
2021년 공공빅데이터 분석 청년 인턴십 결과보고서

부평구 내 손상 도로 보수를 위한 Crack Detection

2022. 02

21-13212	윤 종 찬
21-12199	소 찬 균

인천광역시 부평구청

목 차

I . 프로젝트 개요	1
1. 배경 및 필요성	1
2. 수행 기간	1
3. 수행 목표	2
II . 활용 데이터	2
1. 활용 데이터 개요	2
2. 활용 데이터 세부	3
III . 프로젝트 과정	6
1. 프로세스 개요	6
2. 프로세스 환경	7
IV . 모델링 및 웹 서비스 구현	7
1. 전처리 과정 설계	7
2. 딥러닝 모델링	8
3. 모델 테스트	12
4. 웹 서비스 구현	14
V . 결론	19
1. 활용 방안	19
2. 한계점	19
VI . 참고 자료	20

I. 프로젝트 개요

1. 배경 및 필요성

1.1. 프로젝트 배경

① 주기성을 갖는 도로 손상

- 여름철 장마, 겨울철 제설 작업 이후 도로 노후화 및 손상이 발생
- 국토교통부를 비롯한 지자체에서는 도로의 손상에 대처하기 위해 도로 보수 계획을 수립

② 다양한 형태의 데이터 개방

- 데이터 경쟁력 제고를 위해 다양한 형태의 데이터가 개방되어 있음
- 단순 정형 데이터부터 음성, 사진, 영상 등의 비정형 데이터까지 활용 가능한 데이터들을 손쉽게 획득 가능

1.2. 프로젝트 필요성

① 손상 도로 보수

- 도로 보수 계획 수립에 있어서 선제적으로 행해지는 파손 도로 점검을 수행하는 단계에서 시간 단축 가능
- 모델의 무결성을 온전히 신뢰할 수 없더라도, 선제 조사 시 도로 파손 심각성 수준 점검 가능
- 단순 영상을 기반으로 손상 도로를 점검하므로 초기 비용 절감에 유리

② 적극적인 개방 데이터 활용

- 개방된 데이터를 통해 활용 가능한 수준의 서비스를 제공하면서 유의미한 가치 창출 가능
- “YOLO V5”와 같은 object detection 모델과 접목 시 주행 장애물 인식에 추가적인 기능으로 활용 가능

2. 수행 기간

- 프로젝트 기간 : 2022년 01월 03일 ~ 2022년 02월 28일 (8주)
- 프로젝트 주 차별 Work Flow

표 1 <주차 별 Work Flow Heat Map>

1주 차					
2주 차					
3주 차					
4주 차					
5주 차					
6주 차					
7주 차					
8주 차					

표 2 <Work Flow Heat Map 색상표>

	주제 선정
	데이터 수집 및 전처리
	논문 조사 및 모델 탐구
	딥러닝 모델링
	웹 서비스 개발 및 보고서 작성

3. 수행 목표

- (모델 관점) 영상 파일(.mp4) input 시에 도로 위 손상 라인을 검출하는 딥러닝 모델 개발
- (서비스 관점) 모델 활용을 웹으로 구현하여 네트워크상으로 접근하고 결과물을 받아내는 웹 서비스 프로토타입 구축

II. 활용 데이터

1. 활용 데이터 개요

1.1. Training Data

- 데이터 출처 : [AI Hub - 도로장애물/표면 인지 영상\(수도권\)](#)

도로장애물/표면 인지 영상(수도권) 소개

<div>소개</div> <div>다운로드</div>			
데이터셋명	도로장애물/표면 인지 영상(수도권)		
데이터분야	자율주행	데이터유형	이미지
구축기관	건국대학교 산학협력단	담당자명	조기훈(건국대학교 산학협력단)
가공기관	에이디스원	데이터 관련 문의처	전화번호 02-2049-6265
검수기관	네오피엠	이메일	sangkwonkim@konkuk.ac.kr
구축 데이터량	100만	구축년도	2020년
버전	1.2	최종수정일자	2021.10.26
소개	수도권 지역 도로상의 장애물 및 도로 표면의 이상 상태 인지를 위한 영상 및 이미지 데이터		
주요 키워드	객체 검출, 시멘트 세그멘테이션, 주행 중 이상 상태 인식		
저작권 및 이용정책	본 데이터는 과학기술정보통신부가 주관하고 한국지능정보사회진흥원이 지원하는 '인공지능 학습용 데이터 구축사업'으로 구축된 데이터입니다[데이터 이용정책 상세보기]		

그림 1 <도로장애물/표면 인지 영상(수도권) 소개>

도로장애물/표면 인지 영상(수도권)

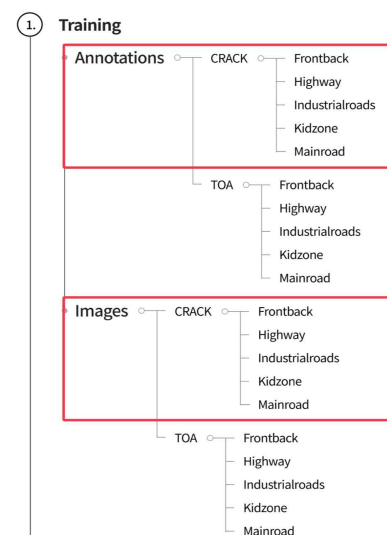


그림 2 <활용 데이터 목록>

- 기본적으로 제공하는 데이터는 Training과 Validation이지만 Validation Set의 오류가 확인되어 Training Set만 활용
- Annotations는 Image에 대한 데이터 셋 정보, 이미지 정보, 어노테이션 정보를 담고 있음
- CRACK은 도로 손상에 대한 정보로, Object Semantic Segmentation에 활용할 수 있는 format으로 구성되어 있음
- TOA는 도로 위 객체에 대한 정보로, Object Detection에 활용할 수 있는 Bounding Box format으로 구성되어 있음. 본 프로젝트에서는 Semantic Segmentation을 통한 Crack 검출이 목표이므로 사용하지 않음

1.2. Test Data

- 부평구 내 주요 도로 및 경로를 선정하여 Video Visualization에 활용
- 최대한 Training Data와 비슷한 환경 설정을 위해 차량을 빌려, 거치대를 통해 유리창 반사를 줄이고 넓은 도로 폭을 촬영할 수 있도록 세팅
- 크게 3개의 경로를 정하고 한 경로당 약 35분, 2분여의 개당 파일을 생성하여 53개의 .mp4 형태의 Test Data 수집

2. 활용 데이터 세부

2.1. Training Data

① 데이터 디렉토리 구조

- Annotations : /도로장애물·표면 인지 영상(수도권)/Training/Annotations/CRACK/
- Images : /도로장애물·표면 인지 영상(수도권)/Training/Images/CRACK/

표 3 <데이터 디렉토리(Crack 하위)>

CRACK	Frontback	C_Frontback_D01 C_Frontback_D02 C_Frontback_E01 C_Frontback_E02 C_Frontback_E03 C_Frontback_F01 C_Frontback_F02 C_Frontback_G01 C_Frontback_G02
	Highway	C_Highway_D01 C_Highway_E01 C_Highway_F01 C_Highway_G01
	Industrialroads	C_Industrialroads_D01 C_Industrialroads_F01 C_Industrialroads_F02 C_Industrialroads_F03 C_Industrialroads_G01
	Kidzone	C_Kidzone_D01 C_Kidzone_E01 C_Kidzone_F01 C_Kidzone_G01
	Mainroad	C_Mainroad_B01 C_Mainroad_B02 C_Mainroad_B03 C_Mainroad_D01 C_Mainroad_D02 C_Mainroad_D02 C_Mainroad_D03 C_Mainroad_D04 C_Mainroad_E01 C_Mainroad_E02 C_Mainroad_E03 C_Mainroad_E04 C_Mainroad_F01 C_Mainroad_F02 C_Mainroad_F03 C_Mainroad_G01 C_Mainroad_G02 C_Mainroad_G03 C_Mainroad_G04

