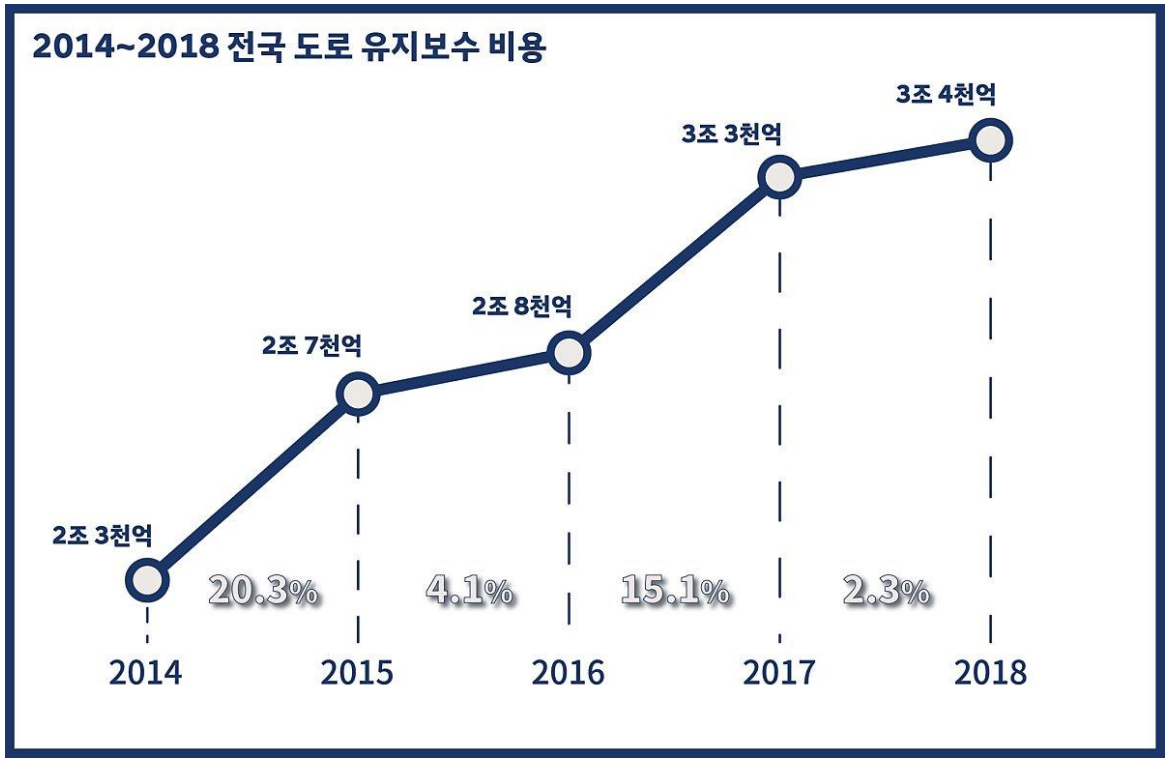
1-1. 프로젝트 제안 배경

도로 유지관리

도로 포장상태는 상태가 크게 떨어지기 전에 예방적 유지관리를 하는 것이 중요하며, 성능이 저하되어 재포장 또는 재건설을 하는 경우 성능이 크게 떨어지기 전의 보수비용보다 6~10배 많이 소요됨

효율적 도로유지관리의 필요성

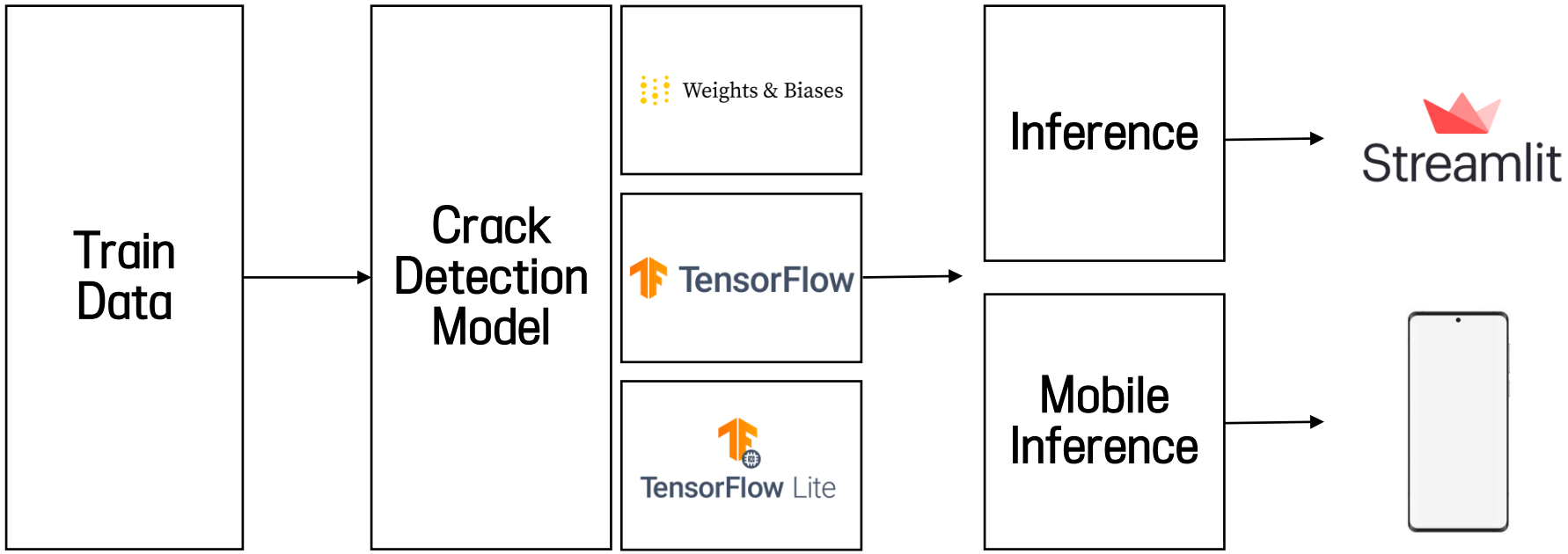
국내 도로 인프라의 경우 노후화가 빠르게 진행되고 있으며, 이로 인해 도로 성능 유지를 위한 관리비용이 증가하고 있음
인프라의 노후화가 빠르게 진행되고 있고, 이에 따른 도로 유지 관리비 비중이 지속적으로 증가할 것으로 예상되고 있어 도로유지관리 정책을 보다 효율적으로 추진할 필요성이 있음



