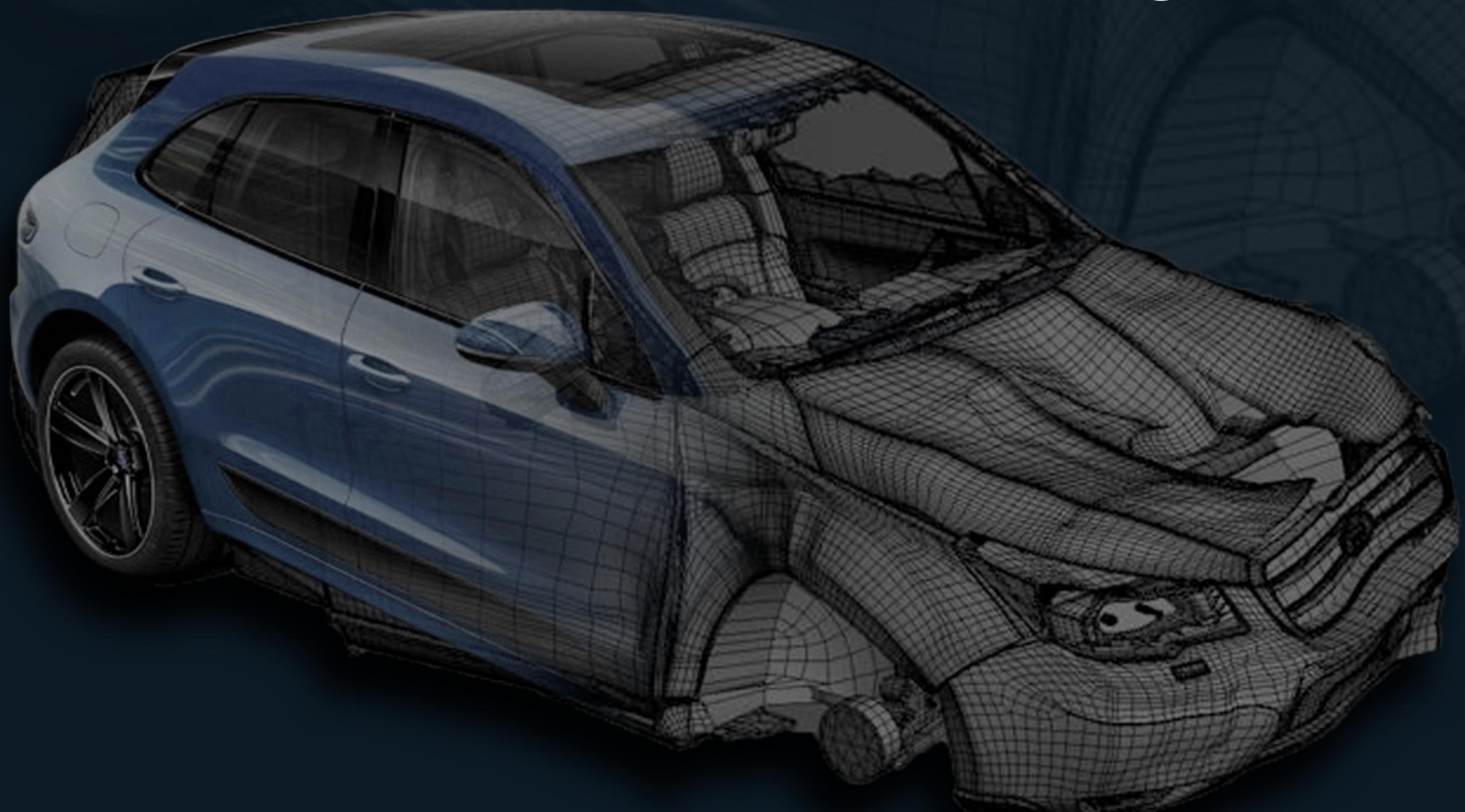


CAR DAMAGE DETECTION

Segmentation



INDEX

TIMELINE

CAR DAMAGE DETECTION _ SEGMENTATION

WHY SEGMENTATION?

WORKFLOW

AFTER PLAN

TIMELINE

WEEK 3

Reference Study

Deeplab Dissertation Search / Study

CAR DAMAGE DETECTION _ SEGMENTATION

WEEK 4

[BINARY]

Performance improvement of Reference
Dataset

- Dataset Search [Github & Kaggle]
- Augmentation

06

07

WEEK 2

[MULTI]

Error Point
Challenge Way about Error

WEEK 5

Model Search & Review [Dissertation Search]
Multi Labeling Dataset [Drawing VIA TOOL]



WHY SEGMENTATION?

WHY SEGMENTATION?

VIDEO : <https://player.vimeo.com/video/433749562?h=c759c8b246>

AI TECHNIQUE
WITH
CAR DAMAGE

美 보험사 솔레라, '사고차량 수리 견적은 AI로'

<http://www.aetimes.com/news/articleView.html?idxno=133261>

사고차량파손촬영 시수리비 A자동계산..."보상업무 획기적변화"

<http://www.seouln.com/news/articleView.html?idxno=388601>

'AI가 자동차 사고 수리비 분석'…금융위, 보험산업 '언택트' 활성화

http://news.tvchosun.com/site/data/html_dir/2020/07/09/2020070990082.html

[카&테크]자동차 파손·결함 잡아내는 비전 AI 기술

<https://auto.v.daum.net/v/20211014150136811>

WHY SEGMENTATION?

SOCAR ‘ACCIDA’

엑시다는 고객이 입력한 이미지가 차량의 어떤 부위를 촬영한 것인지 인지하고, 이미지 내 **픽셀 단위로 분석**해 파손 여부를 판단한다.

자동차 제조뿐 아니라 차량공유(카셰어링) 분야에서도 비전 AI를 활용한다.

국내 1위 카셰어링 서비스 쏘카는 AI 머신러닝을 기반으로 차량의 파손 여부를 감지한다. 고객이 차량 대여 전 앱을 통해 전송한 외관 사진을 차량 이미지 파손여부 자동 탐지 모델 '엑시다'로 분석한다.

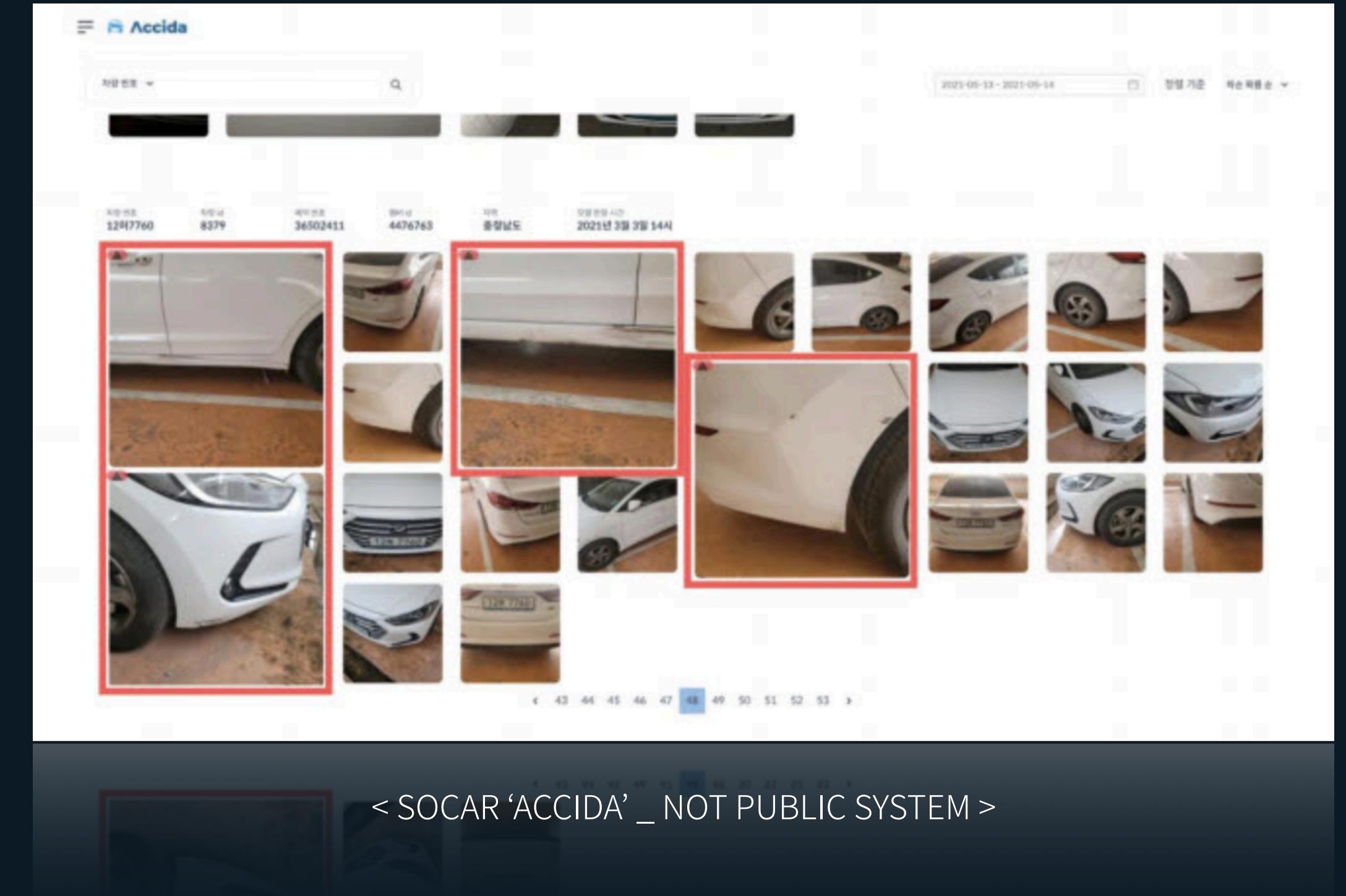
엑시다는 고객이 입력한 이미지가 차량의 어떤 부위를 촬영한 것인지 인지하고, 이미지 내 픽셀 단위로 분석해 파손 여부를 판단한다.

AI에 모든 결정을 맡기진 않는다. 엑시다가 사고 발생 가능성이 높은 이미지를 선별하면, 쏘카 운영본부 직원들이 추가 검수하는 방식이다. 직원들은 기존 수만장의 사진을 검수해야 했지만 엑시다 도입 이후 하루 300건가량만 확인하면 된다.

쏘카는 엑시다를 통해 사고 발생 시점을 명확히 파악해 책임소지를 명확히 하고, 고객 안전에 위해가 될 수 있는 요소들도 검출해 사고를 예방할 수 있다고 설명했다.

위현종 쏘카 최고전략책임자(CSO)는 “엑시다는 차량 파손 시점을 정확히 파악하고 수리 필요 여부를 판단하는 데 큰 도움을 준다”며 “향후 수만대의 자율주행 로보택시를 관리하는 데도 유용하게 사용할 수 있을 것”이라고 말했다.

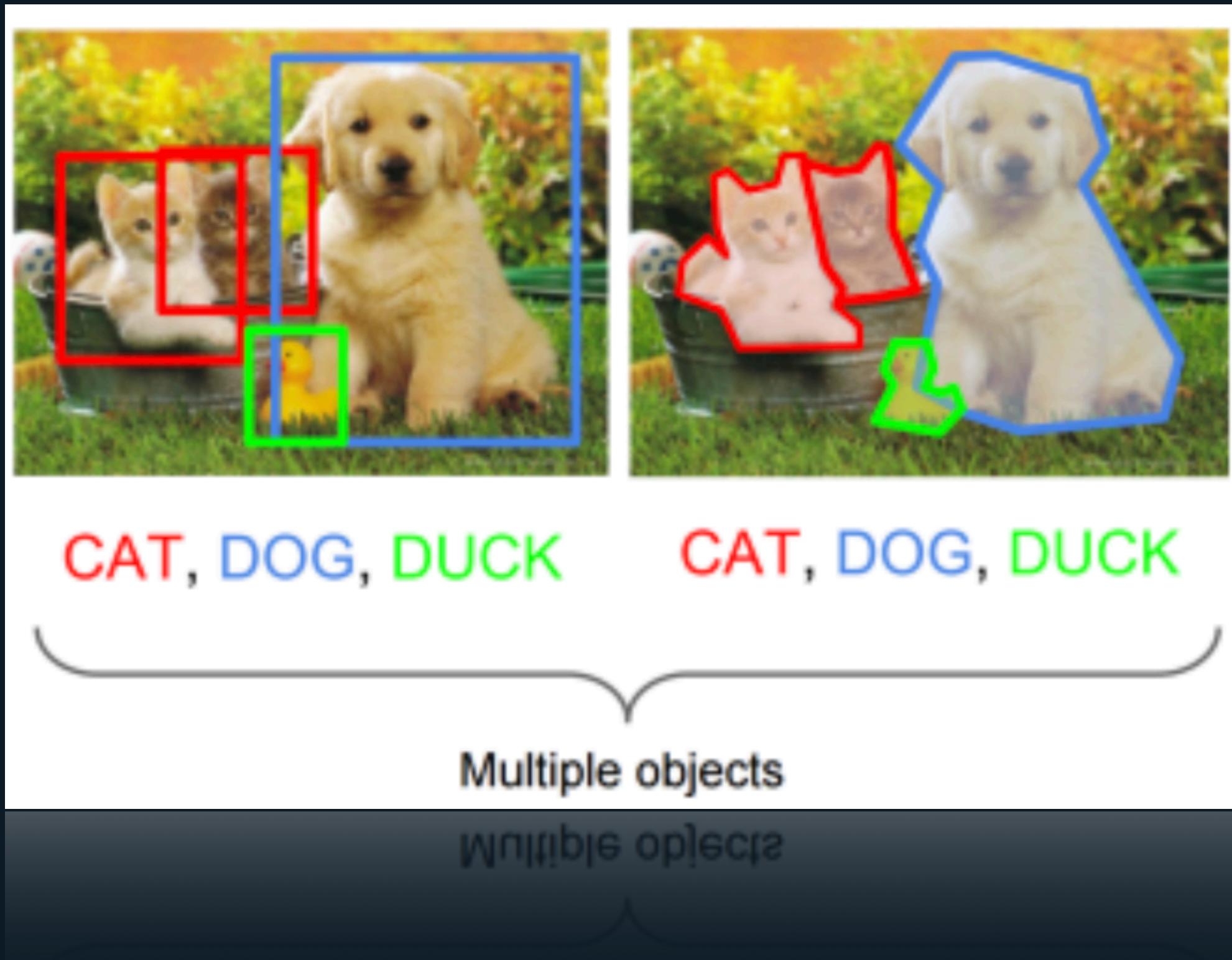
특히 차량의 손상 여부를 정확히 파악하는 것은 차량 공유 서비스에서 매우 중요한 역할을 한다. 차량 손상 여부를 정확히 파악하는 것은 차량의 가치를 보호하는 데 있어 필수적인 요소다.