- 데이터 목록 중 C_Mainroad_D01, C_Mainroad_D02, C_Mainroad_D03의 경우 Images와 Annotations가 일치하지 않는 것이 확인되어 활용 데이터에서 제외

② 데이터 구조

- 데이터 네이밍 :
 <차량구분>_<영상장치>_<동영상 No.>_<촬영일자>_<비식별화>_<카메라 No.>_<촬영지역>_<날씨구분>_<도로상태>_<촬영시간구분>_<png no>.PNG
- 예) 2020년 10월 30일 11시 30분, 부산에서 구름 낀 날씨에 촬영한 이미지 :
 V1F_HY_0002_20201030_113045_E_CHO_Busan_Cloud_Mainroad_Day_0005.png

○ 이미지 파일명 포맷

표 4 <image 데이터 파일명 format>

구분	이름	설명
차량 구분	V0F, V1F, V2F, ... VnF	차량 및 운전자 고유번호
영상장치	HY_0002, HY_0015, ...	동영상(1분,MP4) 고유 No.
촬영일시	촬영일: YYYY/MM/DD 촬영시간: hh/mm/ss	(년월일)_20201118
비식별화	N/E	시/분/초_131958
카메라(채널)	CH01, CH02, ...	N:비식별X E:비식별O
촬영지역	Seoul/Busan	수도권, 광역시 구분
날씨 구분	Sun/Cloud/Rain/Fog/Snow	맑음/흐림/비/안개/눈 등
도로상태	Frontback/Highway/Kidzone/ Mainroad/Industrialroads	도심(골목길), 고속도로, 어린이 보호구역, 국(지방)도, 항만/공단
촬영시간 구분	Day/Night/Sunrise/Sunset	낮/밤/일출/일몰 등
PNG No.	_0005	이미지 생성시 PNG 번호

○ 어노테이션 포맷

표 5 <제공 데이터 어노테이션 포맷>

No	항목명	항목설명	타입	필수 구분	단위
1	데이터셋 정보	데이터셋 전체에 관한 전반적인 정보를 포함하는 메타데이터	object		
1	1-1 데이터셋 명		String	Y	
	1-2 데이터셋 상세		String		
	1-3 데이터셋 URL		String	Y	
	1-4 데이터셋 생성일		String		
2	이미지정보	데이터셋을 구성하는 각 이미지에 대한 메타데이터 및 학습 데이터	List		
1	2-1 이미지 식별자		String	Y	
	2-2 이미지 너비		Number	Y	pixel
	2-3 이미지 높이		Number	Y	pixel
	2-4 이미지 파일명		String	Y	
	2-5 이미지 라이선스		String		
	2-6 이미지 촬영일		String		
	2-7 이미지 촬영날씨		String	Y	
	2-8 이미지 촬영시간대		String	Y	
	2-9 원본 영상 정보		String	Y	
	2-10 프레임 순서		String	Y	
3	어노테이션 정보	데이터 셋의 어노테이션에 대한 메타데이터 및 학습 데이터	List		
3	3-1 어노테이션 식별자		String	Y	
	3-2 연관이미지 식별자		String	Y	
	3-3 어노테이션 속성		Object		
	3-4 어노테이션 바운딩박스		List		
	3-5 어노테이션 이미지		Image		

* CRACK 데이터는 Annotation 내부에 polylines 좌표가 존재하므로 이를 활용

2.2. Test Data

① 경로 설정

	Path 1	경로 요약	부평시장~갈산동~삼산동~굴포천~부평4동~갈월2동~부평구청
		시간	2022.02.22.10:23 ~ 2022.02.22.10:57 (34분)
	Path 2	경로 요약	부평구청~부평4동~굴다리오거리~부평공원~동수역~부평삼거리~청천동~인천나비공원
		시간	2022.02.22.11:15 ~ 2022.02.22.11:50 (35분)
	Path 3	경로 요약	인천나비공원~부평우체국~부평시장역~부개동~굴다리오거리~부평시장
		시간	2022.02.22.11:59 ~ 2022.02.22.12:32 (33분)

그림 3 <데이터 수집 경로 도식화>

② 수집 데이터 샘플

○ 데이터 네이밍 :

<촬영일자 및 시간>_<시작 위치 위도>_<시작 위치 경도>_<녹화 세트>.mp4

- 예) 2022년 02월 22일 10시 23분 10초, (37.51768, 126.73178)부터 녹화된 2번째 세트 : 20220222102310_37.51768_126.73178_1000002.mp4

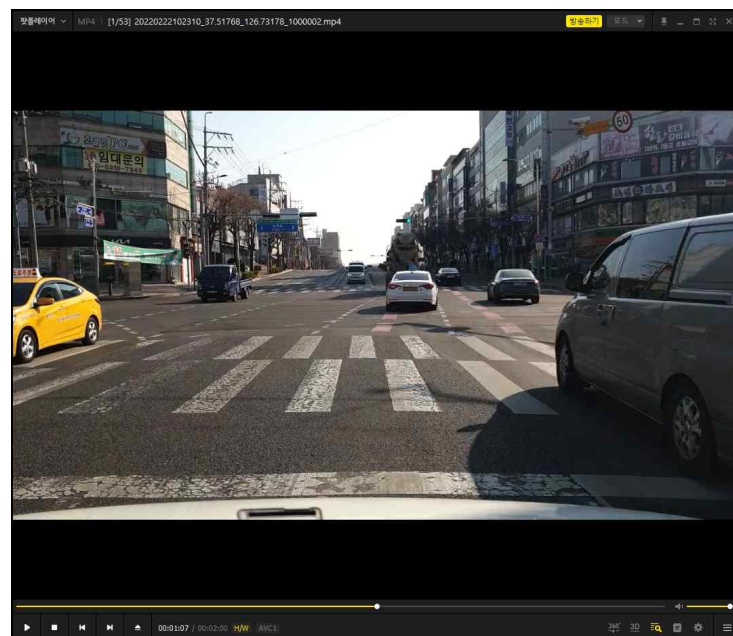


그림 4 <Test Data 샘플 캡처>

III. 프로젝트 과정

1. 프로세스 개요



그림 5 <프로젝트 단계별 과정>

1.1. 데이터 수집

- Ai Hub를 통해 다운로드
- Test에 활용할 부평구 도로 데이터는 차량과 휴대폰 카메라를 이용하여 주요 도로 샘플 데이터 수집

1.2. 전처리 과정 설계

- Images와 Annotations 일치 여부 확인
- Annotations의 Polyline 중 불러오지 못하는 데이터 처리
- Pytorch 기반의 딥러닝 모델에 Data Loading 프로세스 처리