https://m.post.naver.com/viewer/postView.naver?volumeNo=18796572&memberNo=30619985

1-2. 프로젝트 설계 흐름

프로젝트 설계 흐름도



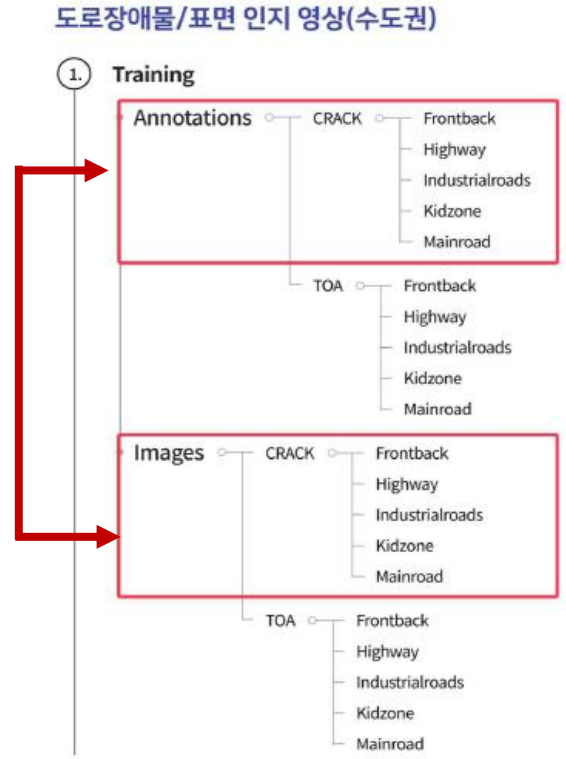
02 활용 데이터

2-1. Train Data Set

AI Hub 도로장애물/표면 인지 영상 (수도권)

- CRACK은 도로 손상에 대한 정보로, SemanticSegmentation 에 활용할 수 있는 format으로 구성되어 있음
- Annotations는 Image에 대한 데이터 셋 정보, 이미지 정보, Annotation 정보를 담고 있음

활용데이터



https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=179

2-2. Test Data Set

대전시 도로 영상 객체 인식 데이터 셋(kisti)

