



**VISOL**

# 합성데이터 기반 객체 탐지 AI 경진대회

팀 쥬혁이

쥬혁이, Bing, ever4red, 중요한건꺾이지않는마음

2023.05.08 ~ 2023.06.19

# Contents

- ▶ I. Intro
- II. EDA
- III. 합성 데이터 활용
- IV. 모델검증
- V. 실험관리

1

## 대회 배경 및 목표

실사와 같은 객체와 배경을

Visol의 3D Rendering과

VFX 기술로 생성된 AI 학습용

고품질 합성데이터를 활용해

자동차 탐지 AI 모델을 개발

2

## 평가 항목

- 모델 성능
- 모델 검증
- 합성데이터 활용
- 실험 관리
- 자료 완성도

3

## 평가 산식

- mAP(IoU threshold=0.85)

→ 정확한 Bounding Box를

탐지하는 능력이 필요합니다.

R-CNN은 물체가 있을만한 영역을

찾고, 각 영역별로 분류 및 조정하는

단계를 거쳐, 정확한 Bounding Box

탐지가 가능하므로 R-CNN 모델

사용 전략 설립이 필요합니다.

# Contents

- I. Intro
- ▶ II. EDA
- III. 합성 데이터 활용
- IV. 모델검증
- V. 실험관리

# EDA

합성 이미지와 실제 세계의 이미지간의 차이를 이해하고,  
이 차이를 극복하기 위한 전략을 구성하는 것이 중요하며,  
합성 데이터의 완벽한 라벨링 같은 장점을 최대한 활용하는 것이  
필요합니다.

- 1 | Domain Knowledge
- 2 | Class 별 시각화
- 3 | Insight

# [클래스 간 분포 확인]

CarDataset Train dataset with number of images 6481, and instance counts:

category	count	category	count	category	count	category	count	category	count	category	count
0 [0]	500	1 [1]	500	2 [2]	500	3 [3]	500	4 [4]	500		
5 [5]	500	6 [6]	500	7 [7]	500	8 [8]	500	9 [9]	500		
10 [10]	500	11 [11]	500	12 [12]	500	13 [13]	500	14 [14]	500		
15 [15]	500	16 [16]	500	17 [17]	500	18 [18]	500	19 [19]	500		
20 [20]	500	21 [21]	500	22 [22]	500	23 [23]	500	24 [24]	500		
25 [25]	500	26 [26]	500	27 [27]	500	28 [28]	500	29 [29]	500		
30 [30]	500	31 [31]	500	32 [32]	500	33 [33]	500	-1 background	0		

각 클래스 별 500개로 불균형이 전혀 없는 상황

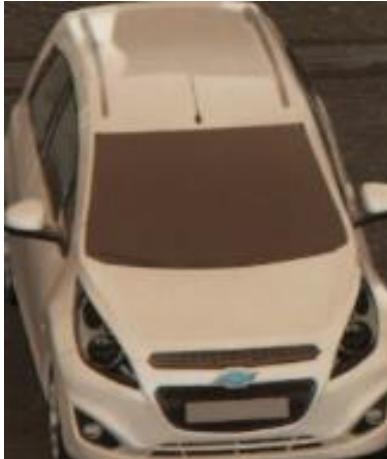
# [Label Miss Match]

스파크 1세대 (2011~2015)



Class.txt 파일에 2번으로 라벨링이 되어있는  
스파크 차량은 2016~2021년으로 표기가 되어있습니다.  
하지만 학습 데이터의 EDA 이후  
저희의 Domain Knowledge와는 다르게 학습 데이터의 이미지는  
2세대가 아닌, 1세대 스파크 차량들과 유사하다 판단했습니다.

## 학습 데이터



로고 위치가 1세대 차량과 같음 + 외관

## 라벨링 된 값

002, 쉐보레 스파크 2016~2021

스파크 2세대



출처 : Daum 자동차

## 스파크 2세대 (2016~2018)



출처 : Daum 자동차

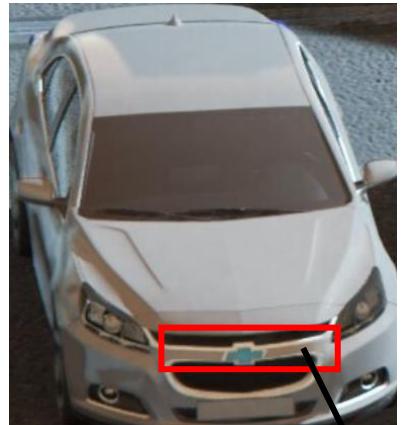
## 스파크 2세대 (2019~)



출처 : Daum 자동차

# [Domain Knowledge]

0번 class



그릴 위 부분이 뒤집혀 있음



32번 class



그릴의 가로 격자가 없음



Domain Knowledge 설명 이미지

실제 이미지와 거의 유사하나, 미세한 부분의 표현이 어렵습니다

# [Domain Knowledge]

IONIQ 일렉트릭 2019



IONIQ 일렉트릭 2016-2018



IONIQ 하이브리드 2017-2018



IONIQ 하이브리드 2019



학습 데이터에는 하이브리드 2019만 존재



Domain Knowledge 설명 이미지

12번 class인 IONIQ 2017~2019 의 경우 연식과 하이브리드 or 일렉트릭에 따라 총 4가지 종류가 있습니다.