1.3. 딥러닝 모델 설계

- U-Net 기반 Crack Detection 모델 설계
- PAN 기반 Crack Detection 모델 설계

1.4. 모델 테스트

- Image Data에 대한 prediction
- Video Data에 대한 Prediction

1.5. 웹 서비스 구축

- Image Data Prediction 웹 형태의 서비스 제공
- Video Data Prediction 웹 형태의 서비스 제공

2. 프로젝트 환경

○ Test Data 수집 환경

표 6 <Test Data 수집에 세팅한 사용 환경>

차량		촬영 장치		거치 환경	촬영 소프트웨어	
모델	더 뉴 레이	갤럭시S9	프로세서	엑시노스 9	로미러 거치대를 활용하여 로미러 하단 거치 * 하이패스, 자동차 앞이 나오지 않도록 거치	DailyRoads Voyager (Application)
차종	경형		메모리	LPDDR4X 4GB		
연료	휘발유		카메라	후면 OIS 지원 1,200만 화소		

○ 모델링 환경

표 7 <모델링에 세팅한 사용 환경 3가지>

Setting	Devices			Tools		Models	
SET1	CPU	Intel 12th i5-12600KF			Python	3.7.11	UNet(50k data).pth
	GPU	Nvidia Geforce RTX 3060 D6 12GB					
	RAM	삼성전자 DDR4-25600 32G(Dual)			CUDA	11.2	
	SSD	Google Drive local connect					
SET2	CPU	인텔 12th i5-12600K			Python	3.8.8	UNet_b6thdn210k_bilinear.pth
	GPU	Nvidia Geforce RTX 3070 Ti D6X 8GB					
	RAM	삼성전자 DDR4-25600 32G(Dual)			CUDA	11.6	
	SSD	Sk Hynix P31 Gold (PCIe 3.0)					
SET3	CPU	Intel 11th i7-11800H			Python	3.8.8	UNet_b2th5dn200k .pth
	GPU	Nvidia Geforce RTX 3060 Laptop Gpu 6GB					
	RAM	삼성전자 DDR4-25600 32G(Dual)			CUDA	10.2	
	SSD	Sk Hynix P31 Gold (PCIe 3.0)					

IV. 모델링 및 웹 서비스 구현

1. 전처리 과정 설계

① Image Data

- Image Data는 이외의 처리 과정 없이, Scale Factor를 설정하여 Input Size로 변환하는 작업 실시 (예 : Scale Factor=0.5, 1280 x 720 → 640 x 360)

② Annotation Data

- Annotation Data는 '.json' 형식의 파일에서 polyline을 추출하고 이를 numpy 색 공간에 나타내주는 방식으로 과정을 설계함
- 데이터 확인에 있어서 예외가 발생함을 확인할 수 있었는데, 이는 다음과 같이 해결함

표 8 <Annotation Data 예외처리>

예외 사항	처리 내용
Image와 Annotation 불일치	학습 데이터에서 제외
Annotation polyline에 'None' 값	none_json = [[x for x in t[0] if x is not None]]

2. 딥러닝 모델링

2.1 Training Condition

표 9 <Contition>

	UNet	PAN
Image size	640x360	640x384
Mask thickness	5 or 10	5
Validation Ratio	5% or 10%	10%
Batch size	2, 6, 8	64
Optimizer	RMSprop	
Otimizer learning rate scheduler	ReduceLROnPlateau	
Loss	CrossEntropyLoss + Dice Loss	
Score	Dice Coefficient	

2.2. U-Net

① Method

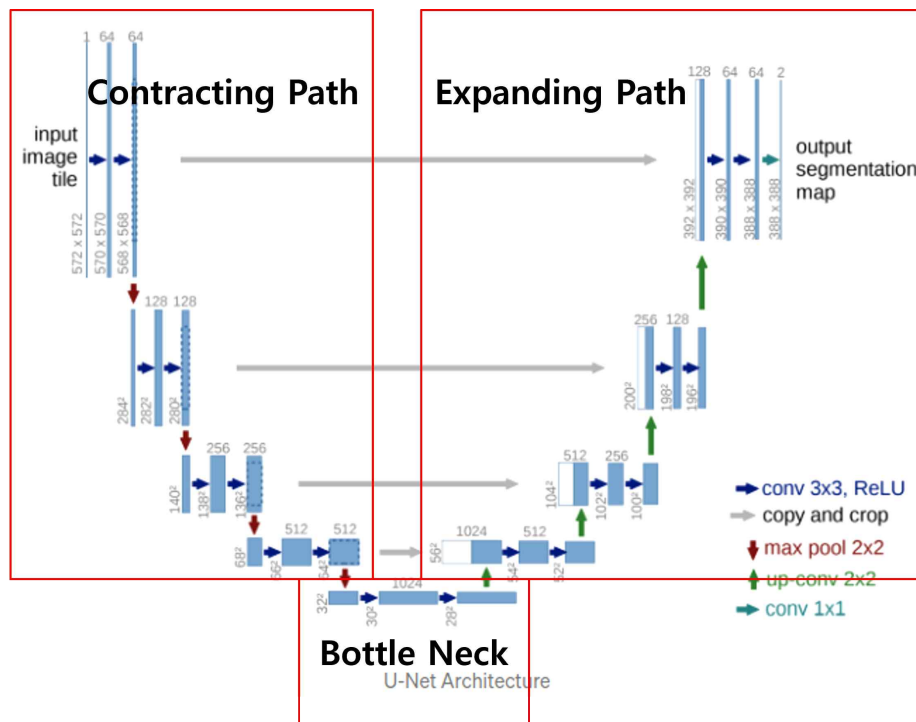


그림 6 <U-Net 도식화>

- Image Segmentation 목적으로 제안된 End-to-End 방식의 Fully-Convolutional Network 기반 모델
- End-to-End 방식은 문제 해결 시 필요한 여러 스텝을 하나의 신경망을 통해 재 배치하는 과정
- 이미지의 전반적인 context 정보를 얻기 위한 네트워크와 정확한 localization을

위한 네트워크의 결합으로 구성됨

* context : 이웃 픽셀 간의 정보 또는 이미지 일부를 보고 이미지 문맥을 파악

- 좌측의 Contracting path는 입력 이미지의 Context 포착을 목적으로 구성, 우측의 Expansive path는 정확한 localization을 위한 구성

② Contracting Path

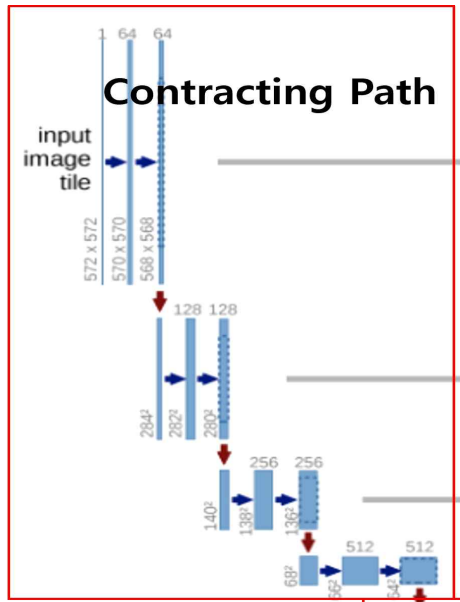


그림 7 <U-Net Contracting Path>

- Contracting Path는 주변 픽셀들을 참조하는 범위를 넓혀 가며 이미지로부터 Context를 추출하는 역할
- Down Sampling 과정을 반복하여 Feature Map을 생성
- 3X3 Convolution 수행 시 padding을 하지 않으므로 Feature Map의 크기는 감소하며, Channel의 수를 2배로 증가시키면서 진행
- 초기 Input Channel 증가를 제외하면, 64→128→256→512→1024개로 증가함