개요

- KISTI가 자체적으로 수집한 도로영상 비디오를 활용한 영상객체 인식용 학습데이터셋
- 대전시 도로 영상데이터에 대해 객체* 라벨링 수행

* 승용차, 버스, 소형트럭, 대형트럭, 바이크, 사람, 공사표지판, 스피드 범프, 포트홀, 크랙, 맨홀, 얼굴, 번호판 등 13종

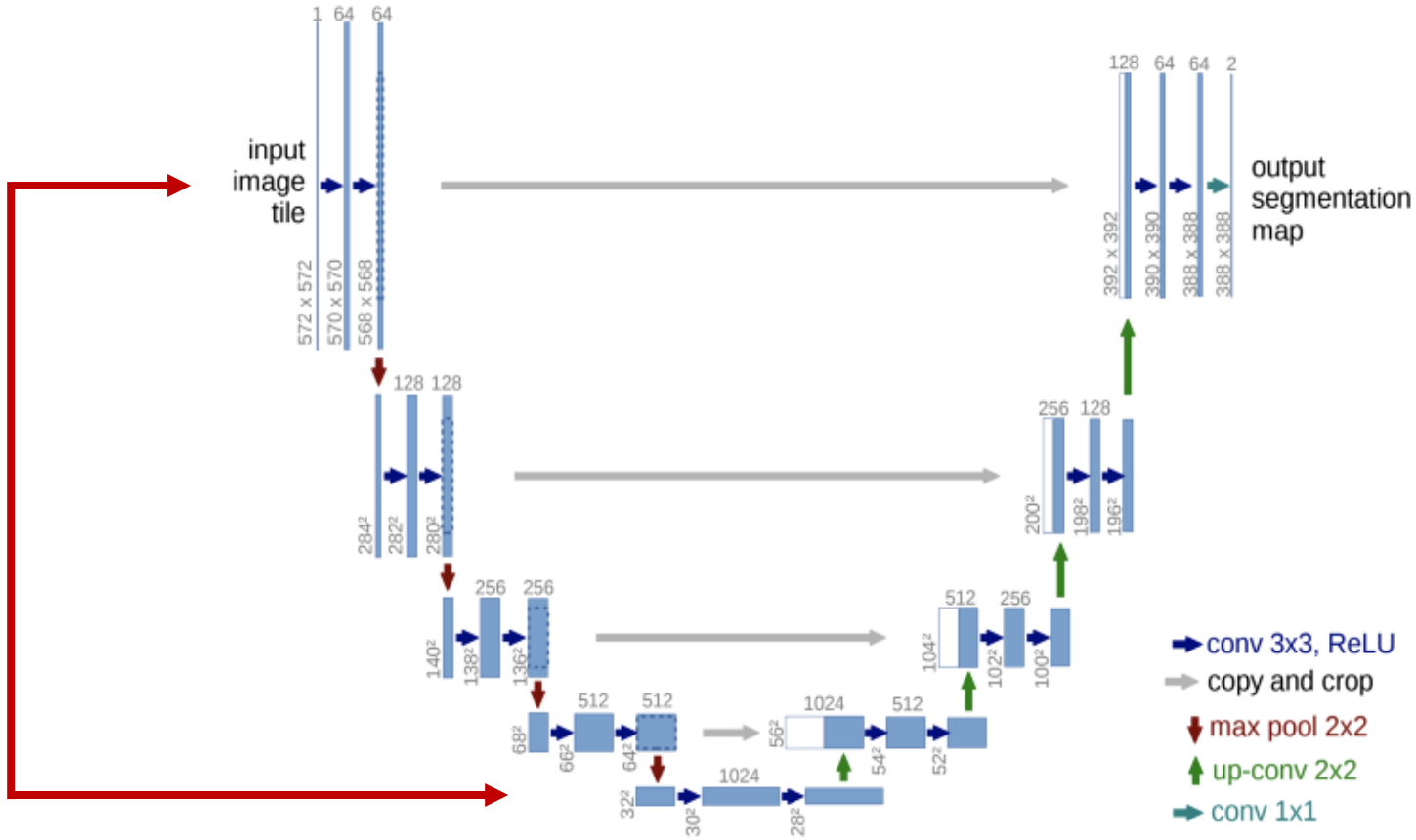
형식

- 2020년 7 ~ 9월 전용차량으로 대전시 주요 도로 주행 시 촬영한 고프로 영상데이터를 대상으로 구축
- 객체 라벨링 데이터는 BOX정보로 구성되어 있으며, 촬영시각/촬영위치 정보를 포함하고 있음.
- 이미지 28.6GB(이미지 49,218장/ 태깅 데이터 255,952건), JSON 223MB
- 도로 이미지 : JPG * 얼굴, 번호판 등 개인정보 비식별화 완료(mosaic 처리)
- 객체 정보 : JSON (KISTI Data Set 형식)

03 모델 개발 방법

3-1. MobileNetV2 기반 UNet

MobileNetV2의 block들을
이용하여 Unet의 Encoder로 사용



Unet 구조

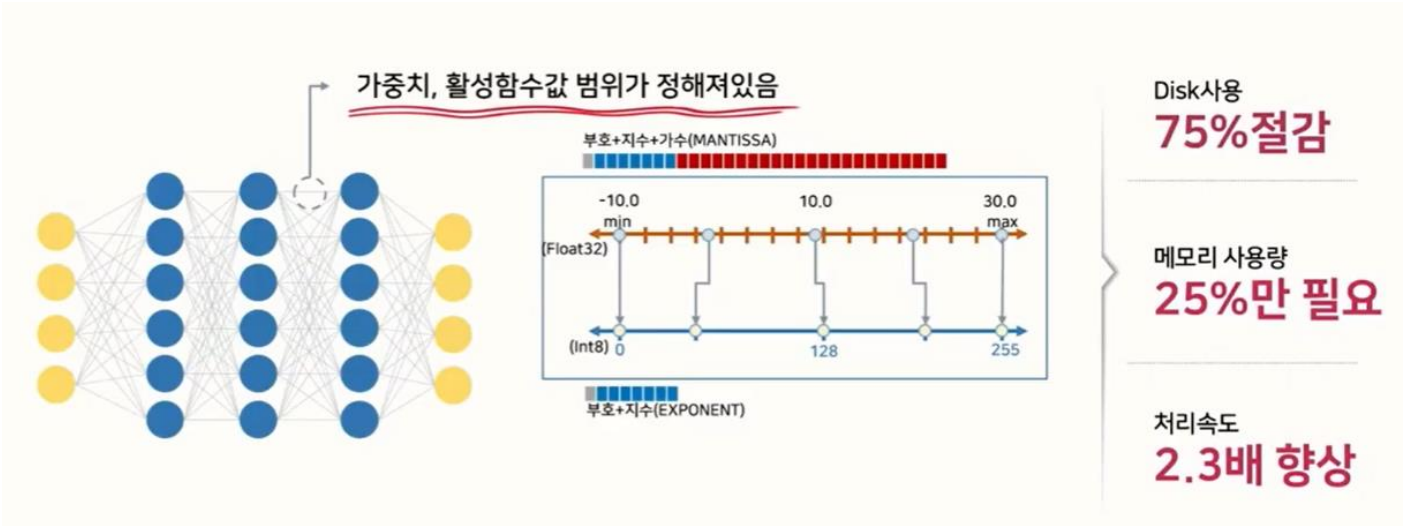
3-2. Quantization Tensorflow Lite Model

Quantization이란?