< SOCAR ‘ACCIDA’ _ NOT PUBLIC SYSTEM >

WHY SEGMENTATION?

◆ Detection | Segmentation



Detection

사진 속 여러 객체를 각 구별하여 Detection

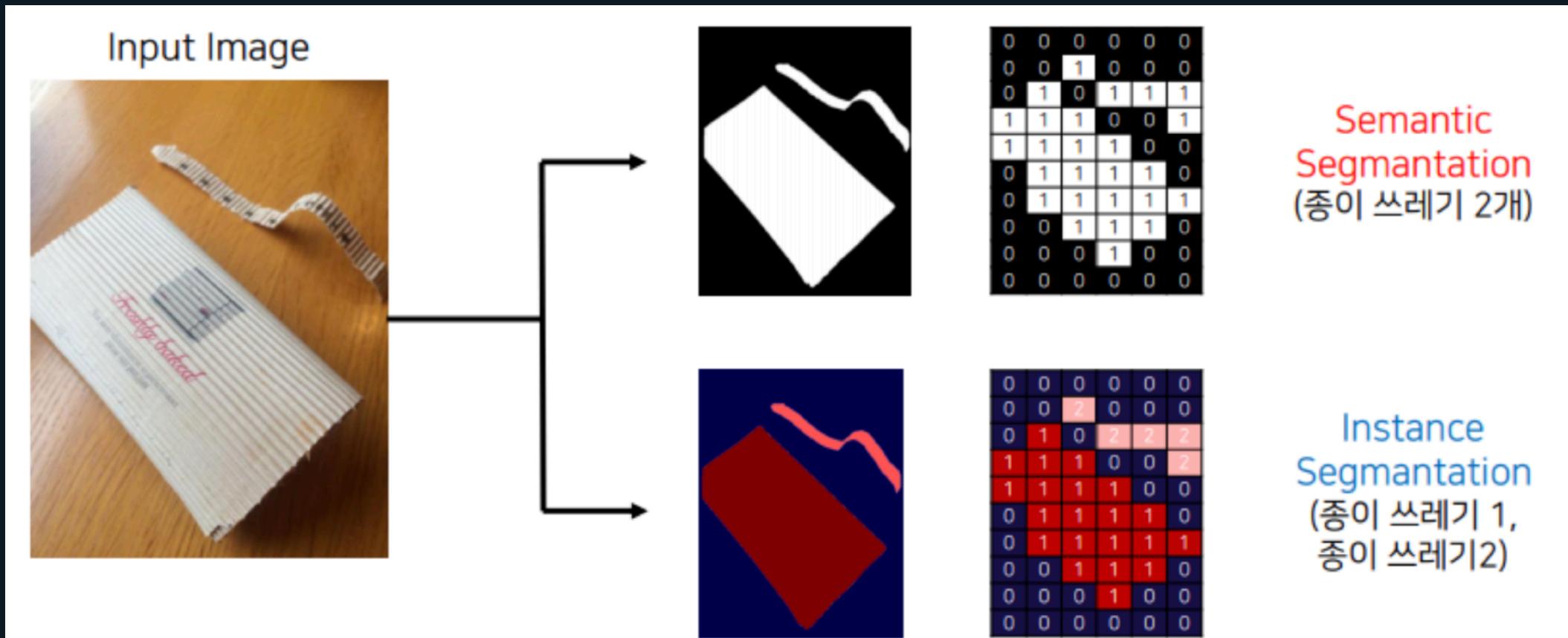
Segmentation

사진 속 객체를 box가 아닌 정확한 영역으로 표시
(Detection 보다 세부적)

WHY SEGMENTATION?

◆ Segmentation

객체탐지 [자율주행 / 의료분야]에서 주로 사용



Instance Segmentation

같은 종류라도 다른 객체이면 다른 클래스로 취급

Semantic Segmentation

같은 종류의 객체는 하나로 취급

- FCN
- U-Net
- DeepLab

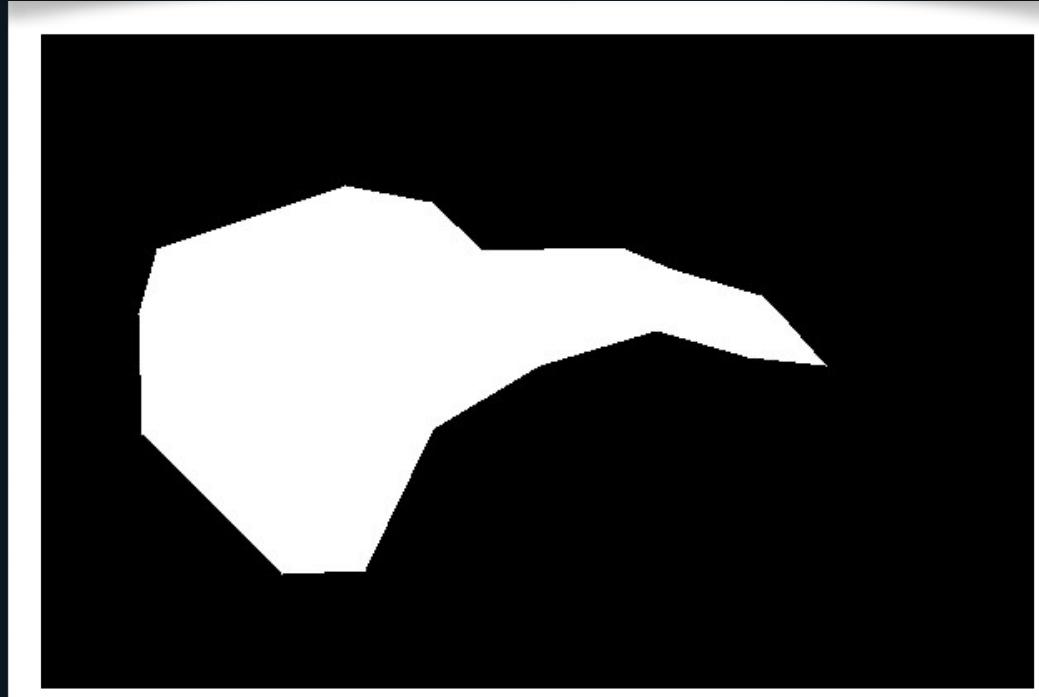
REFERENCE: Semantic Segmentation 사용 [scratch 유 / 무에 대한 데이터만 존재했기 때문이라고 판단]

WORK FLOW



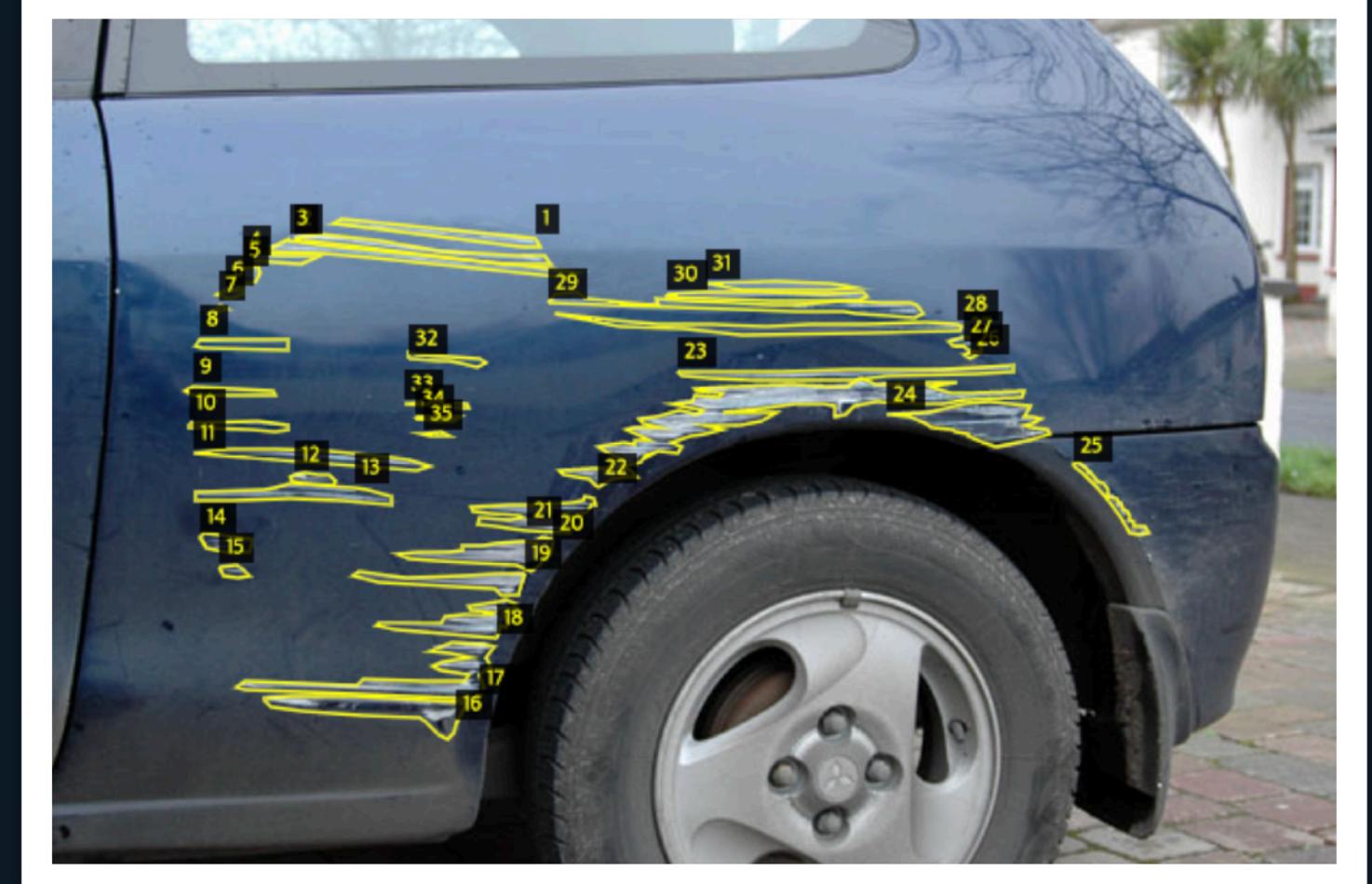


DATASET



Why Making Dataset?

- 작업 시간 : 1일 / 총 400장 작업
- 기존 데이터 셋에서 스크래치와 같은 예시를 보았을 때,
손상이없는 부위까지 영역이 잡힌 것을 알 수 있었다.
- 좀 더 나은 성능을 기대하며 직접 수작업을 통하여 세밀한
영역 선정을 하여 데이터 셋을 구축하고자 하였다.