③ Expansive Path

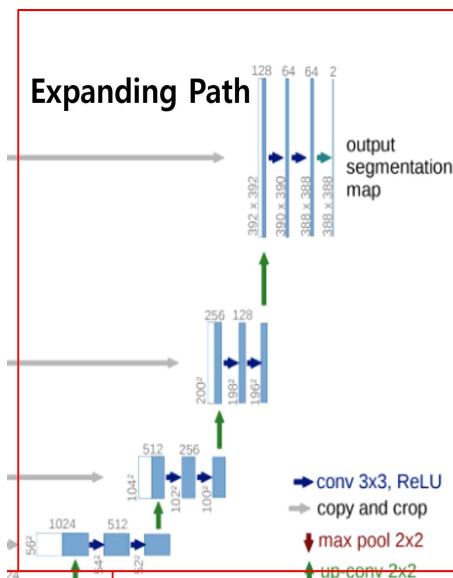


그림 8 <U-Net Expanding Path>

- Expanding Path에서는 Contracting Path에서 생성된 Context와 Local 정보를 결합하는 역할 수행
- Up Sampling 과정을 반복하면서 Feature Map을 생성
- Expanding Step에서 2X2 Convolution 연산 수행 시, Feature Map의 크기는 2배로 늘어남
- Crop 된 Feature Map을 보내, Convolution에서 boarder pixel에 대한 정보 손실에 대한 쌍방 보정을 수행

③ Bottle Neck

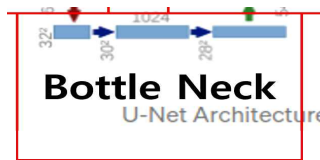


그림 9 <U-Net Bottle Neck>

- Contracting Path에서 Expanding Path로 확장되는 구간
- 마지막에 Dropout Layer가 있다면 Model Generalize와 Noise Robust 역할을 수행

2.3 PAN(Pyramid Attention Network)

① Method

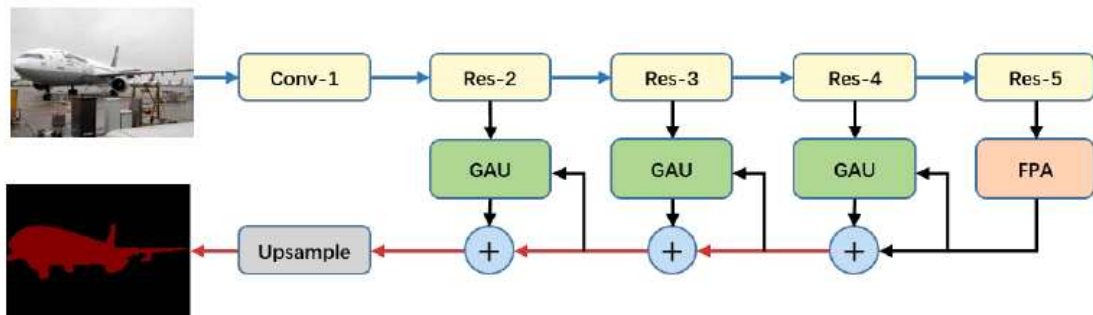


그림 10 <PAN 모델 구조>

- PAN의 전체적인 구조는 위와 같음. 크게 Encoder-Decoder 구조를 띄고 있으며, 주요 특징으로는 GAU(Global Attention Upsample module), FPA(Feature Pyramid Attention module)를 가지고 있음
- Dense feature들을 추출하기 위해서 Resnet을 사용하며, 파란색 화살표는 Downsample을 의미하고, 빨간색 화살표는 Upsample을 의미함

② Feature Pyramid Attention

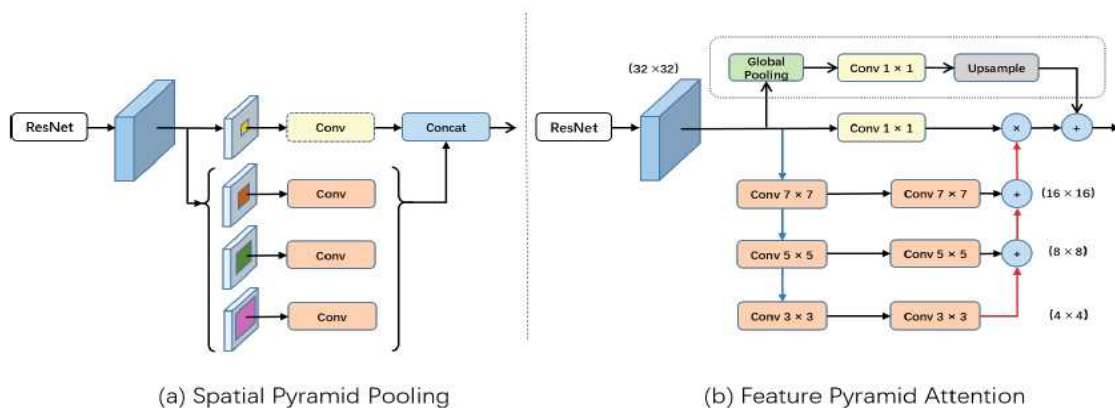


그림 11 <Feature Pyramid Attention>

- pyramid구조는 다른 scale의 feature 정보를 추출할 수 있고, pixel 수준에서 receptive field를 효과적으로 증가 가능

- pyramid attention module은 FPN(Feature Pyramid Network)와 같이 U모양의 구조를 실행함으로써 3개의 다른 피라미드 규모 하에 feature들을 통합
- high-level feature map의 해상도는 작기 때문에 큰 kernel size를 사용하는 것은 큰 연산 낭비를 가져오지 않음
- pyramid structure는 step-by-step으로 다른 scale의 정보들을 통합한다. 이로써 context feature들의 인접 scale들을 더 정확하게 통합할 수 있게 됨
- 또한 output feature에 global average pooling branch를 추가하였는데 이것은 FPA module의 성능을 더 향상시킴
- 결론적으로 FPA는 다른 scale의 context information을 통합하고, 그동안에 high-level feature map에 대한 더 나은 pixel-level attention을 생성함

③ Global Attention Upsample

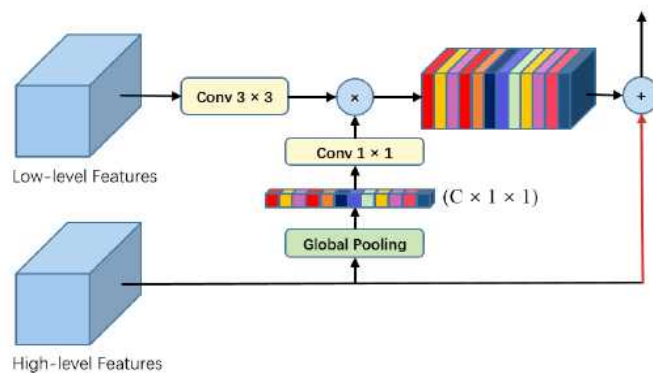


그림 12 <Global Attention Upsample>

- GAU module은 category localization detail들을 선택하기 위한 low-level feature의 guidance로써 global context를 제공하기 위해 global average pooling을 수행
- 이 모듈은 다른 크기의 feature map을 더 효과적으로 배포하고 간단한 방법으로 low-level feature map에 guidance information을 제공하는 high-level feature를 사용함

3. 모델 테스트

3.1. 테스트 결과 (Dice Score)