Quantization은 실수형 변수(floating-point type)를 정수형 변수(integer or fixed point)로 변환하는 과정을 뜻한다.

왜 quantization을 하는가?

Quantization을 통하여 float 타입을 int형으로 줄이면서 용량을 줄일 수 있고, bit수를 줄임으로써 계산 복잡도도 줄일 수 있음
정수형이 하드웨어에 좀 더 친화적임



즉, quantization을 수행함으로써 딥러닝 모델을 경량화 하고, edge device에서 빠른 추론을 달성할 수 있도록 함
따라서 모바일에서 딥러닝 모델을 돌리기 위해 기존에 학습된 모델을 quantization을 적용한 tensorflow Lite로 변환하는 과정을 거침

Quantization

Tensorflow Lite

04 실험 및 평가

4-1. Model별 Training

실험 구성 및 결과

모델에 따라서 성능이 얼마나 차이나는가를 비교
성능 측정을 위한 metric은 *Dice Score를 사용
학습 시, Train : 9020장, Validation : 1002장, Total : 10022장 사용
1000개의 Test image dataset을 사용

실험 구성	
Image Size	256x256(가로x세로)
Base Model(Encoder)	MobileNetV2
Loss	Categorical Cross-Entropy
Batch Size	32
Epochs	30
Learning Rate	1e-3
Learning Rate Decay	Cosine Decay
Optimizer	Adam

* Dice Score

$$Dice = \frac{2 \times TP}{(TP + FP) + (TP + FN)}$$

Training

Result

Model	Dice Score
UNet	0.35
DeepLabV3+	0.27
PAN	0.24

4-2. Base Model별 Training

실험 구성 및 결과

Base model에 따라서 성능이 얼마나 차이나는 가를 비교함
성능 측정을 위한 metric은 이전과 동일하게 Dice score를 사용함
학습 시, Train : 9020장, Validation : 1002장, Total : 10022장 사용
1000개의 Test image dataset을 사용

실험 구성	
Image Size	256x256(가로x세로)
Model	UNet
Loss	Categorical Cross-Entropy
Epochs	30
Learning Rate	1e-3
Learning Rate Decay	Cosine Decay
Optimizer	Adam

Training

Result

Base Model	Batch Size	Dice Score
MobileNetV2	32	0.35
VGG16	16	0.40

But VGG16 기반 UNet 모델은 용량이 커서 성능 대비 효율을 따져 본다면 MobileNetV2가 더 낫다고 생각함
따라서 Unet+MobileNetV2를 이용하여 추가적인 모델 학습을 진행

4-3. Test Data 평가

최종 모델 학습 구성 및 결과

다음과 같은 구성으로 최종적으로 모델 학습을 진행하고, quantization을 적용한 Tensorflow Lite model을 생성함
학습 시, Train data : 46316장, Validation data를 : 5146장, Total : 51462장 사용
1000개의 Test image dataset을 사용

실험 구성	
Image Size	384x288(가로x세로)
Model	Unet
Loss	Categorical Cross-Entropy
Epochs	20
Batch Size	16
Learning Rate	1e-3
Learning Rate Decay	Cosine Decay
Optimizer	Adam