하지만 EDA 결과 학습데이터는 IONIQ 하이브리드 2019만 존재하였습니다.

티볼리 (1세대) 2015-2017



티볼리 F/L (1세대) 2020



학습에는 2020년식만 존재함



티볼리 아메 2017-2019



티볼리 에어 2016-2019



Domain Knowledge 설명 이미지

32번 class의 경우 티볼리 2016~2020는 1세대(15~17, 20~), 아메, 에어 등의 종류가 있지만

학습 데이터는 티볼리 1세대 2020년식만 존재하였습니다.

Avante N Line(2020)



Avante (2020)



Sonata 2020 Sensuous



Sonata 2020

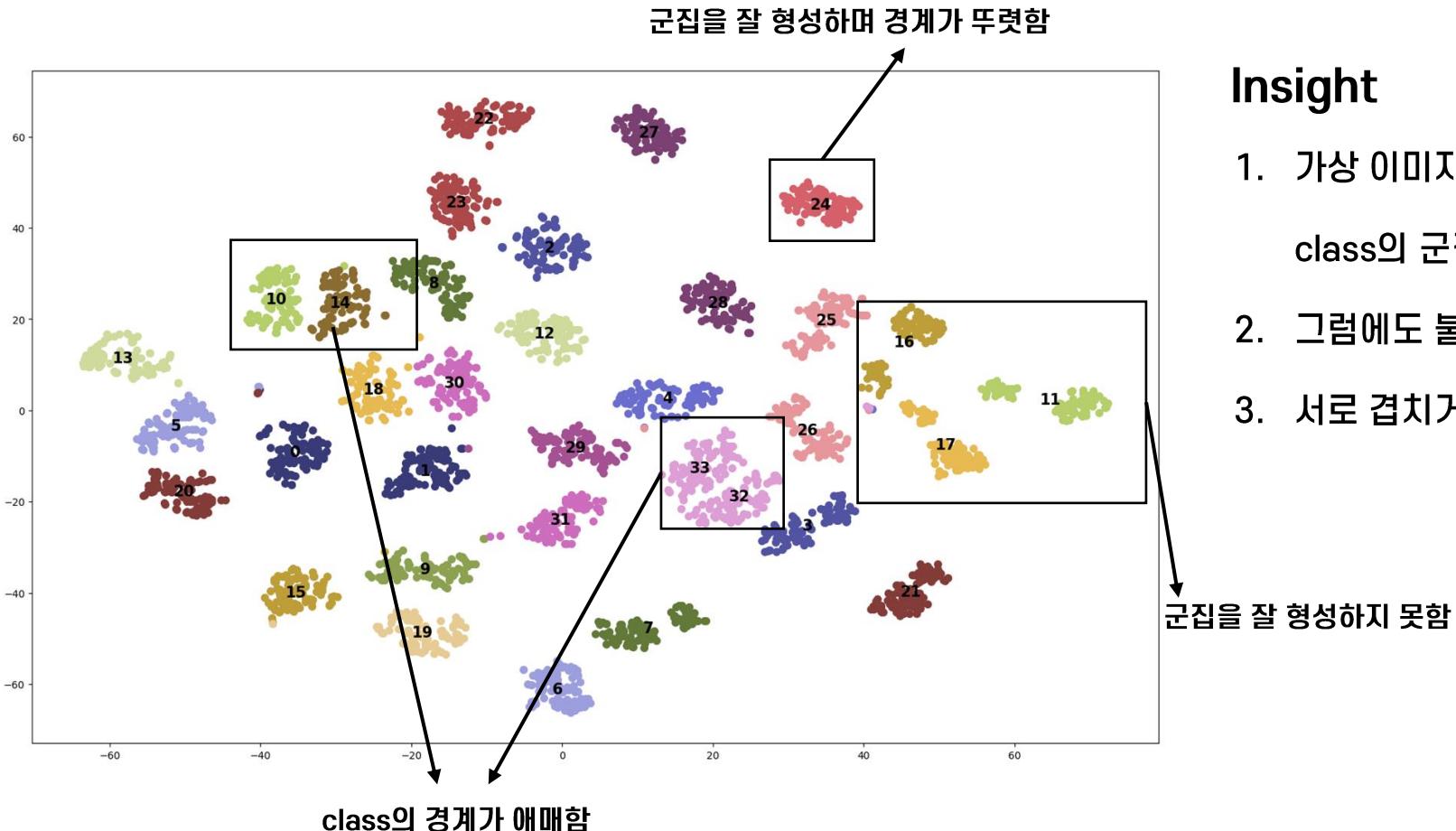


## 같은 차량 내 여러 Line 존재

실제 차량은 같은 연식 모델이라도 여러 Line으로 출시가 되고 소비자의 기호에 따라 약간씩 다른 디자인을 선택할 수 있습니다.