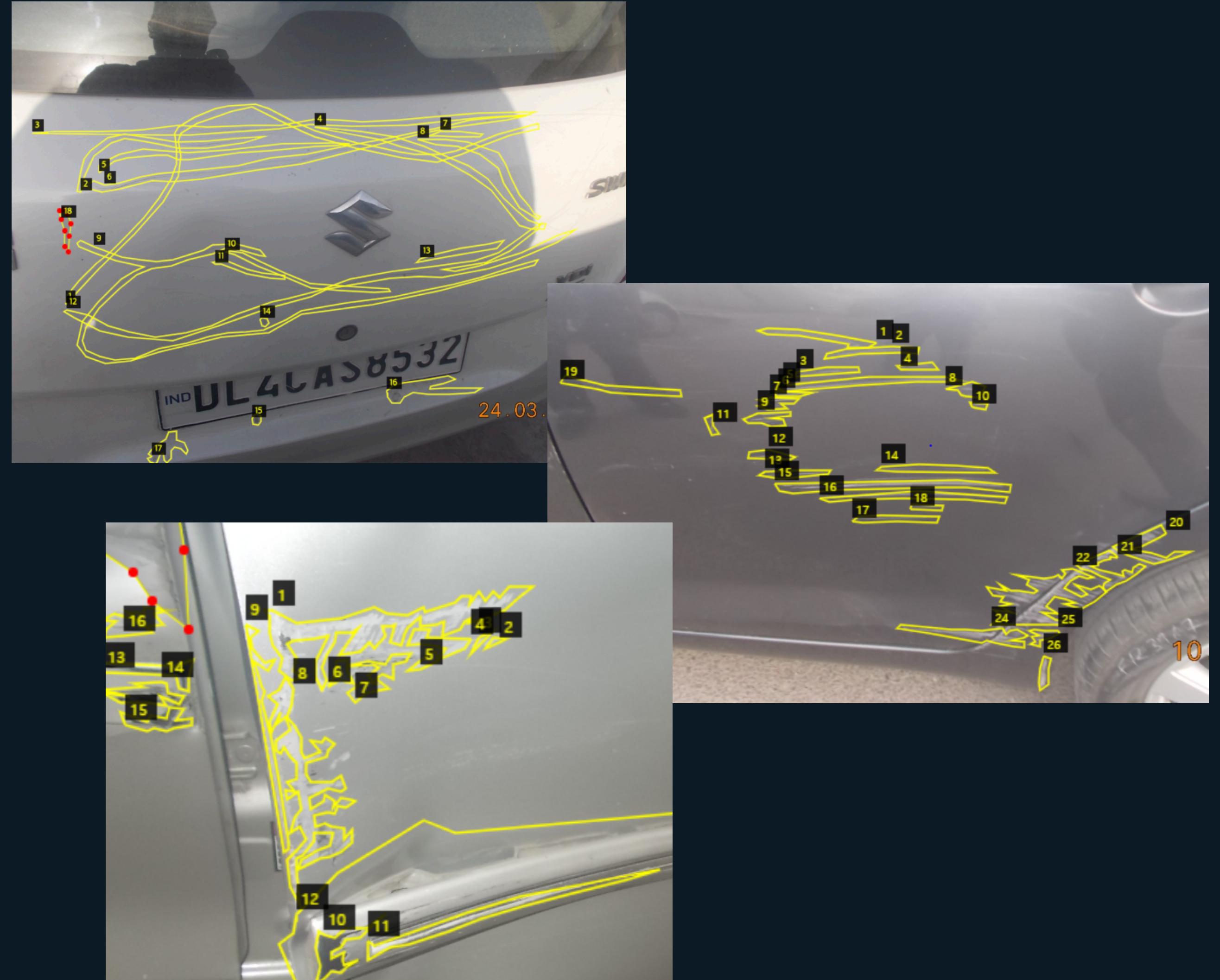
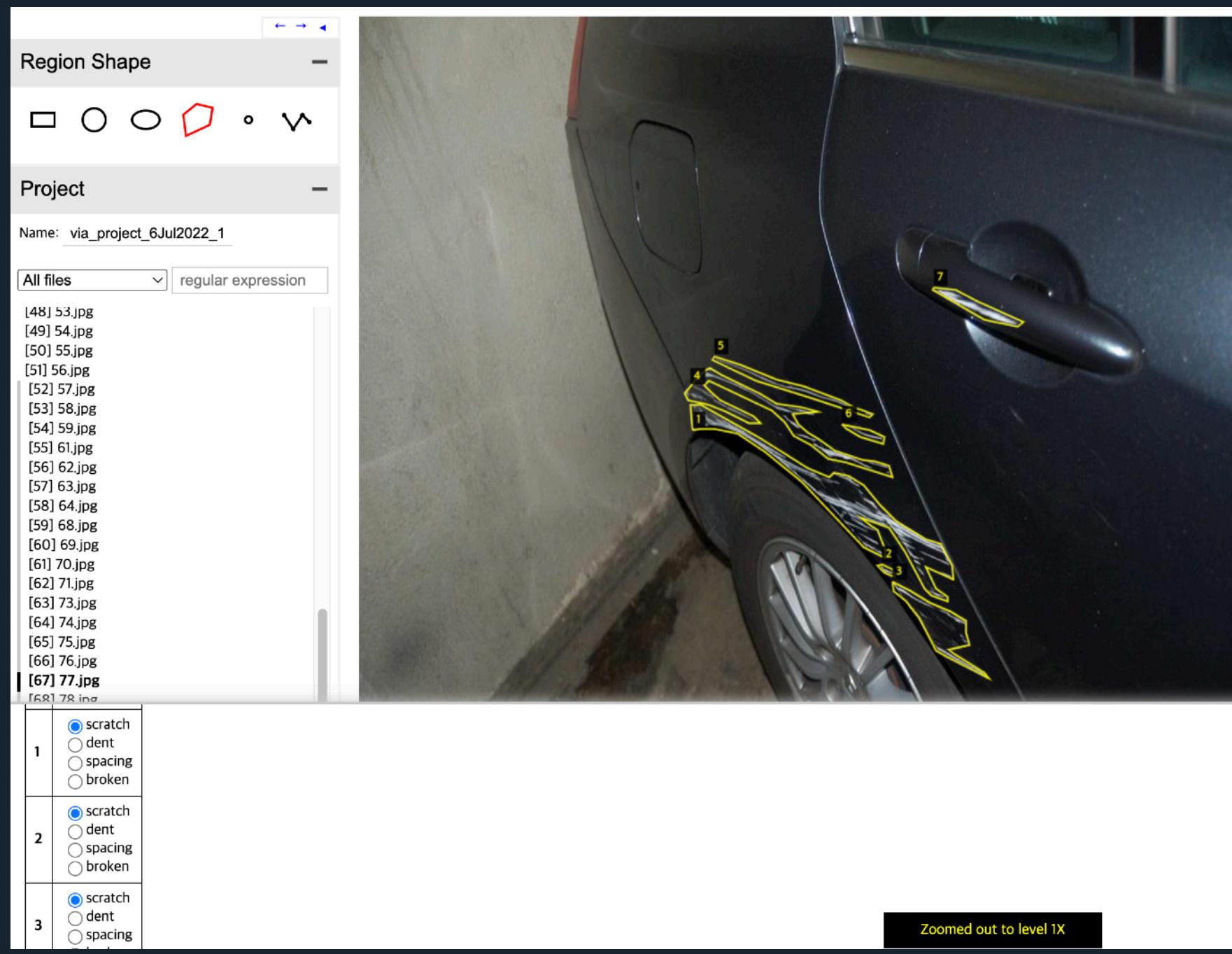


◆ MAIN DATASET [VIA TOOL]

DATASET

Multi Label Class Setting

CLASS ID | MAKIMG DATASET





PRE-PROCESSING

PRE-PROCESING

LACK OF DATASET

AUGMENTATION

MAKING DATASET
ABOUT MULTI LABEL

CLASS ID (json to image)

PRE-PROCESING

◆ ALBUMENTATION

HorizontalFlip

VerticalFlip

Grayscale

Blur

OpticalDistortion

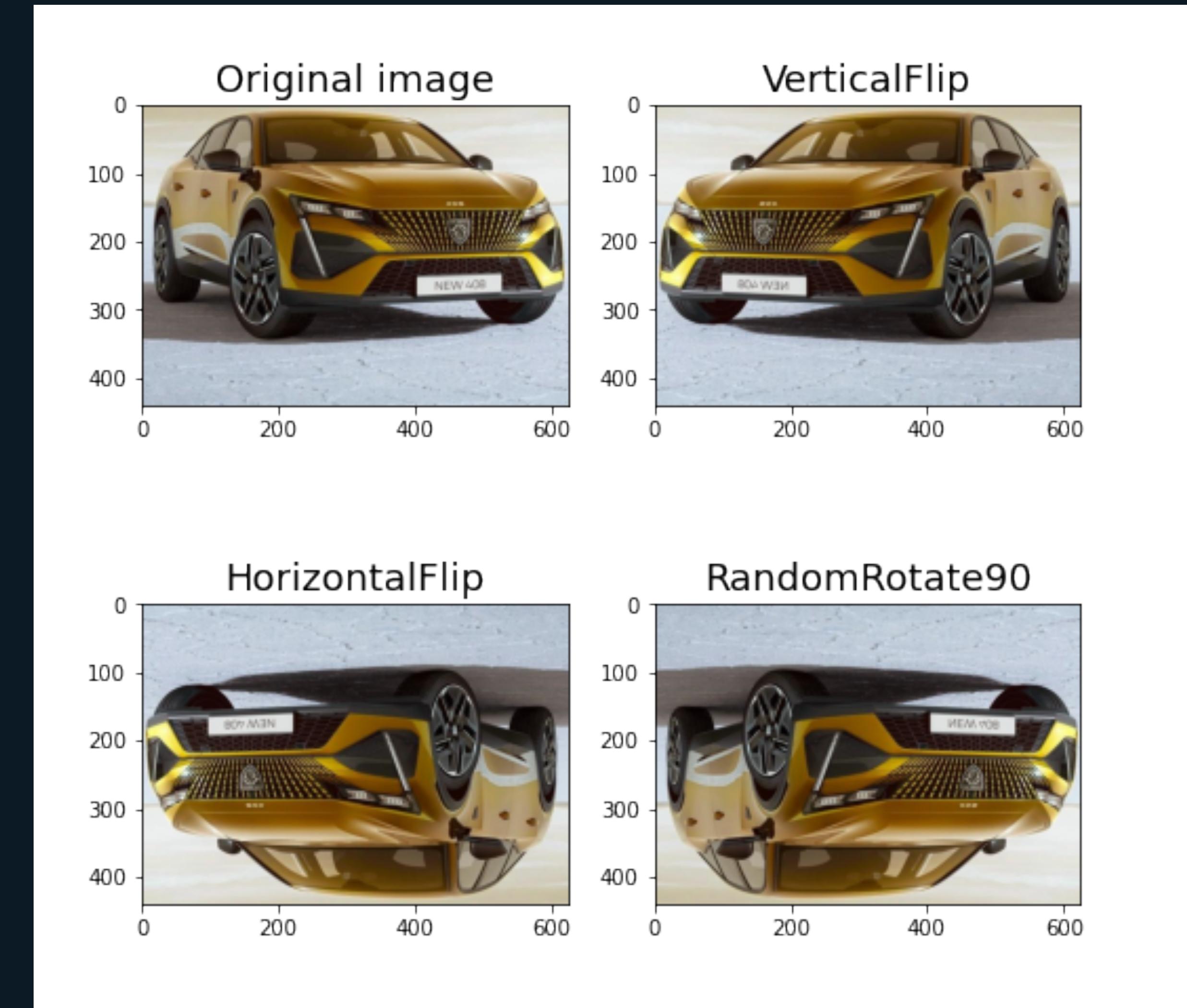
Resize

RandomRotate90

Transpose

COMPOSE

AUGMENTATION | Albumentation



◆ JSON TO IMAGE

PRE-PROCESING

```
"image67.jpg47277": {
    "fileref":"",
    "size":47277,
    "filename":"image67.jpg",
    "base64_img_data":"",
    "file_attributes":{},
    },
    "regions":{},
    "0": {
        "shape_attributes": {
            "name": "polygon",
            "all_points_x": [
                298,
                327,
                346,
                369,
                522,
                512,
                488,
                393,
                307,
                288,
                298
            ],
            "all_points_y": [
                199,
                203,
                199,
                203,
                199,
                203,
                199,
                203,
                199,
                203,
                199
            ]
        }
    }
}
```

```
"0002.JPG7164": {
    "filename": "0002.JPG",
    "size": 7164,
    "regions": [
        {
            "shape_attributes": {
                "name": "polygon",
                "all_points_x": [
                    216,
                    188,
                    155,
                    53,
                    15,
                    20,
                    72,
                    159,
                    188,
                    215,
                    220
                ],
                "all_points_y": [
                    24,
                    51,
                    51,
                    64,
                    67
                ]
            }
        }
    ]
}
```

TYPE_1 [DICT]

TYPE_2 [LIST]