Training

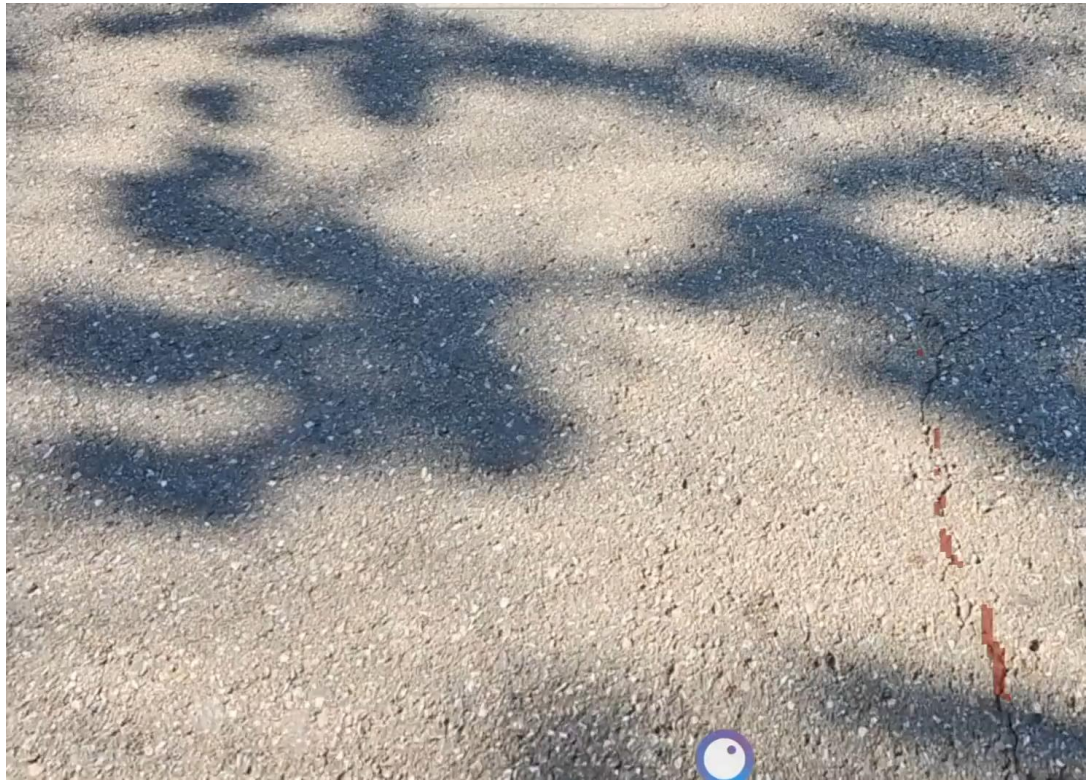
Result

Dice Score
0.46

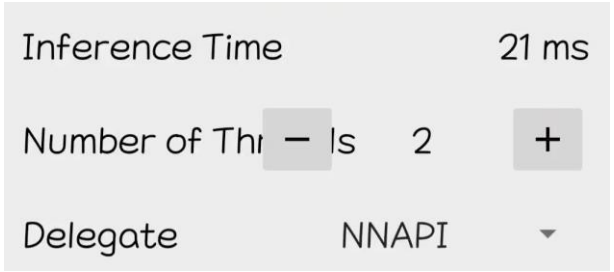
4-4. Deep learning model in Moblie

모바일에서의 성능

Device : Galaxy Z flip 4



평균적으로 약 20~30ms의
추론 속도를 보여줌



4-5. Inference 구조화

Inference 시 Masking Rate 추출

Inference 로직은 기본적으로 이미지, 비디오에 대응하도록 설계
Test Dataset : 대전시 도로 영상 객체 인식 데이터셋 (49218장 이미지 사용)
추론 시, .json 파일을 로드하여 대응하는 Masking Rate를 .csv 파일로 저장

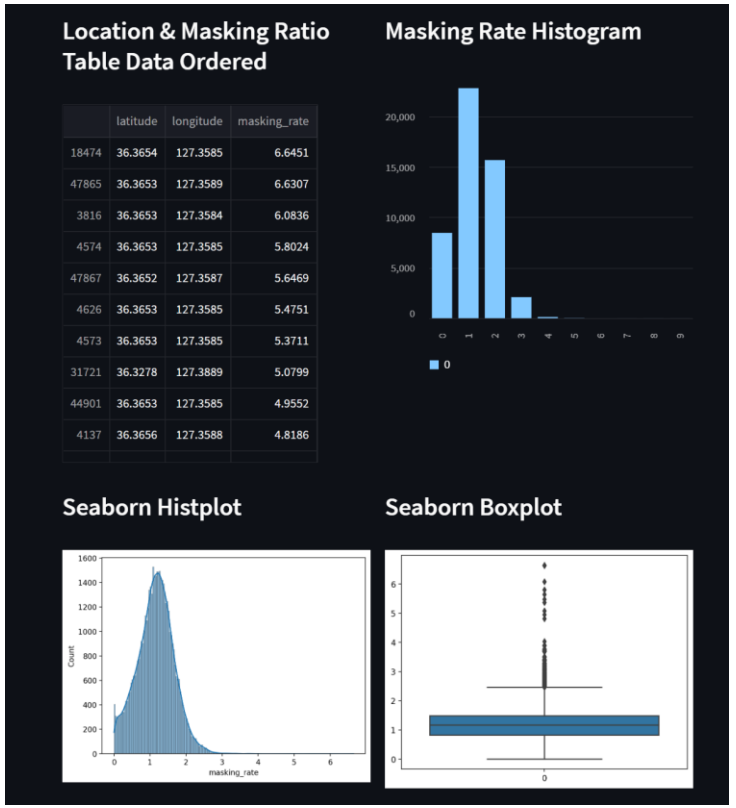
$$Making\ Rate = \frac{Num\ of\ Masking\ Pixels}{Num\ of\ Total\ Pixels}$$

Streamlit Dashboard

Inference 과정에서 추출한 Masking Rate 에 대한 결과를 Streamlit을 통해 시각화
Streamlit을 활용한 Dashboard 구성

- Model Environment
- Result DataFrame (ordered)
- Histogram (streamlit built-in)
- Histogram (seaborn)
- Boxplot (seaborn)
- Heatmap (pydeck)
- Grid Layer (pydeck)

Result



4-6. Test Data 평가

추론 결과 DataFrame

Columns	Dtypes	Example
path	object	./images/GH040389/GH040389.MP4#t=449.jpg
latitude	float64	36.317938
longitude	float64	127.417693
mask_indices	object	[118, 115, 116, 117, 112 , ···..]
mask_indptr	object	[0, 0, 0, 0, 0 , ···]
masking_rate	float64	2.464916

4-7. Test Data Dashboard – Streamlit (1)

모델 운용 환경

Model Environment

Training

CPU : Intel 12th 12600KF

VGA : Nvidia RTX 3060 D6 12GB

RAM : Samsung DDR4-25600 32GB(Dual)