표 9 <모델별 테스트 결과 - Dice Score>

번호	Model name	Setting	epoch	batch	Learning rate	Optimizer	Number of data	val ratio	image size(rescale)	thickness	Validation Dice Score	Test Data Dice Score
1	PAN (using train cropped, val rescaled)	SET1	10	64	1.00E-04	RMSprop (weight_decay=1e-7)	10021	0.1	640x384	5	0.4827	0.3065
2	PAN (using train rescaled, val rescaled)	SET1	10	64	1.00E-04	RMSprop (weight_decay=1e-7)	10021	0.1	640x384	5	0.4945	0.4173
3	Unet (using train cropped, val rescaled)	SET1	9	8	1.00E-05	RMSprop (weight_decay=1e-6)	10021	0.1	640x384	5	0.4639	0.3198
4	Unet (using train rescaled, val rescaled)	SET1	10	8	1.00E-05	RMSprop (weight_decay=1e-6)	10021	0.1	640x384	5	0.6103	0.4678
5	Unet(50k train data)	SET1	9	8	1.00E-04	RMSprop (weight_decay=1e-8)	51461	0.1	640x384	5	0.5963	0.5068
6	Unet(10k train data)	SET3	5	2	1.00E-05	RMSprop (weight_decay=1e-8)	10000 (radoms ample)	0.1	640x360	5	0.4751	0.4077
7	Unet(10k train data)	SET3	5	2	1.00E-05	RMSprop (weight_decay=1e-8)	10000 (radoms ample)	0.1	640x360	10	0.4929	0.4444
8	Unet(50k train data)	SET3	5	2	1.00E-05	RMSprop (weight_decay=1e-8)	50000 (radoms ample)	0.1	640x360	10	0.5669	0.4676
9	Unet(200k train data)	SET3	5	2	1.00E-05	RMSprop (weight_decay=1e-8)	205640	0.1	640x360	5	0.5441	0.5189
10	Unet(210k train data)	SET2	5	6	1.00E-05	RMSprop (weight_decay=1e-8)	211650	0.05	640x360	5	0.5593	0.5363

- 테스트 결과, 동일 조건에서 U-Net이 PAN 모델에 비해 더 나은 Validation Dice score를 보임
- train data를 crop한 데이터를 학습시킨 모델보다 rescale을 적용시킨 모델이 더 나은 Validation Dice score를 보임
- 보통 10epoch 정도에서 loss가 더 이상 줄지 않고, 일정 범위의 validation dice score를 유지함
- 많은 데이터를 학습할수록 validation dice score가 더 높음.
- 따라서 최종적으로 Unet(210k train data) 모델을 사용하여 예측을 진행함.

3.2. 영상 Test 결과 (Visualization)

- Unet(210k train data) 모델을 사용하여 opencv를 통한 비디오 예측 시행
- opencv를 이용해 Video frame 단위 loop를 수행하고, loop 내부에서 Unet(210k train data) 모델을 불러와 frame 단위 predict를 진행
- 원본 데이터를 Input하면, Unet(210k train data) 모델의 Input에 맞게 640X360 해상도로 변경
- Predicted frame을 모아 out logic 수행
- Predicted frame의 Crack은 Green Channel로 변경하여 640X360으로 변경된 원본 이미지 Frame과 병합

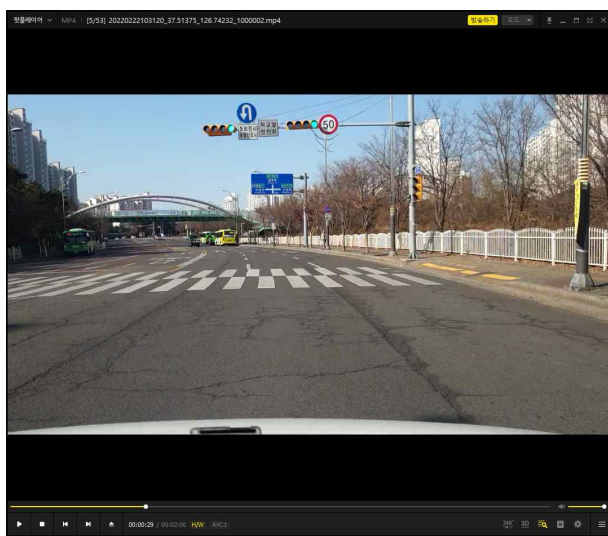


그림 13 <Test Input Video>

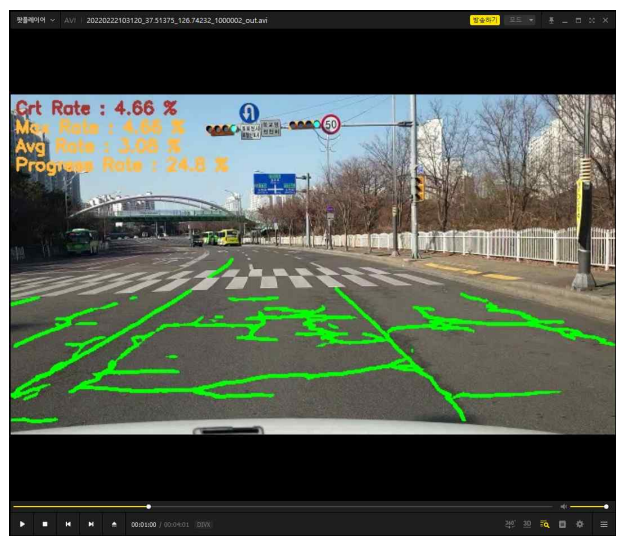


그림 14 <Predict Output Video>

- <그림 14>에서 보듯이 현재 Crack이 차지하는 픽셀 %, 최대 Crack 픽셀 %, 평균 Crack 픽셀 %, 현재 진행률을 오버레이 형태로 나타냄
- 모든 비디오를 전부 확인할 수는 없었지만, 전반적으로 Crack 인식이 준수한 수준으로 확인됨. 그러나 가로수, 신호등, 전깃줄 등 얇은 객체에서 생성되는 그림자를 Crack으로 인식하는 경우 또한 있었기 때문에 이에 대한 개선 사항 필요