CLASS ID | JSON TO IMAGE

TYPE_1

Regions 의 형식이 DICT 형식

TYPE_2

Regions 의 형식이 LIST 형식



MODELING

◆ Deeplab_V3 ResNet

SOCAR REFERENCE

BACKBORN



PRE-TRAINING MODEL

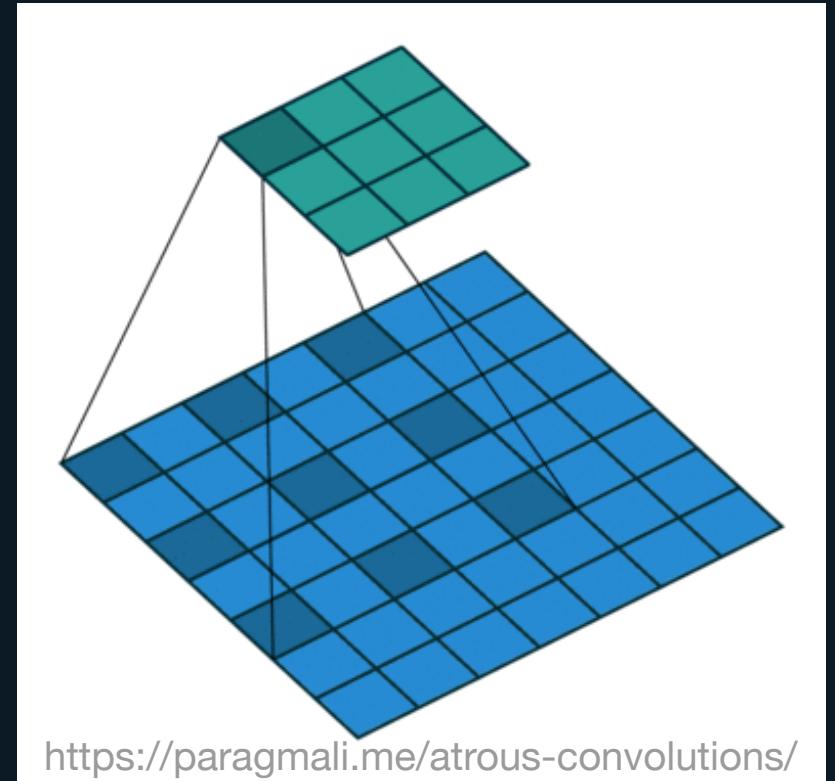
- Pascal VOC DATASET [20 Category]
- COCO train2017의 일부분 데이터 셋에 대해 훈련

ASPP

다른 dilation rate를 가진 Atrous Convolution layer를
중첩하여 multi scale에 더 잘 반응할 수 있도록
피라미드 형식으로 쌓은 방식

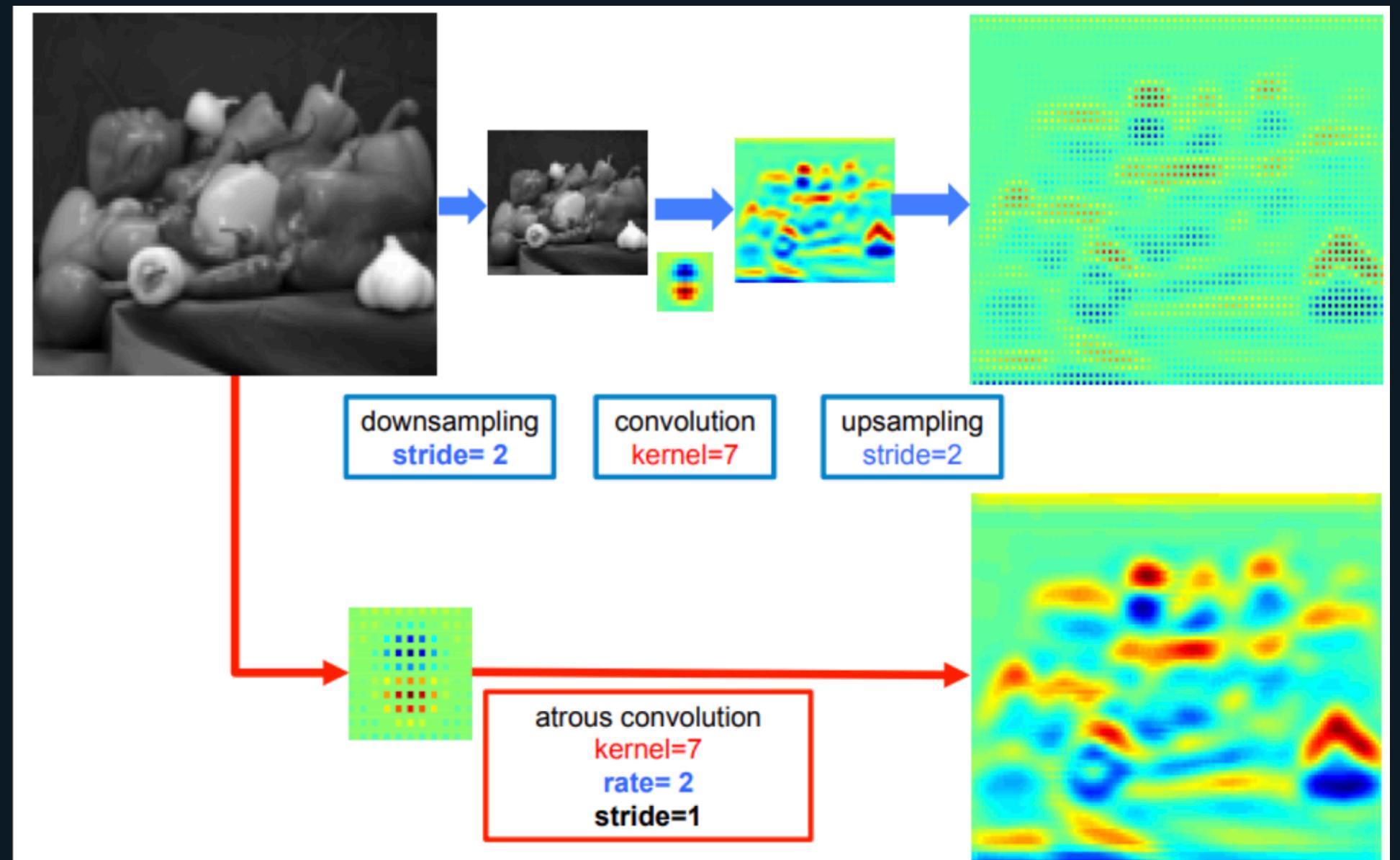
REFERENCE MODEL | DEEPLAB_V3 [ASPP]

Atrous Convolution Layer



<https://paragmali.me/atrous-convolutions/>

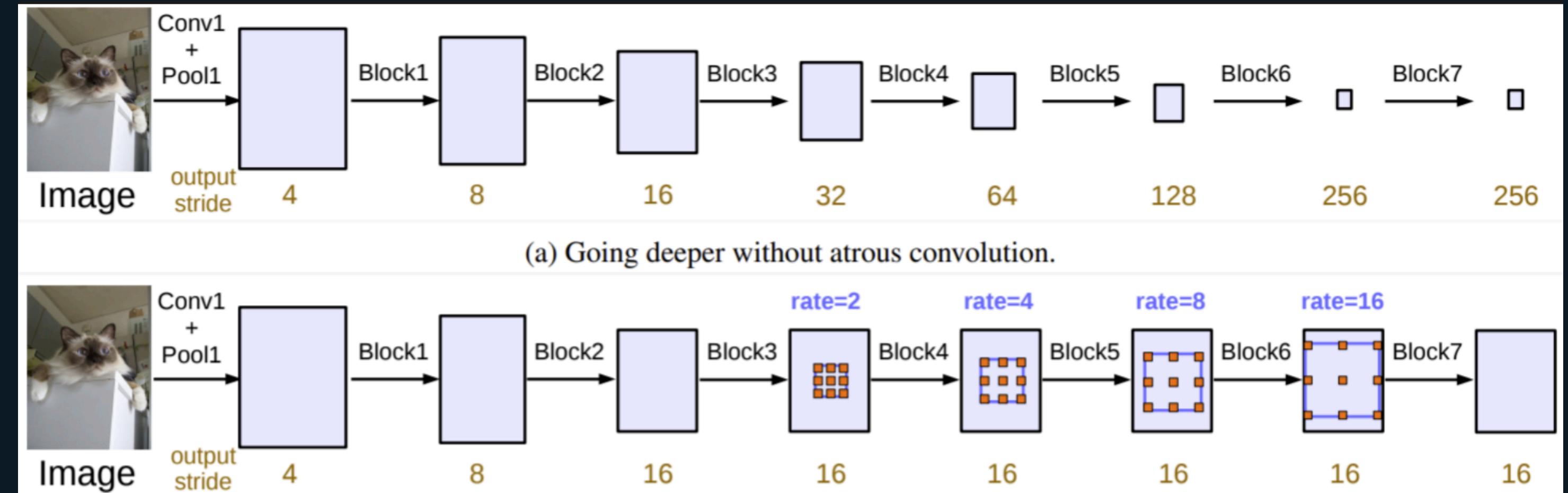
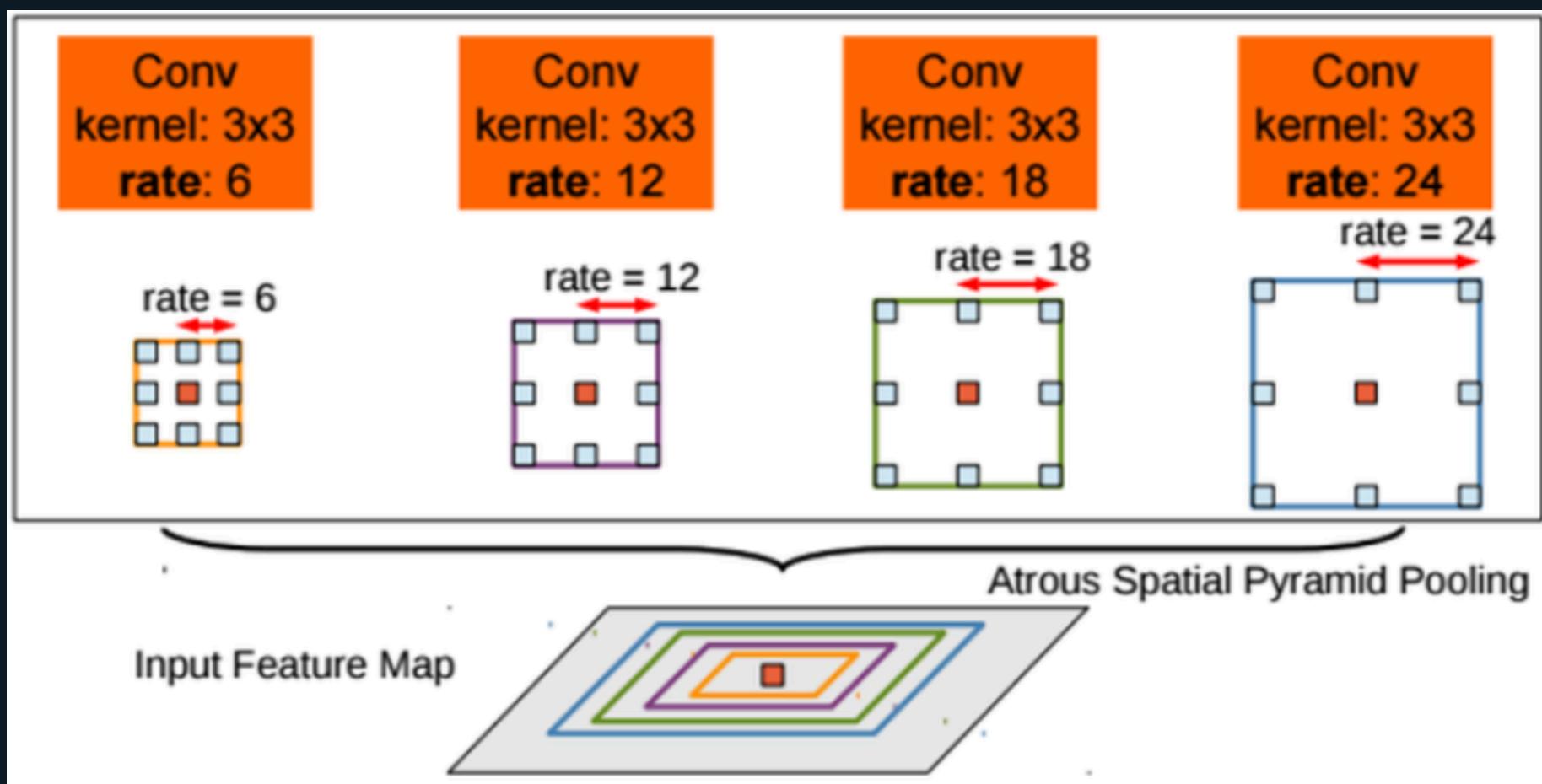
ex . Atrous Convolution Layer



SOCAR REFERENCE

REFERENCE MODEL | DEEPLAB_V3 [ASPP]

◆ ASPP (Atrous Spatial Pyramid Pooling)



ASPP 의 구조

ASPP 비교 [General CONV. Layer | Atrous CONV. Layer]

◆ PROCESS

- DeeplabV3 모델들을 이용하여 성능개선을 시도
[Reference DataSet 만 사용]

USED IN REFERENCE : ResNet - 50

TRY MODEL : ResNet - 101

REFERENCE DATA ONLY [USING ResNet-50]	
Model structure	Mean IOU
deeplabv3_resnet50	66.4
deeplabv3_resnet101	67.4