Python Version 3.9.13

Inference

CPU : Intel 12th 12600K

VGA : Nvidia RTX 3070Ti D6x 8GB

RAM : Samsung DDR4-25600 32GB(Dual)

Python Version 3.9.13

Requirement

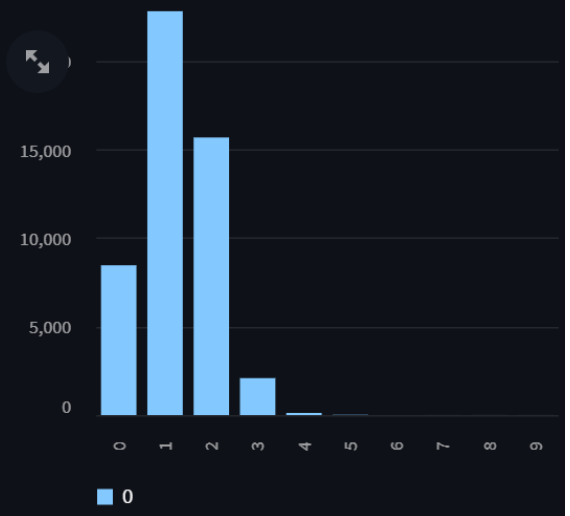
	libarary	version
0	absl-py	1.2.0
1	altair	4.2.0
2	anyio	3.6.2
3	argon2-cffi	21.3.0
4	argon2-cffi-b	21.2.0
5	asttokens	2.0.8
6	astunparse	1.6.3
7	attrs	22.1.0
8	backcall	0.2.0
9	beautifulsou	4.11.1

추론 결과 (정렬) & 히스토그램 (내장함수)

Location & Masking Ratio Table Data Ordered

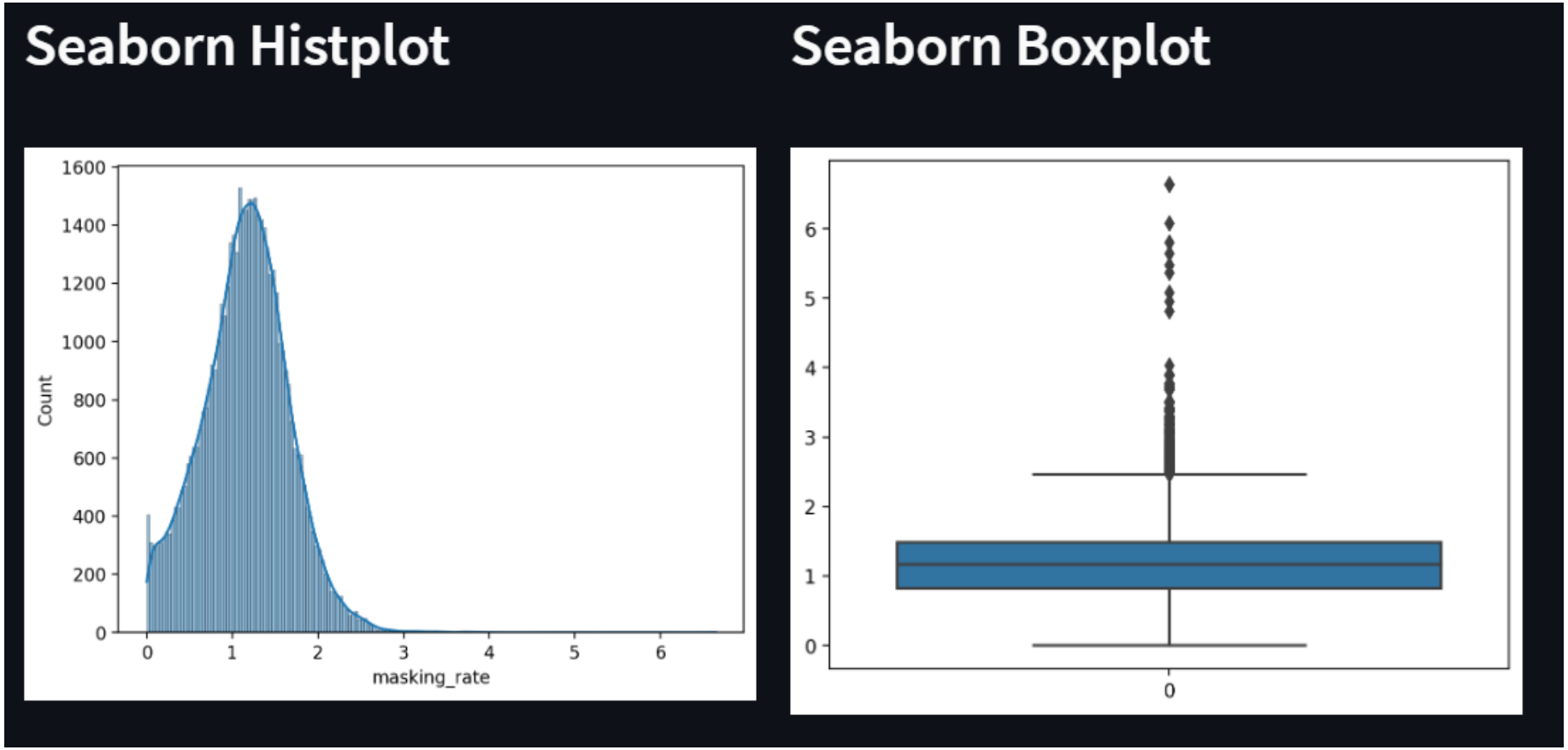
	latitude	longitude	masking_rate
18474	36.3654	127.3585	6.6451
47865	36.3653	127.3589	6.6307
3816	36.3653	127.3584	6.0836
4574	36.3653	127.3585	5.8024
47867	36.3652	127.3587	5.6469
4626	36.3653	127.3585	5.4751
4573	36.3653	127.3585	5.3711
31721	36.3278	127.3889	5.0799
44901	36.3653	127.3585	4.9552
4137	36.3656	127.3588	4.8186