4. 웹 서비스 구축

4.1. Home

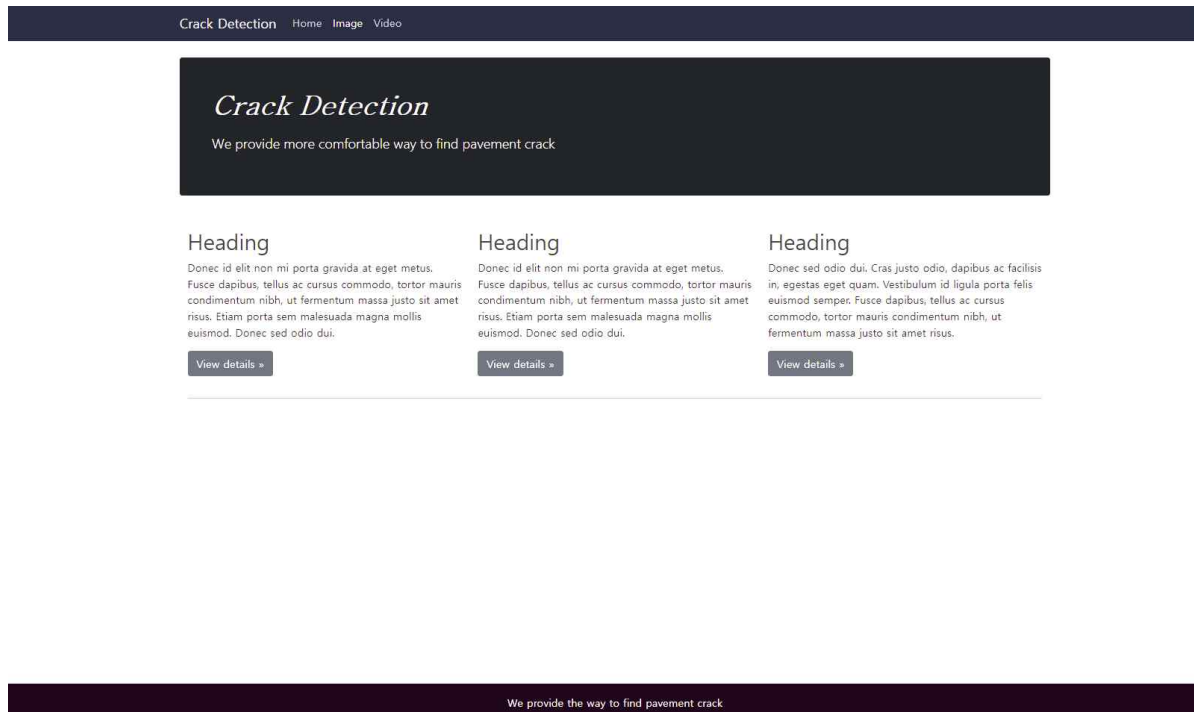


그림 15 <Homepage>

- 위 사진은 메인 홈페이지임. Image detection과 video inference 기능을 제공
- 현재는 필수적인 기능만 탑재하여 main homepage를 완전하게 구성하지 못하였으나, 추후 업데이트를 통해 다양한 기능 및 설명을 추가할 예정

4.2. Image

① Image selection

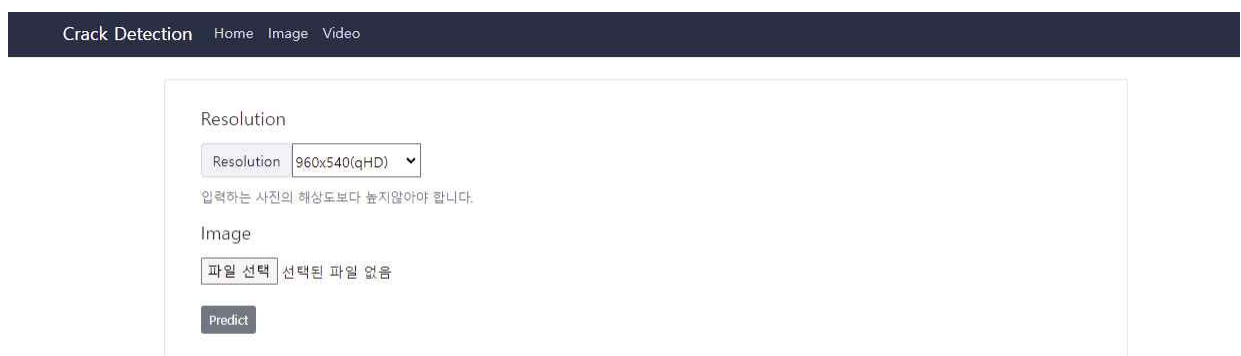


그림 16 <Image selection page>

- 페이지 상단의 'Image'버튼을 클릭하면, 위와 같은 페이지로 이동
- 우선, 예측한 결과 이미지의 해상도를 선택(단, 예측한 결과로 나오는 이미지는 원본 이미지의 해상도보다 클 수 없음)



그림 17 <Select image resolution>

- qHD(960x540), HD(1280x720), FHD(1920x1080) 이렇게 총 3개의 해상도를 선택 가능
- 해상도를 선택하고 나서, 예측하고자 하는 이미지를 선택(복수 선택 가능)
- 'predict' 버튼을 누르면 crack detection이 실행됨

② Image prediction result

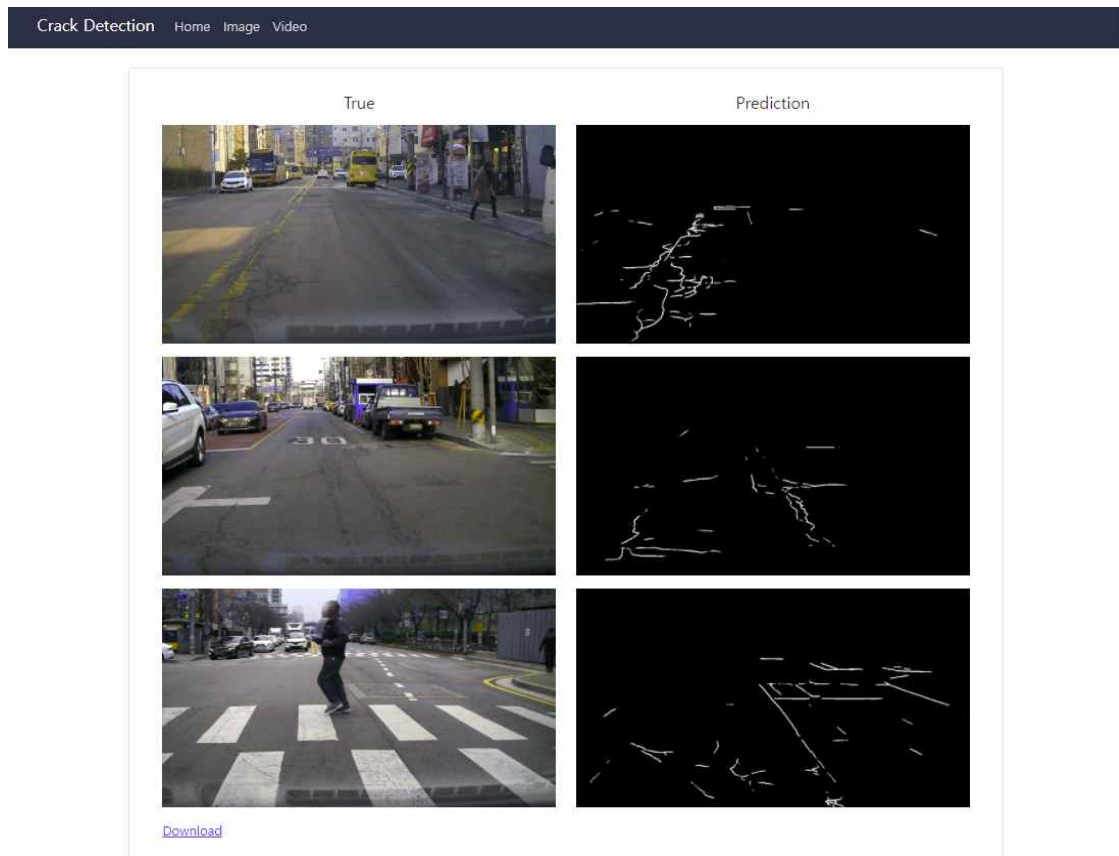


그림 18 <Result of crack prediction>

- 예측결과로써 위와 같은 페이지를 출력. 좌측은 사용자가 입력한 이미지목록이며 우측은 사용자가 입력한 이미지의 예측결과를 출력
- 이미지 결과 하단의 'Download'버튼을 클릭하면 위의 결과를 다운로드 받을 수 있음