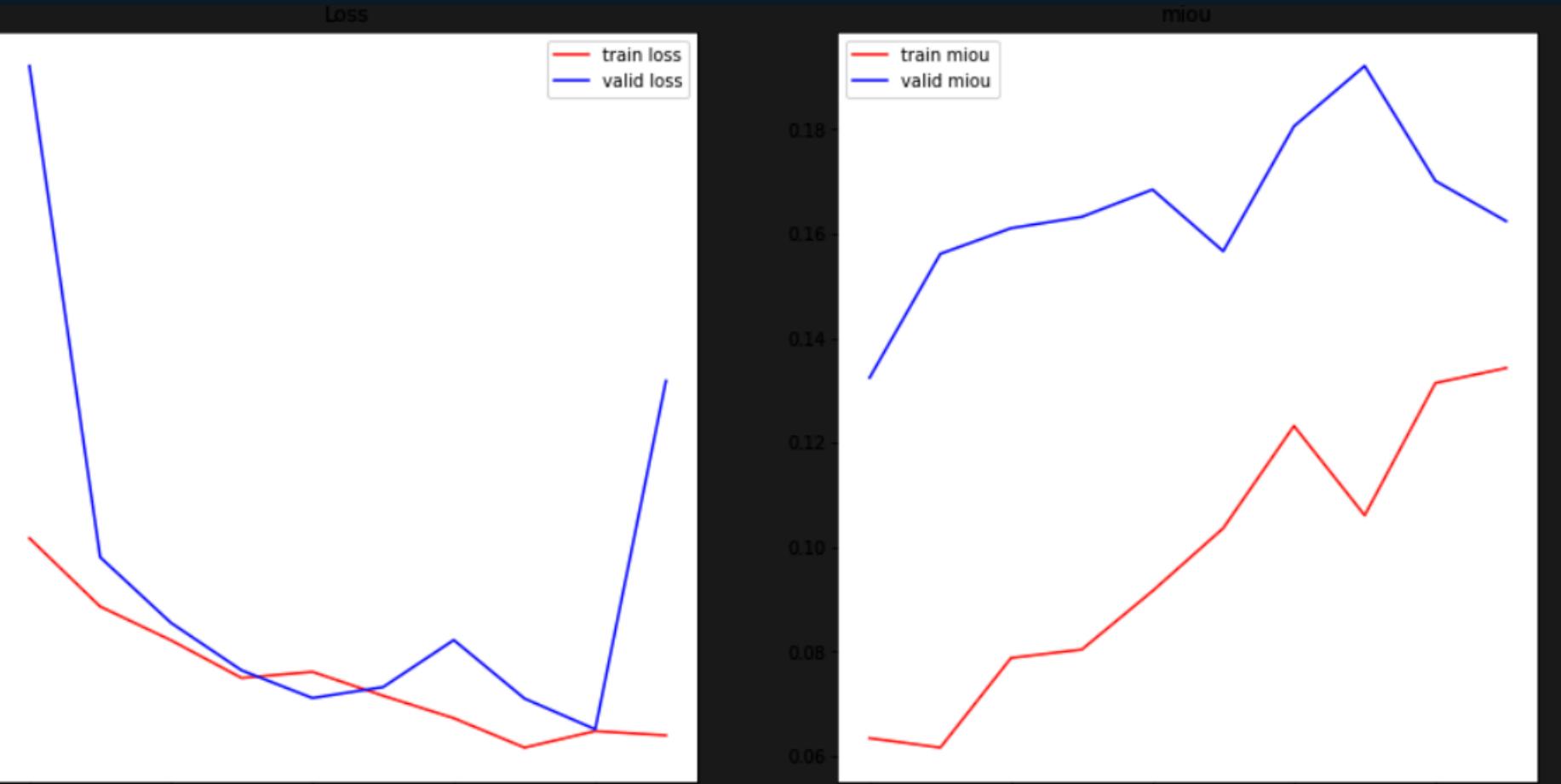
성능 개선 시도

- Augmentation
- Hyper parameter 변경

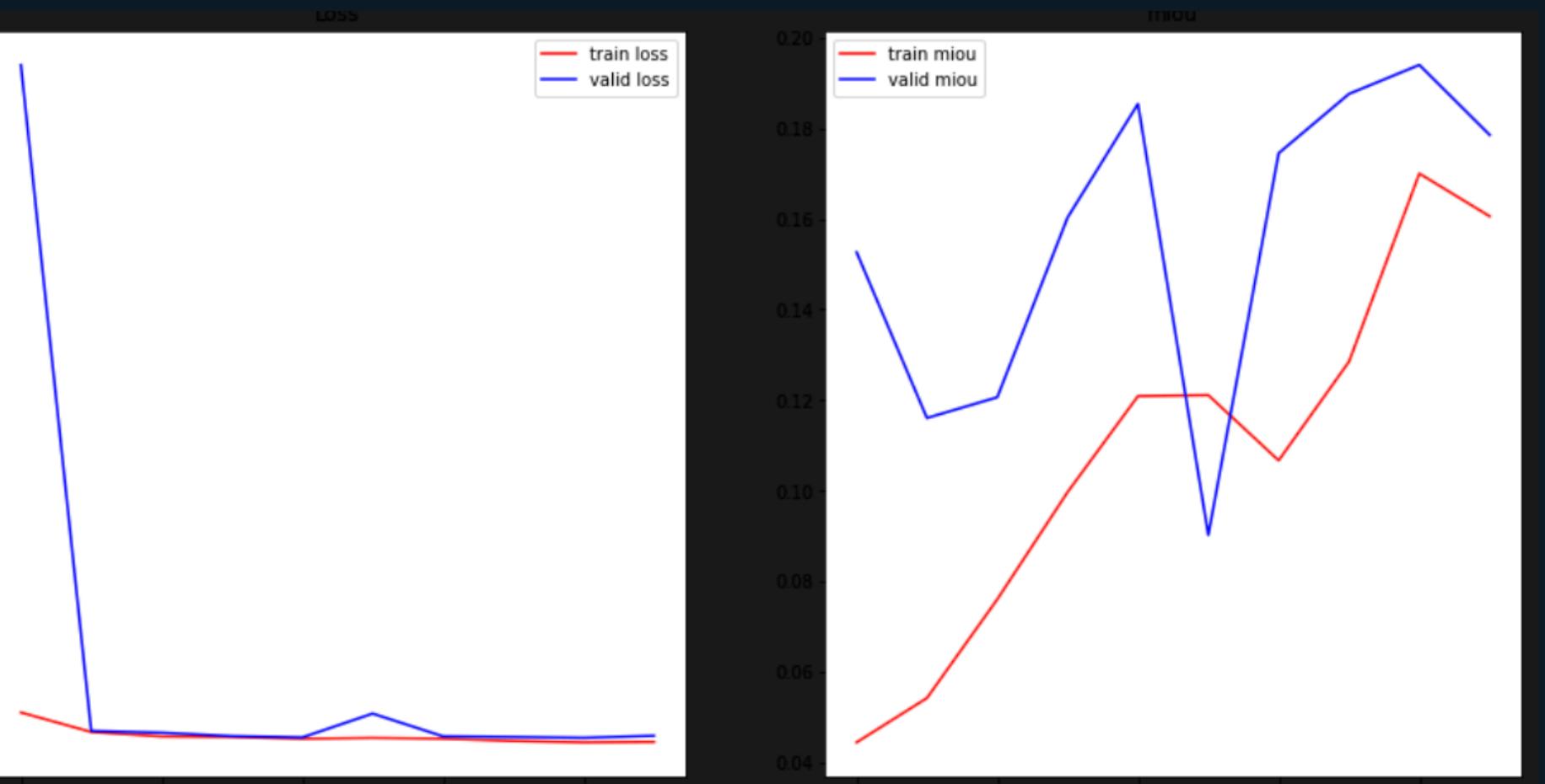
SOCAR REFERENCE

CHALLENGE POINT | 래퍼런스 성능 향상 시도

DeeplabV3_ResNet-50



DeeplabV3_ResNet-101





RESULT

BINARY TO MULTI

MIoU SCORE를 어떻게 해야 높일 수 있을까?

IDEA

ADDING
ORIGINAL DATASET

TEST _ 1

ADDING
NON DAMAGE DATASET

TEST _ 2

TRINING
SCRATCH IMAGES ONLY

◆ TRY TRAINING

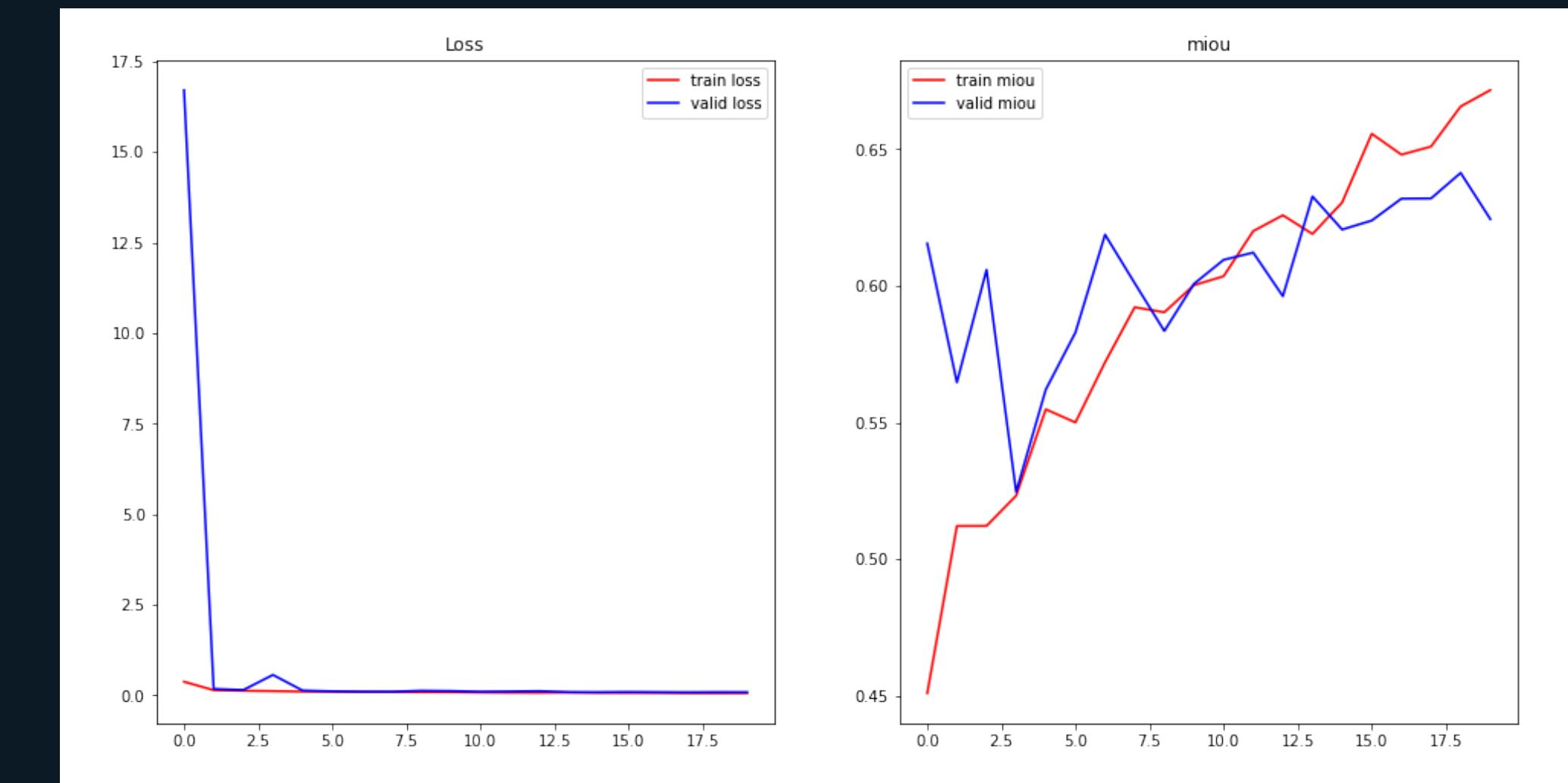
IDEA

GRAPH 1 (LEFT)

Using only reference + Modify hyper-parameter

Epoch : 20

iou_back	0.970279
iou_scratch	0.278590
train loss	0.059257
valid loss	0.086989
train miou	0.671645
valid miou	0.624434