Masking Rate Histogram



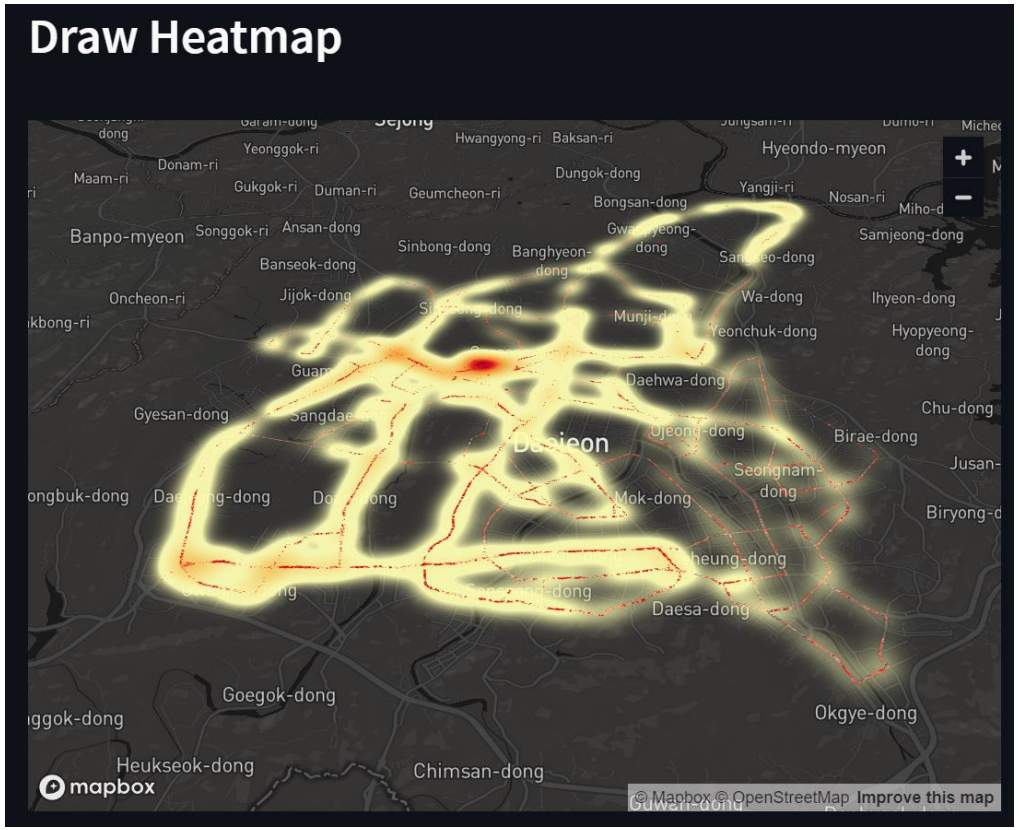
4-7. Test Data Dashboard – Streamlit (2)