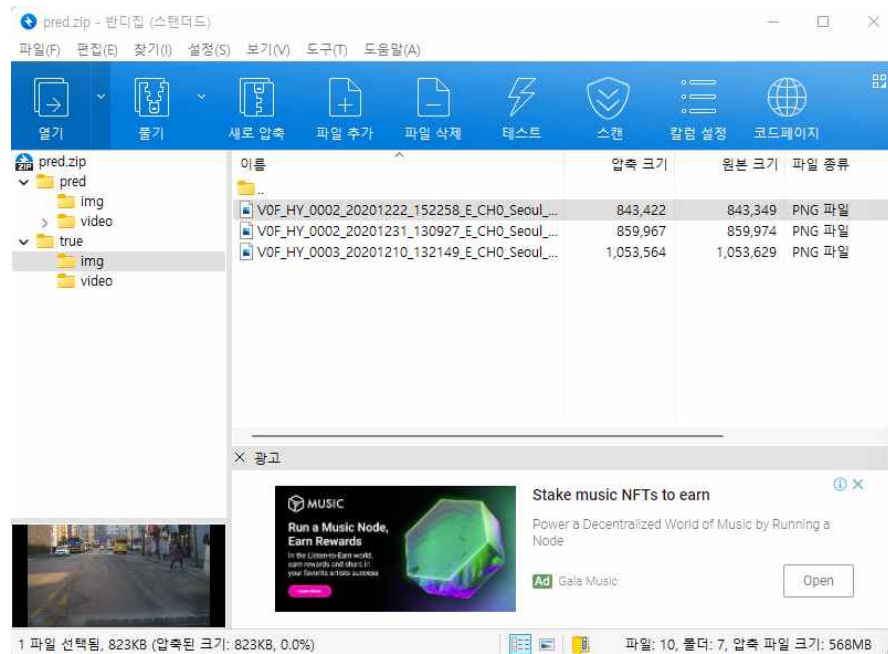


그림 19 <>

- 다운로드 결과는 위와 같음. true-img 폴더에는 사용자가 입력한 이미지가 담겨 있고, pred-img 폴더에는 사용자가 입력한 이미지의 예측한 결과가 담겨짐

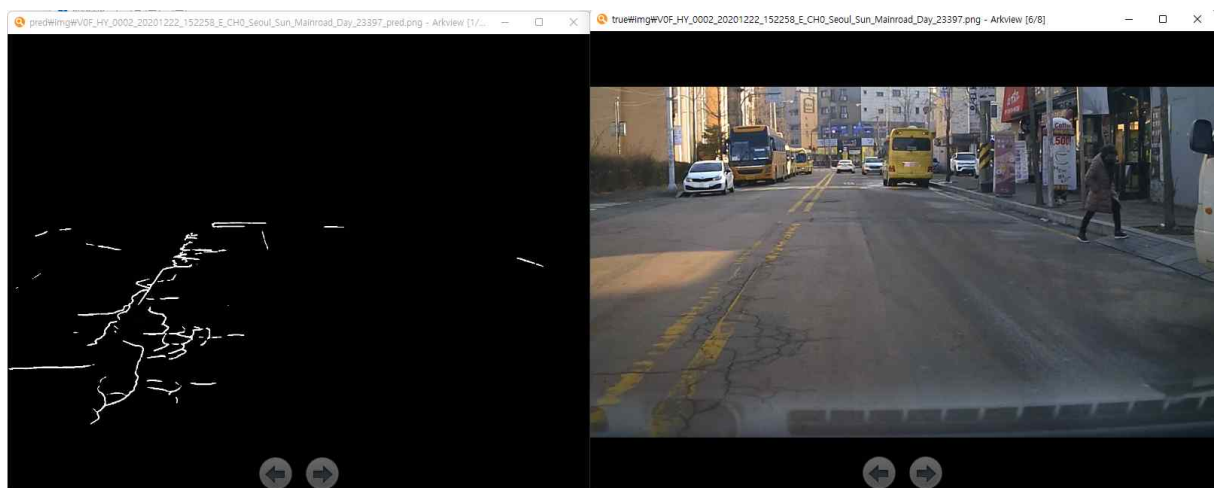


그림 20 <이미지 다운로드 결과>

- 다운 받은 이미지들을 실행해보면 위와 같이 원본, 예측 이미지를 비교 가능

4.3. Video

① Video selection



Crack Detection Home Image Video

Resolution
Resolution 960x540(qHD) ▼
입력하는 사진의 해상도보다 높지않아야 합니다.

Frame
Value: 15fps

Threshold
Value: 0.8%


Videos
파일 선택 선택된 파일 없음

Predict

그림 21 <비디오 설정>

- 페이지 상단의 'video' 버튼을 클릭하면 위와 같은 페이지로 이동함
- 우선 예측한 결과로 나오는 비디오의 해상도를 선택하고 차례로 frame, Threshold값을 선택함
- frame은 초당 몇 개의 frame을 사용할 것인지를 나타냄. 15fps~30fps 사이의 값을 가지며, 5단위로 선택할 수 있음
- 이미지에서 크랙이 차지하는 비율이 threshold를 넘어서면 해당 이미지를 저장하는 기능을 포함하고 있는데, 이 threshold는 이미지를 저장하기 위한 임계값을 의미한다. 0.5%~2.0% 사이의 값을 가지며 0.1단위로 선택할 수 있음
- 이후 crack detection을 실행하고자하는 video를 선택함(복수 선택 가능)

② Video result



Crack Detection Home Image Video

If you want to download predict video, click on Download button

[Download](#)

그림 22 <비디오 예측 결과>

- 비디오 예측이 완료되면 위와 같은 창이 보여짐. 결과는 'Download' 버튼을 눌러서 확인할 수 있음

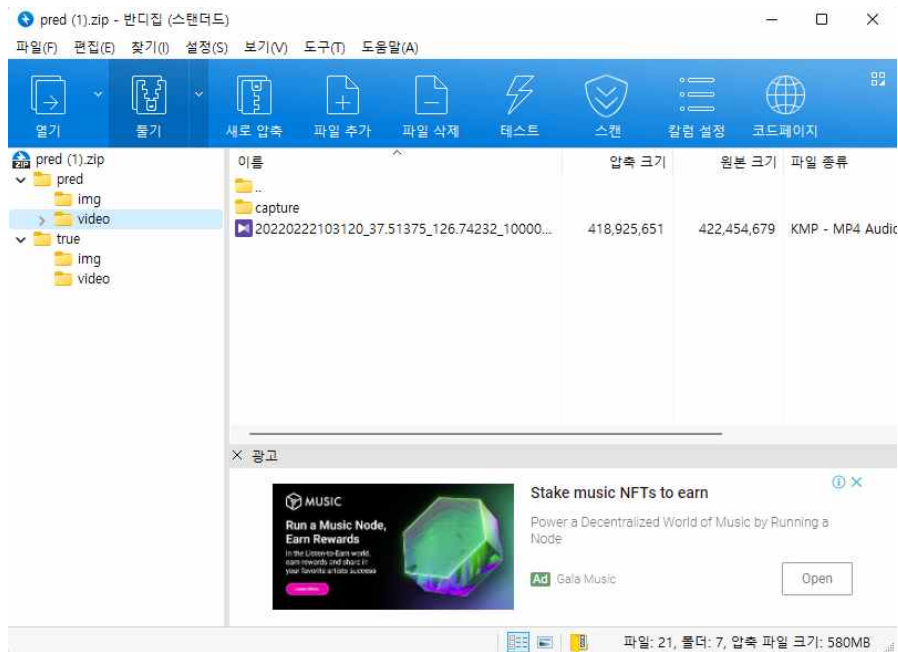


그림 23 <비디오 다운로드>

- 다운로드 버튼을 누르면 위와 같이 폴더를 얻을 수 있음
- true-video 폴더 안에는 사용자가 업로드한 원본 동영상에 담겨있고, pred-video 폴더 안에는 예측 결과 동영상과 threshold를 넘는 장면들의 이미지들이 담겨있음