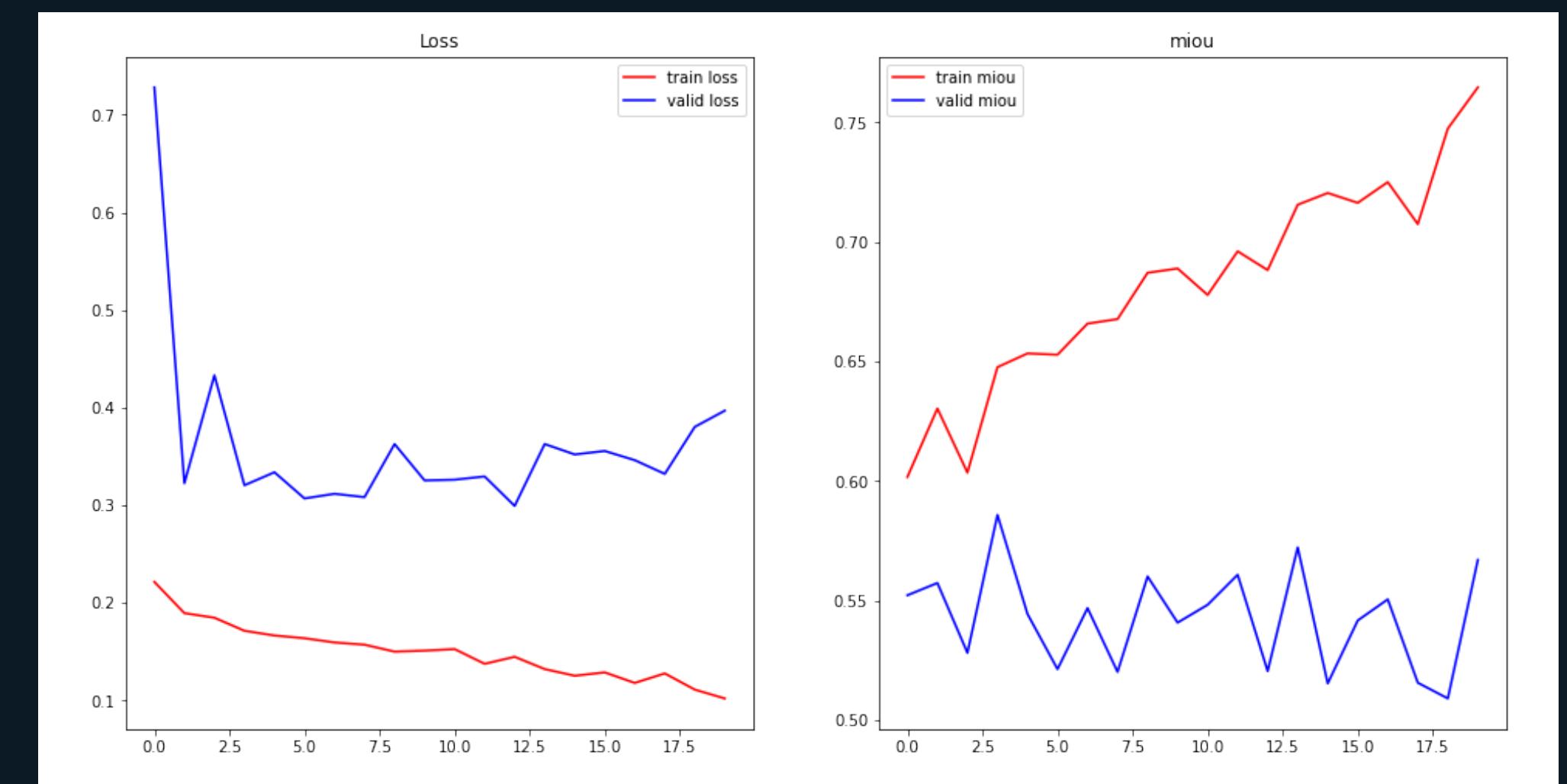


GRAPH 2 (RIGHT)

Adding Damage Data + Augmentation

Epoch : 20

iou_back	0.890323
iou_scratch	0.243610
train loss	0.101921
valid loss	0.396822
train miou	0.764633
valid miou	0.566967



ADDING ORIGINAL DATASET

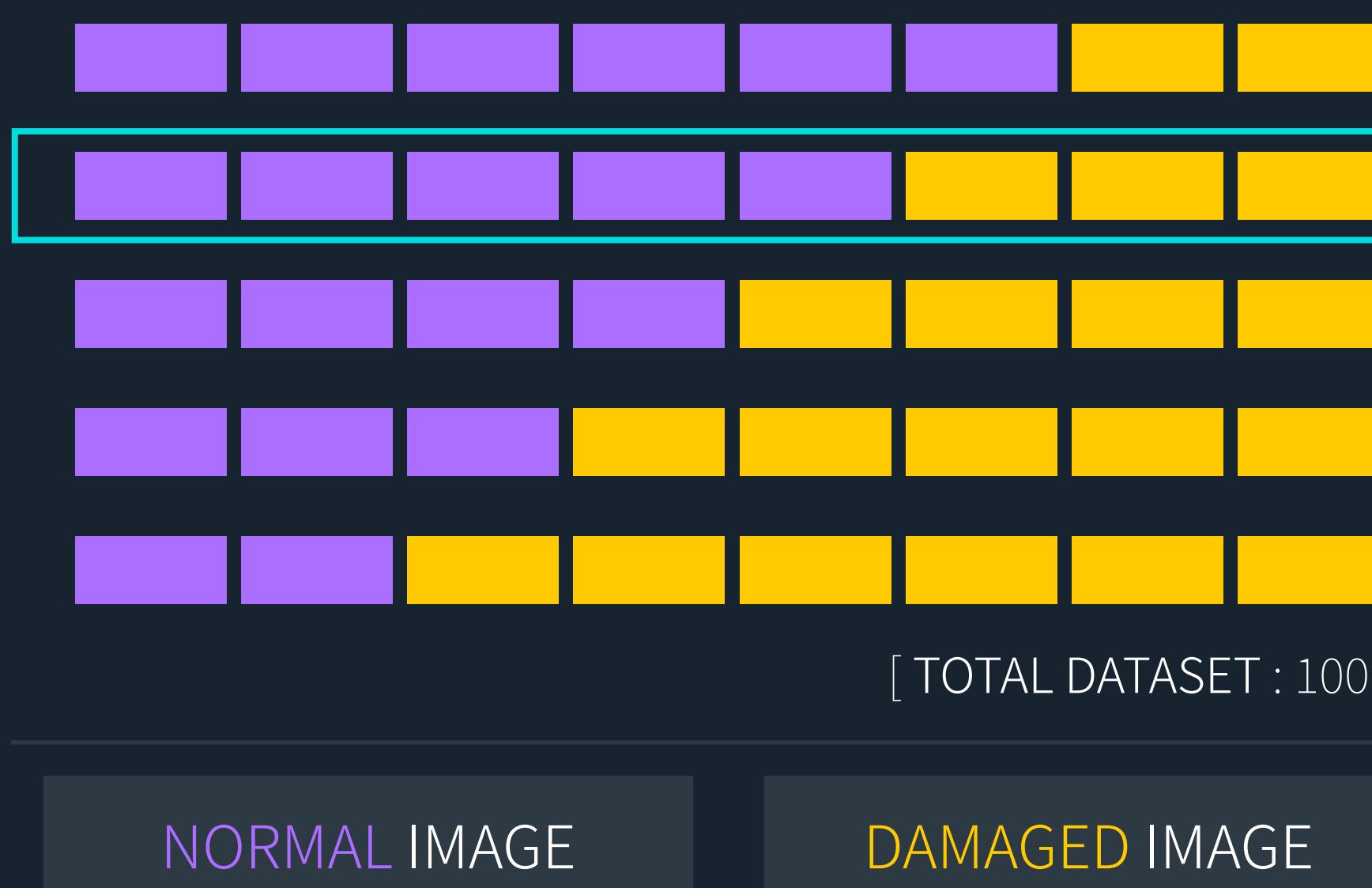
“파손 데이터와 정상 데이터의 비율이
MIoU 점수 하향에 영향을 미치는가?”

데이터의 비율을 조정 해보자

TEST_1 >>

◆ DATASET RATE SETTING

TEST _ 1

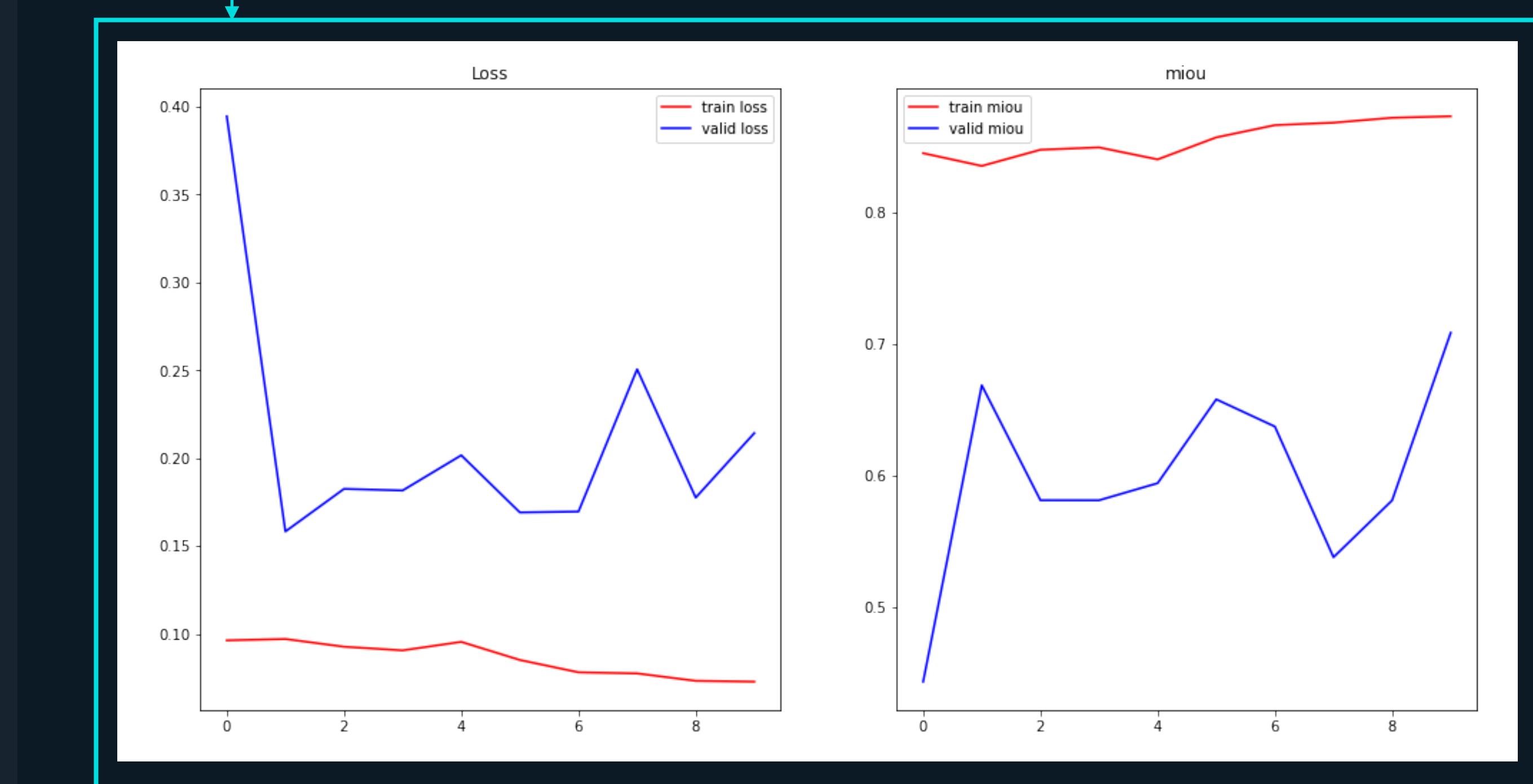


GRAPH 1 (LEFT)

Increasing Normal Data Rate

Epoch : 10

iou_back	0.968292
iou_scratch	0.448358
train loss	0.072800
valid loss	0.214234
train miou	0.872723
valid miou	0.708325



정상 데이터 : 파손 데이터 비율이 5 : 3 이었을 때,
좋은 성능을 확인할 수 있었다.

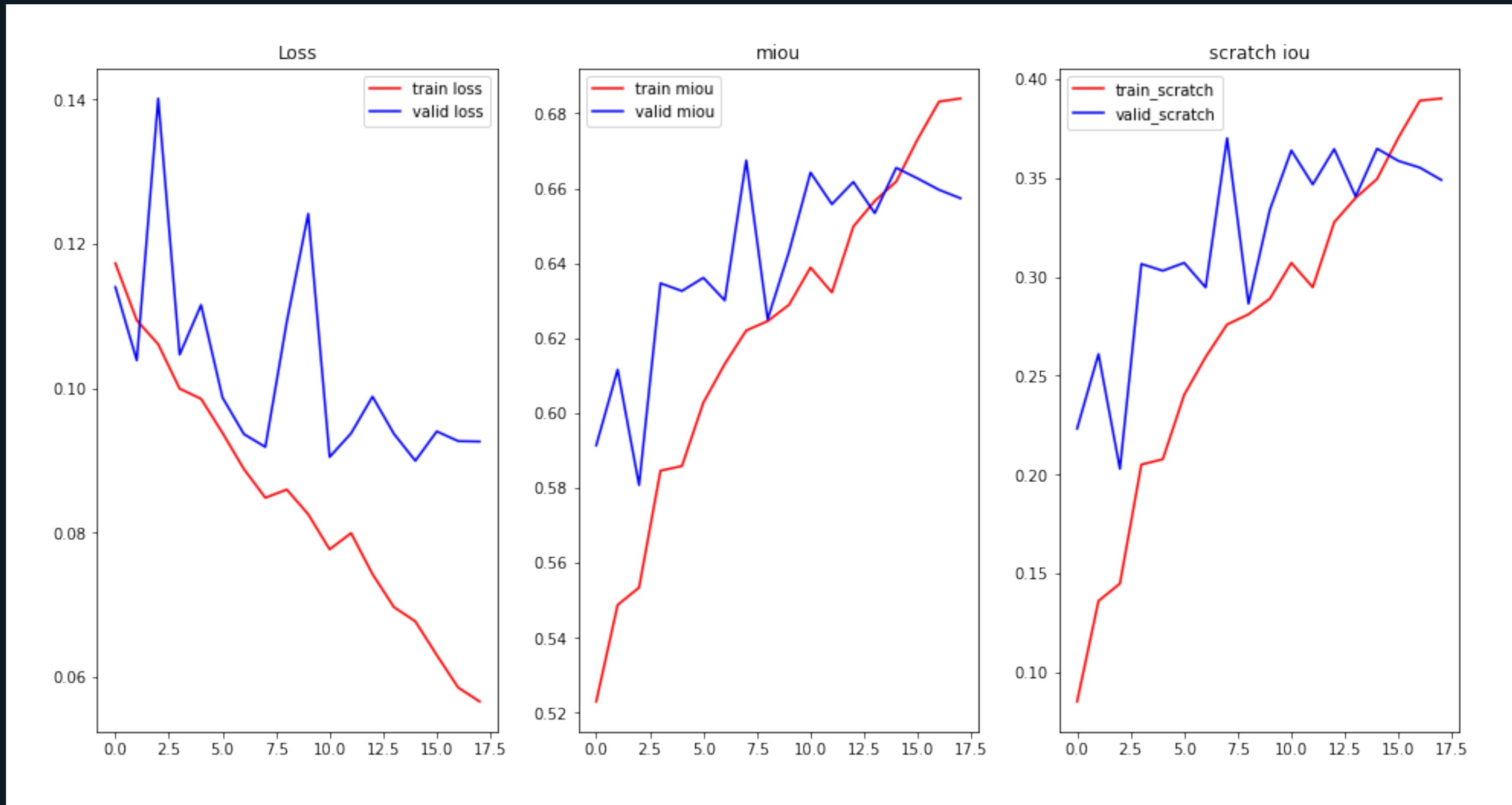
“ 정상 데이터의 비율 증가 : 준수한 valid MIoU
TEST _ 스크래치를 잡지 못하는 맹점 발생 ”

순수 스크래치 데이터만 학습시킨다면 어떨까?