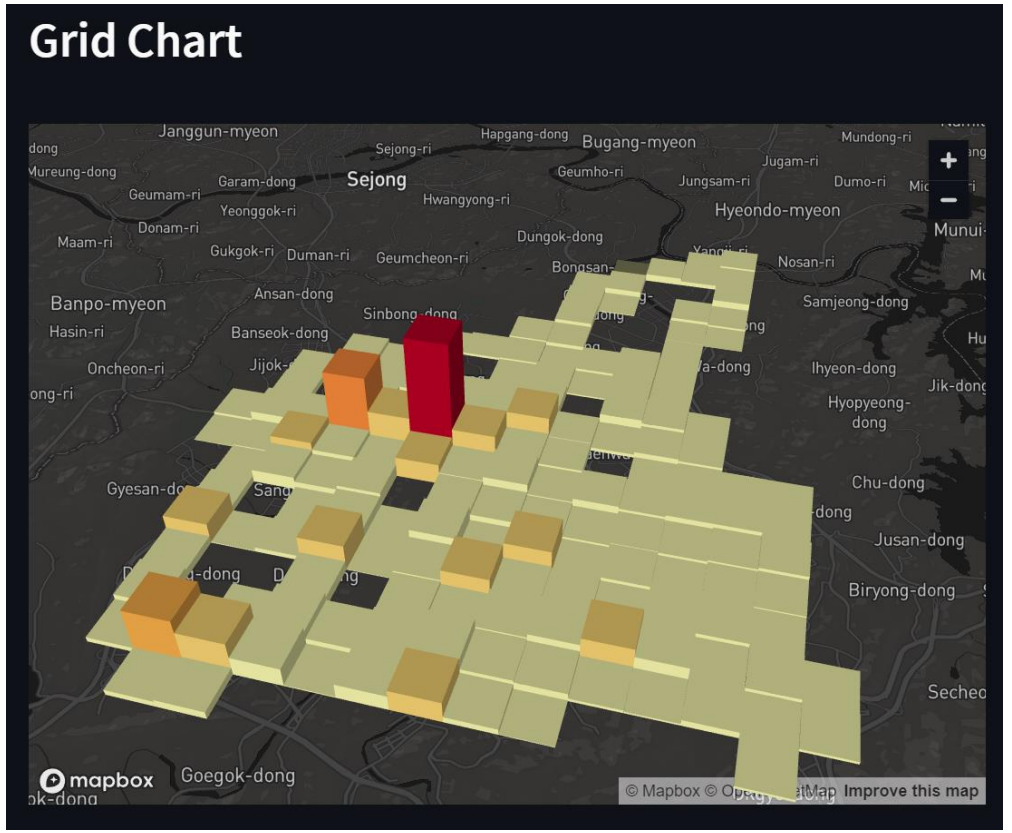
seaborn chart



4-7. Test Data Dashboard – Streamlit (3)

pydeck chart



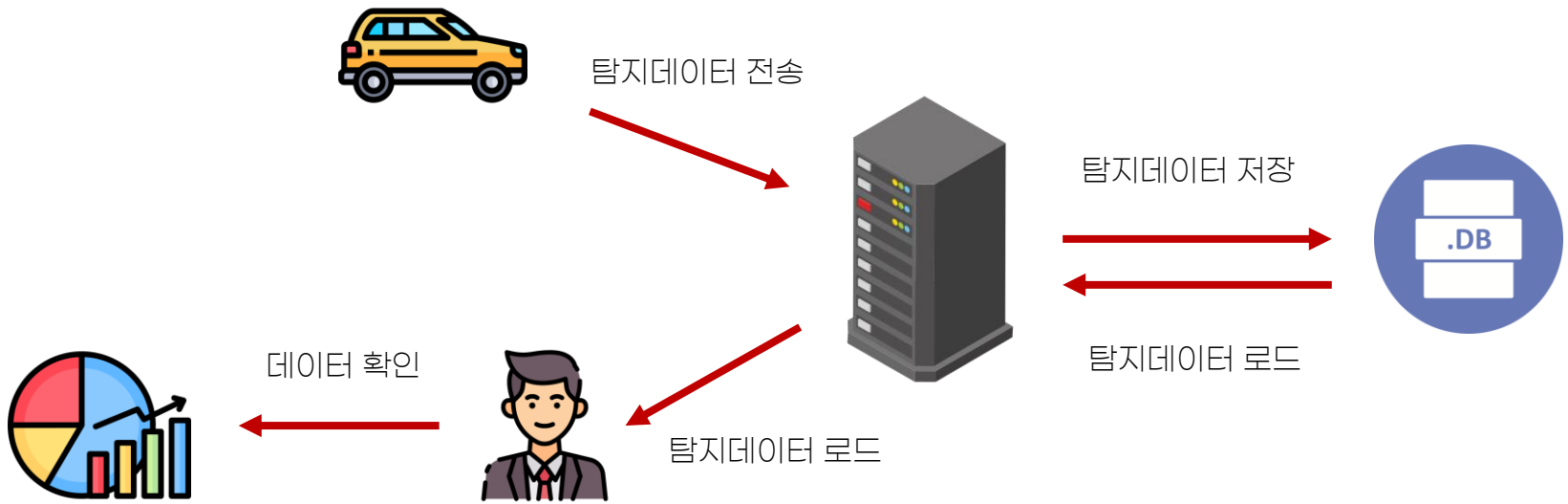


05 활용 계획 및 기대 효과

5-1. 활용 계획

딥러닝을 활용한 크랙 자동 탐지시스템

- 별도의 장비 필요없이 모바일 폰만으로 도로의 크랙을 탐지하고, 탐지 데이터를 서버로 전송하여 데이터 베이스에 도로의 상태를 저장하도록 함
- 이를 통해 도로의 유지보수 프로세스를 더욱 쉽게 만들어 줄 수 있음
- 추후 이를 기반으로 크랙 예측 머신러닝 모델을 생성하여, 어떤 지역에서 크랙이 많이 발생할지 예측해 볼 수 있음



5-2. 기대 효과 및 한계점

기대 효과

- **비용 절감** : 고성능의 장비를 부서별로 보유하지 않더라도 모바일 폰으로 탐지 과정을 수행할 수 있음
- **실시간 처리** : 빠른 반응의 Detector로 데이터 수집 시 실시간 모니터링 가능
- **단계 절감** : 데이터 수집 후 Streamlit을 통한 빠른 요약 보기 가능

한계점

- **한정된 Inference** : Inference 대상 데이터셋이 대전시로 한정되어 있음
- **Streamlit Dashboard** : 운용 기관이 필요로 하는 정보를 명확히 알지 못함으로 한정된 정보만을 제공
- **요구되는 앵글** : Training Data에서 사용된 Data와 마찬가지로, 차량 전면부 노출을 최소화 해야 함

감사합니다.