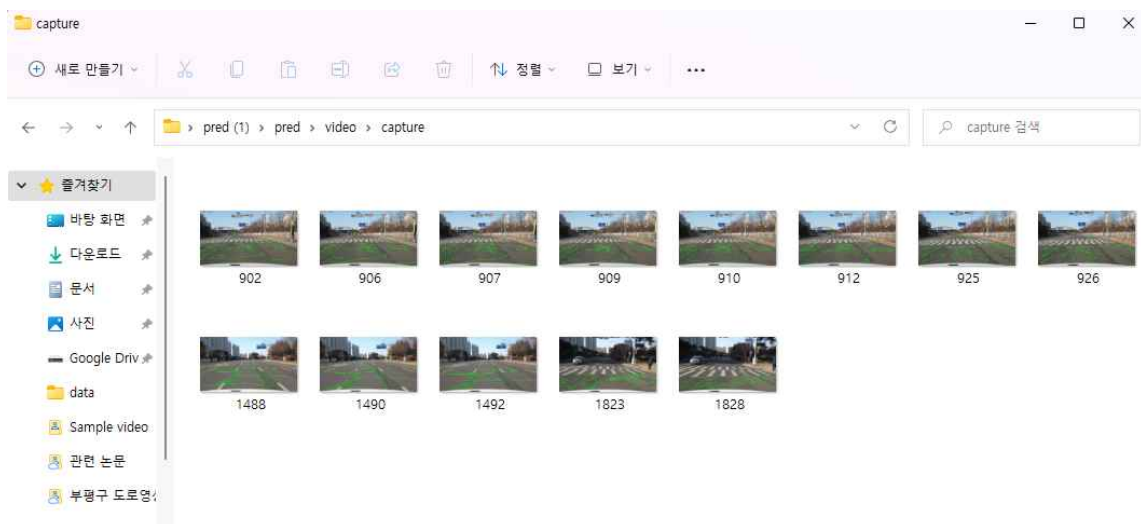


그림 24 <비디오 다운로드 캡처 결과>

- 위와 같이 threshold를 넘는 장면들의 이미지를 확인할 수 있음

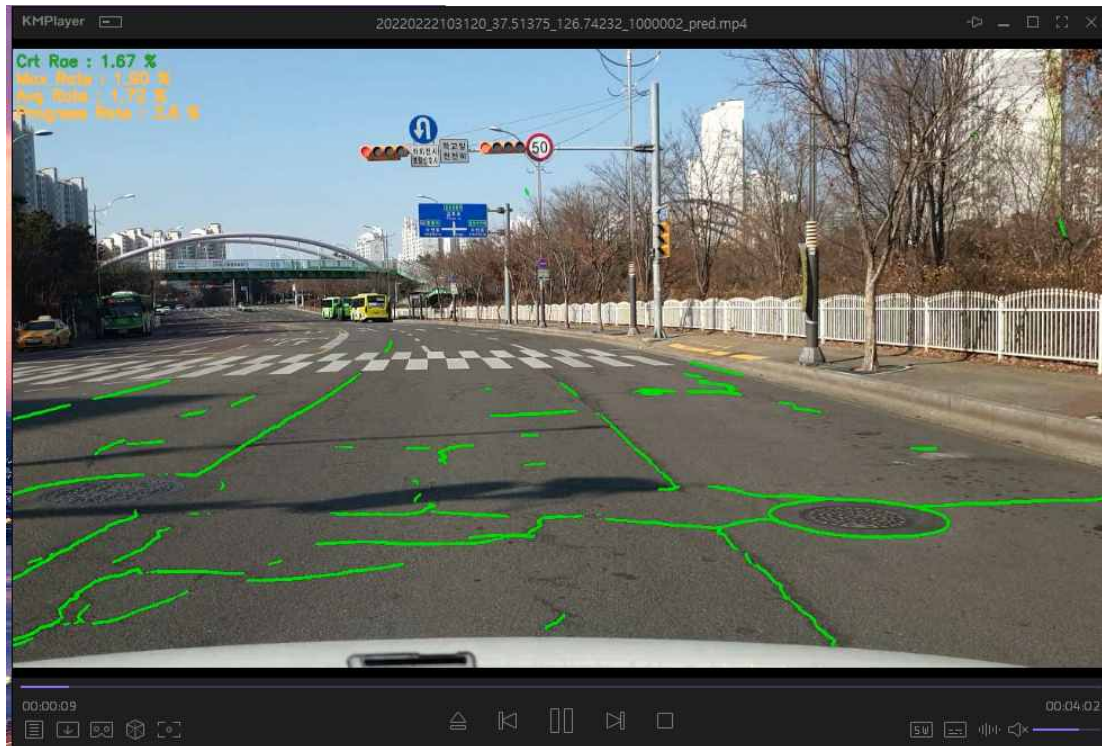


그림 25 <비디오 다운로드 결과 영상>

- 다운로드 된 동영상을 실행하면 위와 같이 예측한 결과 영상이 출력됨을 확인할 수 있음

V. 결론

1. 활용 방안

- 향후 행정구역 내의 도로 보수 지점을 선정하기 위해서 수치적인 데이터를 이용하여 객관적인 선정이 가능토록 함
- 예측된 내용들을 데이터베이스화하고 다른 데이터들과 결합하여(차량 통행량, 유동인구, 교통사고율 등) 도로 보수 지점의 우선순위를 결정할 수 있음
- 축적된 데이터들을 이용하여 어떤 곳에서 도로 균열이 많이 발생하는지 예측해 볼 수 있음

2. 한계점

- Training data의 마스킹이 제대로 되지 않은 데이터들이 많아서 오히려 model이 더 잘 예측했음에도 loss가 증가하는 현상이 발생함. 이로 인해 model의 성능이 떨어지는 결과가 발생하였음. 추후 더 나은 양질의 data로 학습을 진행하면 더 높은 성능을 보일 것으로 생각함.
- 현재 모델에서 crack을 detection할 때 어떤 종류의 crack인지는 분류하지 않음. 어떤 종류의 crack인지 분류모델을 만들어서 각각의 crack마다 구체적인 해결방안을 수립할 수 있도록 할 필요가 있음

- 모바일기기를 통해서 실시간으로 crack을 detection하려고 했지만, 예측에 너무 오랜 시간이 소요되어 빠른 속도로 예측이 불가하였음. 이는 모바일기기의 cpu로 예측을 진행하다보니 나타난 결과이고, 고사양의 서버, 빠른 통신을 이용해서 예측된 결과를 불러온다면 모바일기기에서도 실시간 예측이 가능할 것으로 생각함
- 웹서버에 예측을 수행하는 서비스를 탑재하였으나 현재의 서버(No GPU, RAM 1GB, SSD 255GB)로는 qHD 수준의 사진 한 장도 예측하는데 많은 시간이 소요됨. 따라서 웹사이트 배포를 위해서는 GPU가 탑재된 고사양의 서버를 필요로 함

VI. 참고 자료

① 학술지 논문

- 하종우 외. 2021. “딥러닝 기반의 분할과 객체탐지를 활용한 도로균열 탐지시스템 개발”, 「The Journal of Society for e-Business Studies」, 26(1), pp. 93-106.
- Li, H., Xiong, P., An, J., & Wang, L. (2018). Pyramid attention network for semantic segmentation. arXiv preprint arXiv:1805.10180.

② Reference Code

- [UNet For Sementic Segmentation](#)
- [Pyramid Attention Network](#)