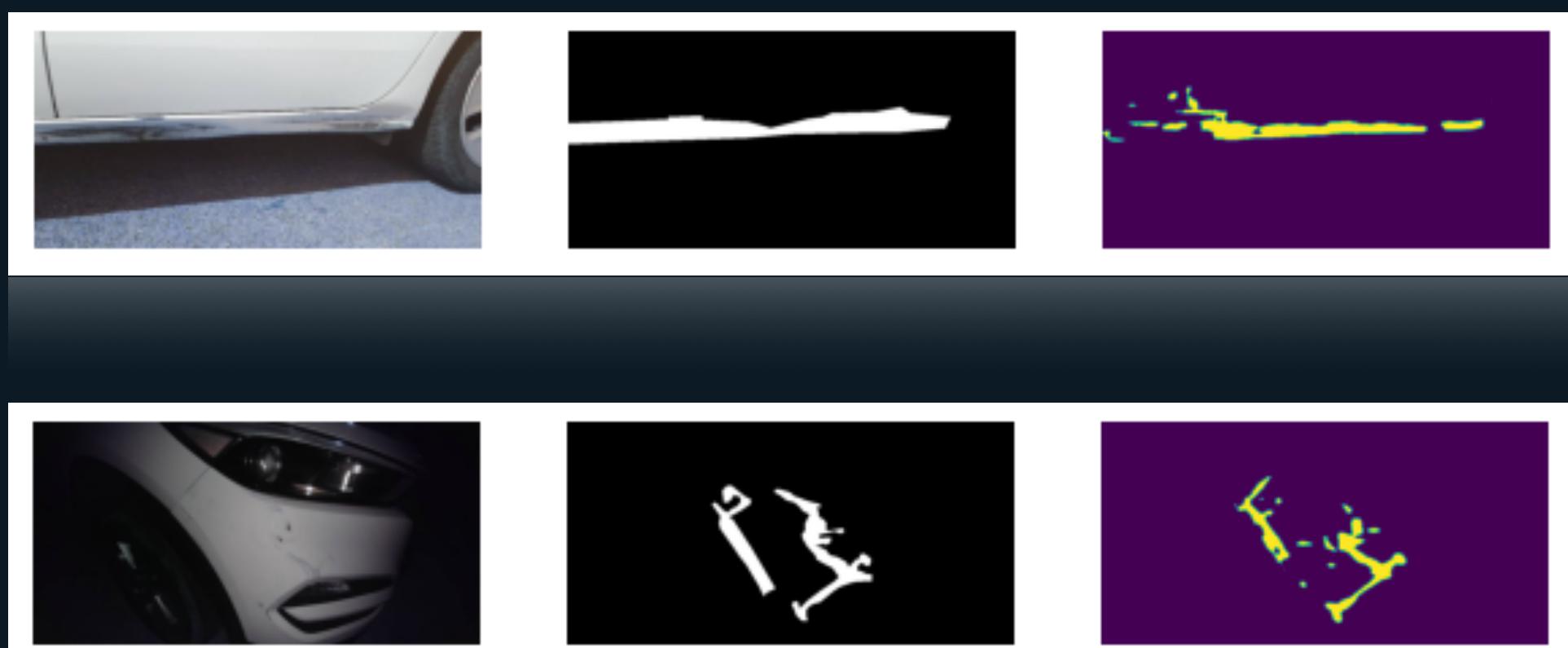
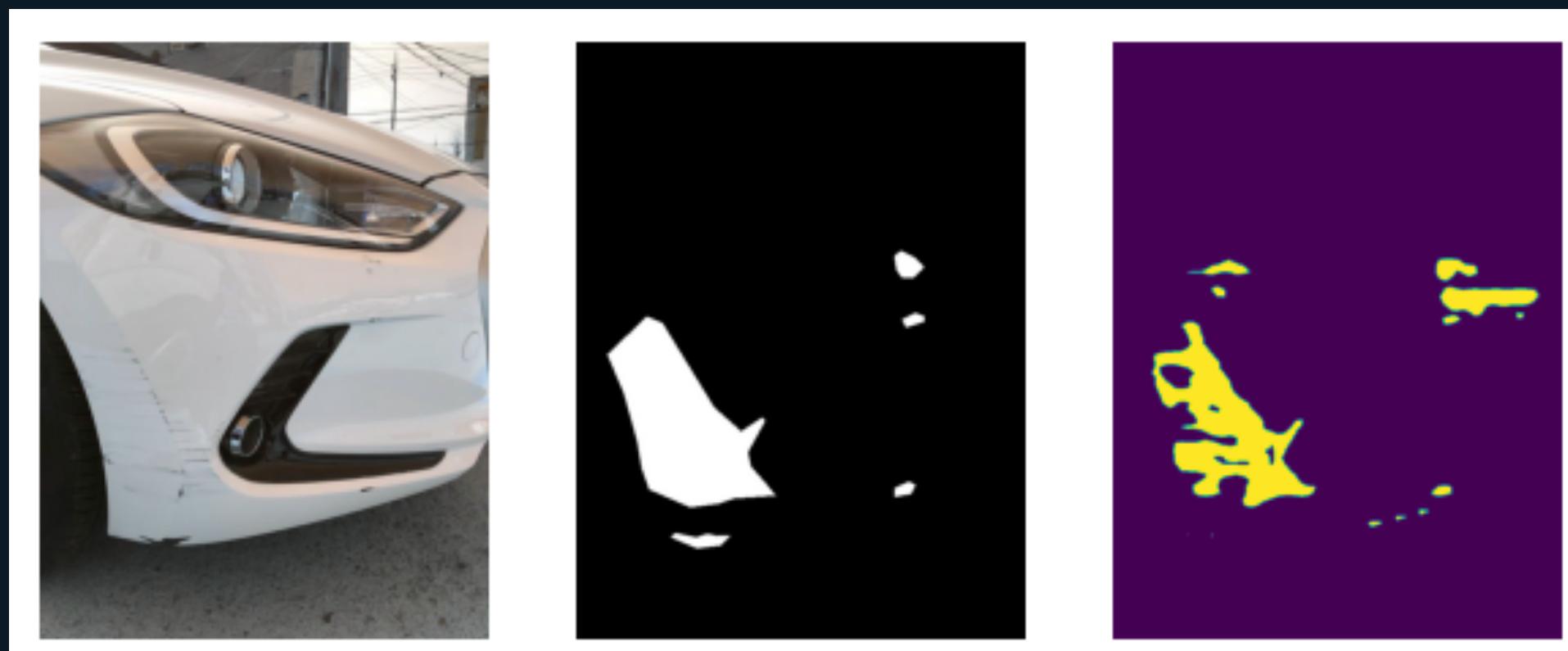
TEST _ 2 >>

TEST _ 2

TRINING SCRATCH IMAGE ONLY



TEST RESULT



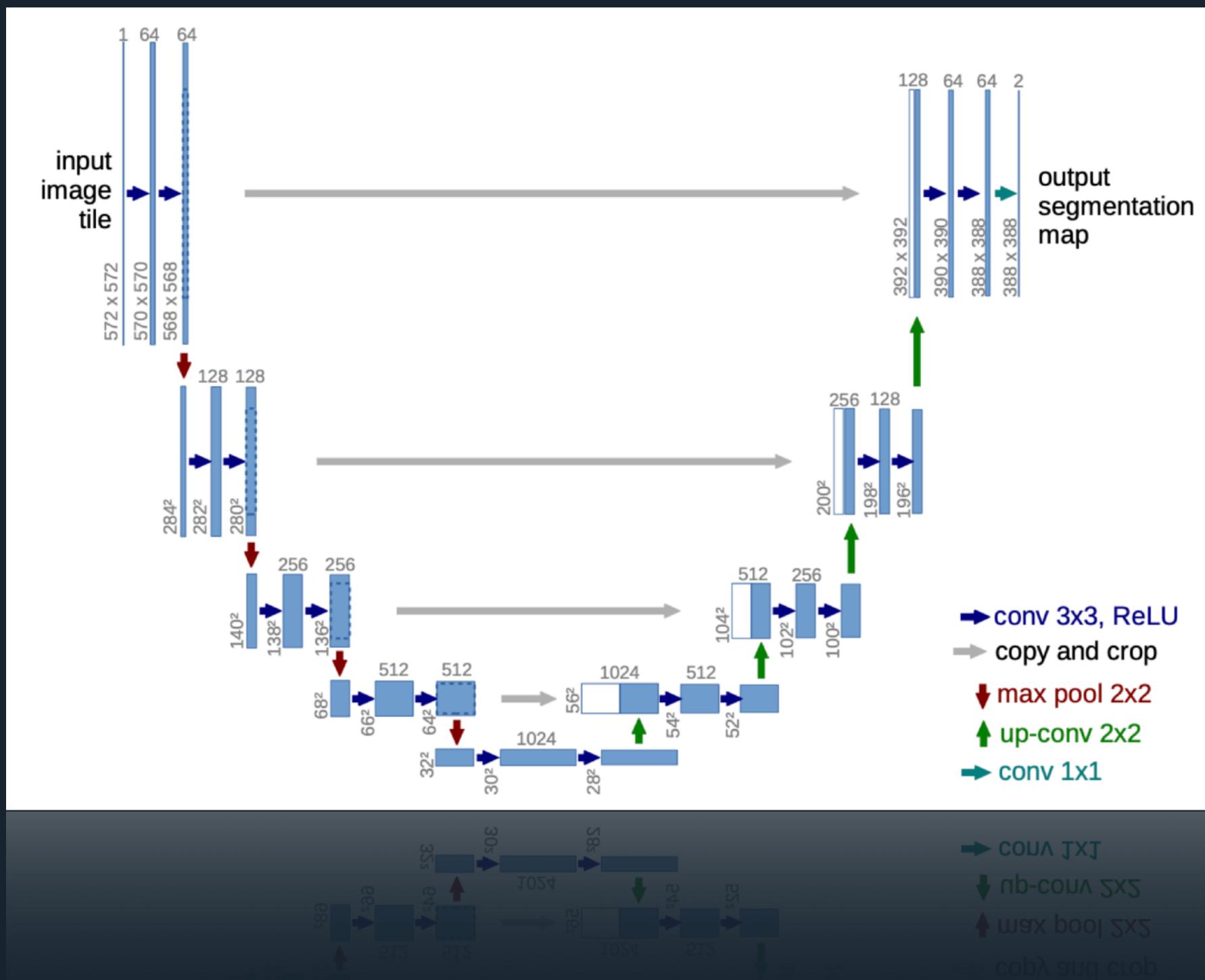


AFTER PLAN

“ TRYING MULTI LABEL
FROM BINARY LABEL ”

MODEL SEARCH

◆ UNet



DISSERTATION ABOUT UNet

정확도가 중요한 의료 업계에서의 UNet의 평가지표

Rank	Group name	Warping Error	Rand Error	Pixel Error
	** human values **	0.000005	0.0021	0.0010
1.	u-net	0.000353	0.0382	0.0611
2.	DIVE-SCI	0.000355	0.0305	0.0584
3.	IDSIA [1]	0.000420	0.0504	0.0613
4.	DIVE	0.000430	0.0545	0.0582
:				
10.	IDSIA-SCI	0.000653	0.0189	0.1027

- Semantic Segmentation의 대표적 모델
- 의료용 Biomedical image 분석에 사용한 모델
[세포 사이에 간격이 촘촘한 경우,
그 간격을 구분할 수 있도록 학습하게끔 만들어주는 장점이 있다.]
- 적은 데이터셋에 학습시켜도 더 좋은 성능이 나오는 모델

논문 저자 : “U-Net의 구조가 다양한 task에 쉽게 응용될 수 있을거라 확신”

결론 : 높은 정확성을 바탕으로 한 UNet 모델을 사용해 보기로 결정 하였다.

◆ UNet

MULTI LABEL

1. Git - Clone (Unet) 으로 학습 진행

출처 : <https://github.com/4uiiurz1/pytorch-nested-unet>

2. Dataset 오류 발견

ERROR 1

- Reference가 binary label을 위한 모델임을 인지

ERROR 2

- Input dic shape 의 비율 상이

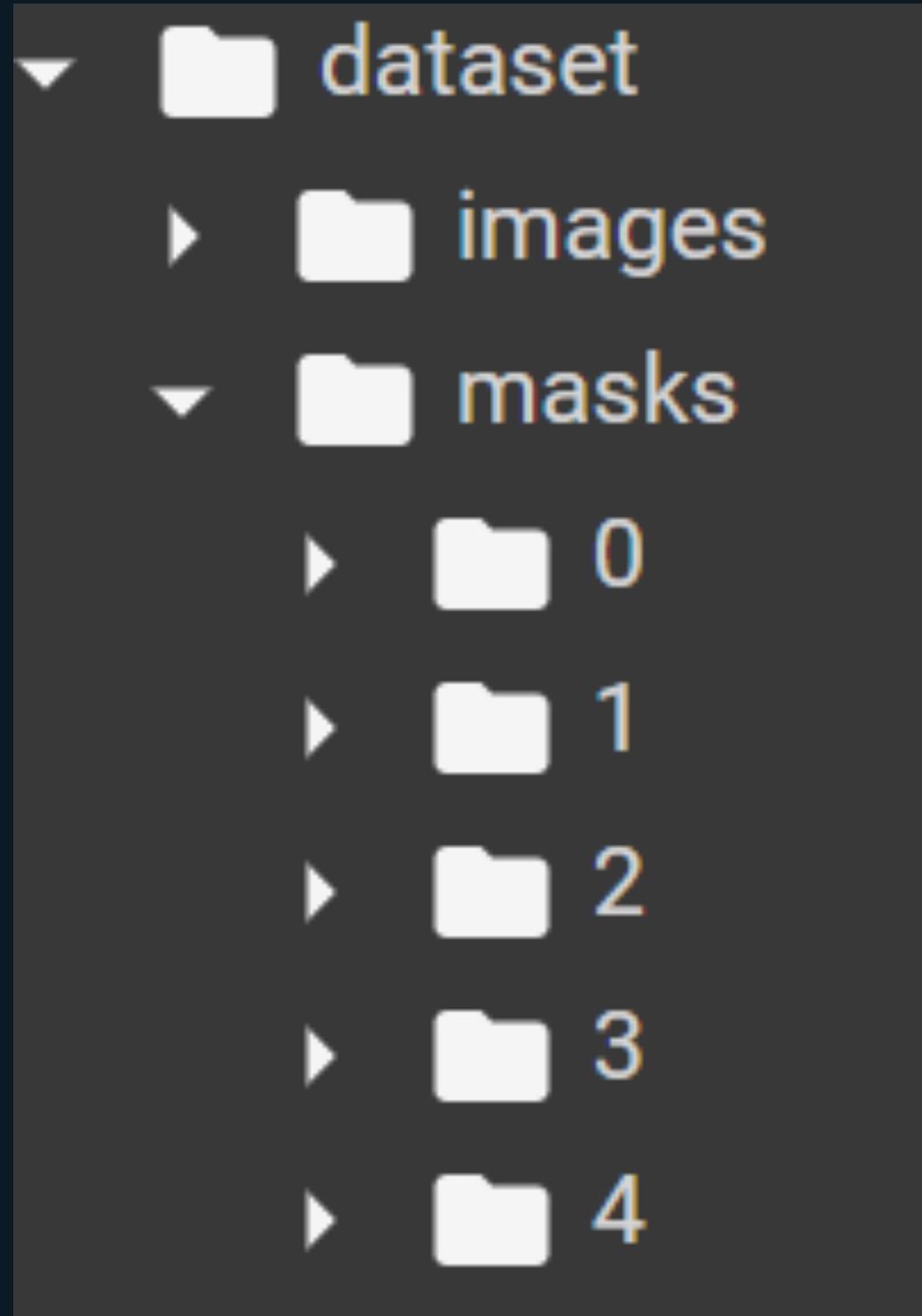
REF. | IMG : MASK = 1 : 1

OUR | IMG : MASK = 1 : 2

◆ PLAN

- 데이터셋 형태 변경 [IMG : MASK = 1 : 1]
- Multi Class Unet Reference 학습

```
inputs
└── <dataset name>
    ├── images
    │   ├── 0a7e06.jpg
    │   ├── 0aab0a.jpg
    │   ├── 0b1761.jpg
    │   └── ...
    └── masks
        ├── 0
        │   ├── 0a7e06.png
        │   ├── 0aab0a.png
        │   ├── 0b1761.png
        │   └── ...
        └── 1
            ├── 0a7e06.png
            ├── 0aab0a.png
            ├── 0b1761.png
            └── ...
```



ERROR 1

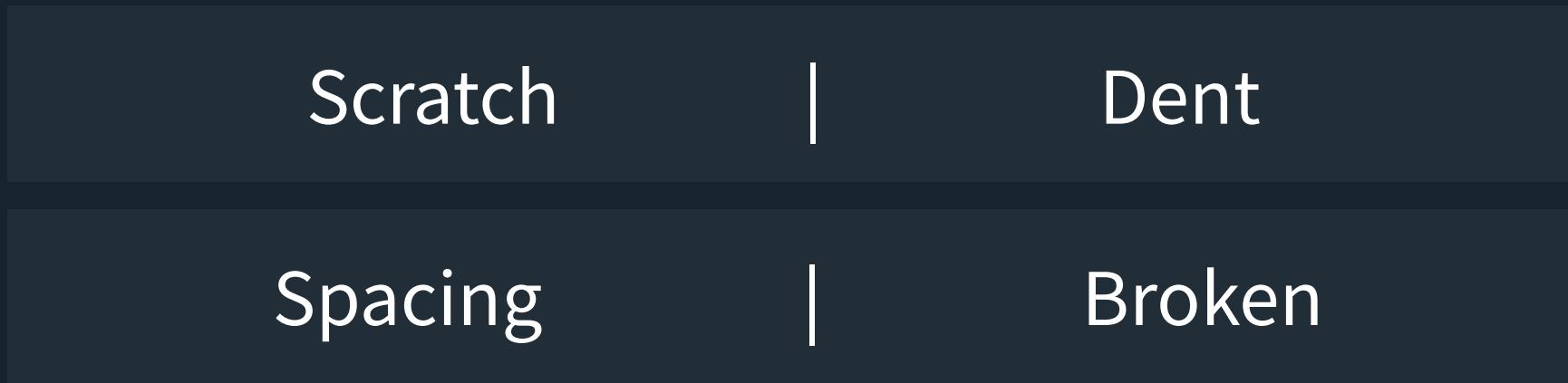
Reference Readme 오해석

ERROR 2

Input dataset dic shape

◆ CHALLENGE WAY ABOUT ERROR # 1

2개의 겹치는 영역 Label 1개 제거



각 Class_ids 에 색 부여 / 이미지 병합



MULTI LABEL

UNet CHALLENGE WAY ABOUT ERROR

```
1 mask_re_1 = []
2 mask_re_2 = []
3 mask_re_3 = []
4 mask_re_4 = []
```

4개의 이미지에 한번에 색 부여

```
1 # 1 = [0.3, 0, 0]
2 # 2 = [0, 0.3, 0]
3 # 3 = [0, 0, 0.3]
4 # 4 = [0.3, 0.3, 0.3]
```

```
1 for a in tqdm(mask_2_paths):
2     aa = plt.imread(a)
3     new = np.where(aa > 0, [[[0, 255, 0]]], [[[0, 0, 0]]])
4
5     mask_re_2.append(new)
```

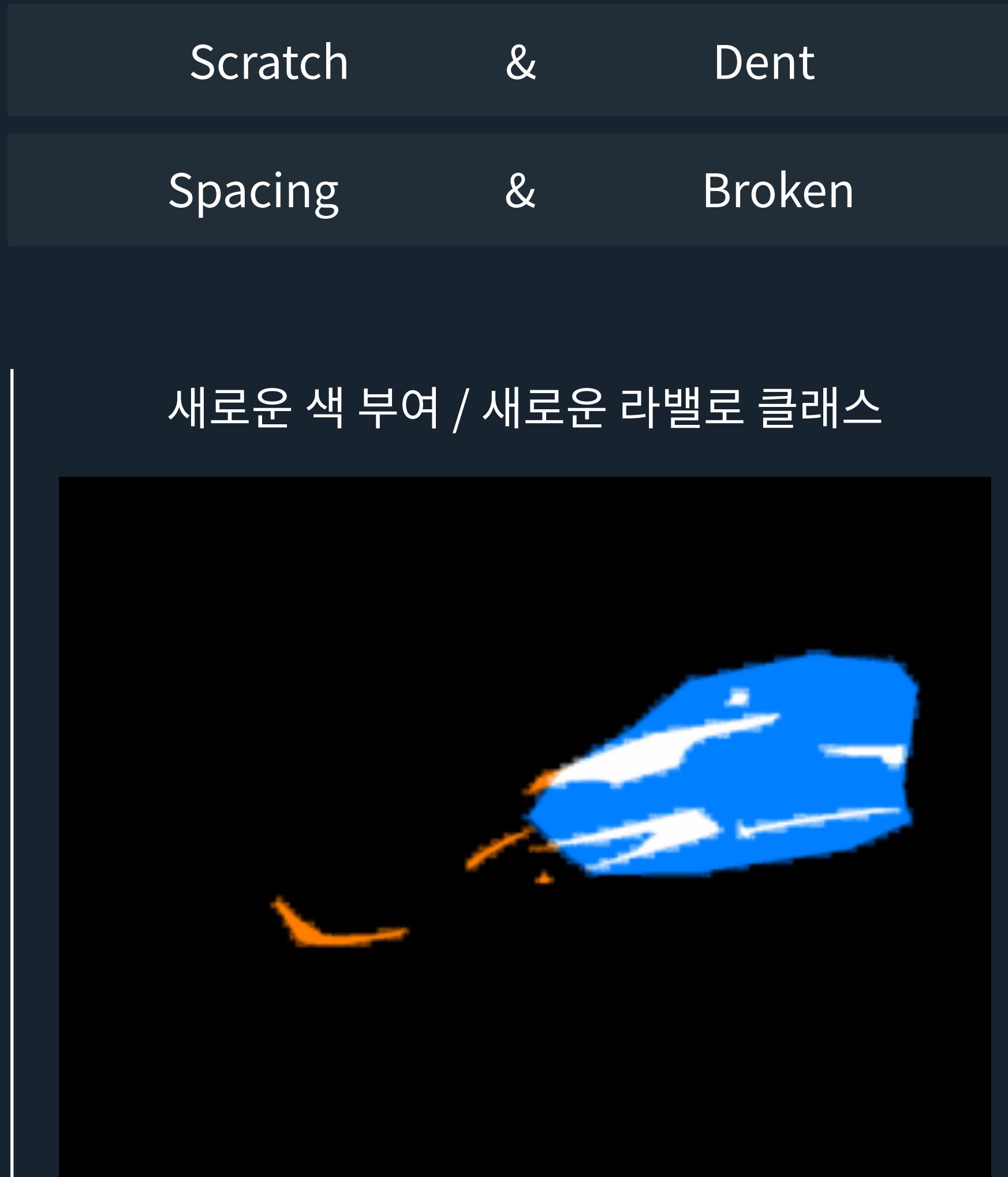
mask_re_2.append(new)

저장 과정 중의 RAM 용량 초과

[공간복잡도의 문제]

◇ CHALLENGE WAY ABOUT ERROR #2

겹치는 영역 New_Label로 생성



MULTI LABEL

UNet CHALLENGE WAY ABOUT ERROR

```
Class_ids = { '0' : 'background',
              '1' : 'scratch',
              '2' : 'dent',
              '3' : 'spacing',
              '4' : 'broken',
              '5' : 'scratch & dent',
              '6' : 'spacing & broken' }
```

07

[MULTI]

WEEK 3

CHALLENGE WAY ABOUT ERROR #2 내용 기반

DEVELOP PROJECT

>> 새로 구축한 데이터셋으로 학습 진행

WEEK 4

데이터셋이 부족할 경우

- augmentation 진행
- 아래 툴을 이용하여 데이터셋 보충

Create MultiClass Segmentation Dataset Tool

출처 : <https://www.apeer.com/home/>

WEEK 5

U-Net++

- U-Net3+ [U-Net+++] 까지 나왔음을 확인
- 모델크기 / 현실적인 상황을 고려한 선택
- 모델의 기본적인 구조 및 성능 연구
- Reference Search / 학습

FUTURE PLAN

PLANNER

❖ EXPECT EFFECTIVENESS

CURRENT WAY

차량 파손 형태 대한 MultiClass Segmentation 진행중

DEVELOP WAY

- 모델링 + 차량 파손 부위 detecting 추가
- 파손부위, 형태, 면적에 따른 수리비용 추가

고객은 빠르게 견적을 확인가능

기업은 수작업의 번거로움을 딥러닝 모델에 적용시킴으로써
문제를 다소 해결할 수 있을거라 기대