저희의 학습 데이터에는 기본 라인만 존재하지만, 현실 세계에서는 여러 라인이 존재합니다.

저희는 ImageNet으로만 pre-train된 모델(ViT)로 각 차량 class image의 feature를 추출하여 t-SNE그래프를 확인하였습니다.

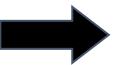


t-SNE : t-SNE란 고차원 데이터를 2차원으로 축소할 수 있고 낮은 차원의 데이터를 시각화 하여 데이터 구조 이해하는 방법론입니다.

## Feature들의 유사도 비교(Cosine similiarity)

- tSNE로 유사한 class가 꽤 존재한다는 것을 확인 하였고 더 정확한 정량적인 측정을 하기 위해 각 class 별 500개의 image feature를 평균 한 후 코사인 유사도를 비교하였습니다.

class	10	...	32
14	0.9710	...	...
...	...	...	...
33	...	...	0.9683



서로 코사인 유사도가 0.96 이상 class들을 찾았고 그 결과 저희는 (10번과 14번), (32번과 33번) 등의 클래스가 매우 유사하다는 것을 확인할 수 있었습니다.

현실 세계에서의 여러가지 변수들을 고려한다면 이러한 class들은 test에서 분류하기 어려워 할 것이라고 생각하였습니다.

따라서 서로 유사한 class 를 잘 분류하는 방법을 고안하는 것이 중요하다는 Insight를 얻었습니다.

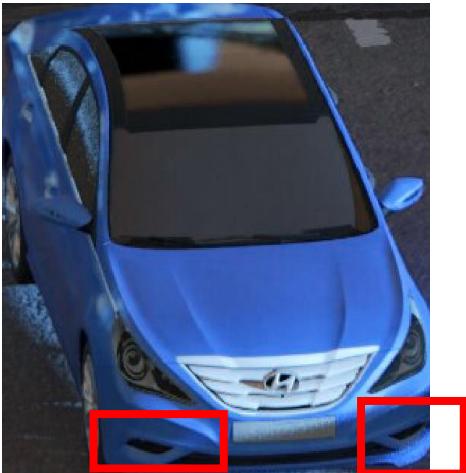
# [Check Insight]

유사한 차량들이 왜 비슷한지, 어떤 특징을 가지고 있는지를 확인해보았습니다.

10번



14번



32번



33번

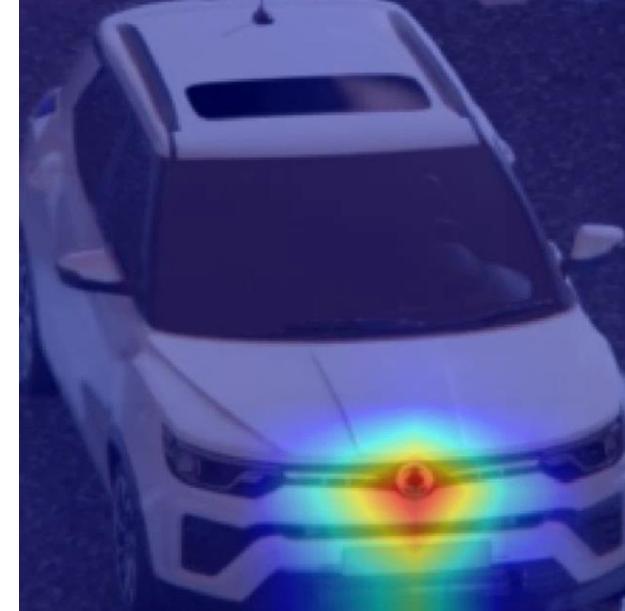


10번, 14번의 경우 로고, 그릴, 자동차의 쉐입이 비슷합니다.  
→ 포그램프를 집중적으로 봄아 구별하기 쉽습니다.

32번, 33번의 경우 헤드라이트, 포그램프 등이 유사합니다.  
→ 라디에이터 그릴의 아래 부분을 봄아 구별이 쉽습니다.

→ 유사한 차량들은 제조사(로고)가 같거나, 유사한 특징들을 공유합니다.

따라서 차량의 class를 잘 구분하기 위해서는 한 부분만 보는 것보다는  
차량의 여러 부위를 전부 고려해야 한다는 Insight를 얻었습니다.



또한 저희는 실제로 모델이 자동차의 어느 부분의 특징으로 차량의 class를 판단하는지 알아보기로 하였습니다.

학습데이터의 bbox를 이용해 이미지에서 차량만 crop하여 추출하였고 간단한 classification을 해보았습니다.

Grad CAM을 확인해본 결과 class 예측 시 일부분(특히 로고와 그릴)을 보고 속단하는 경향이 있는 것을 확인하였습니다.

여기서 저희는 로고와 그릴이 차량의 class를 예측할 때 핵심적인 역할을 한다는 것을 확인하였습니다.

## 전체 Pipe 라인 설계의 방향성

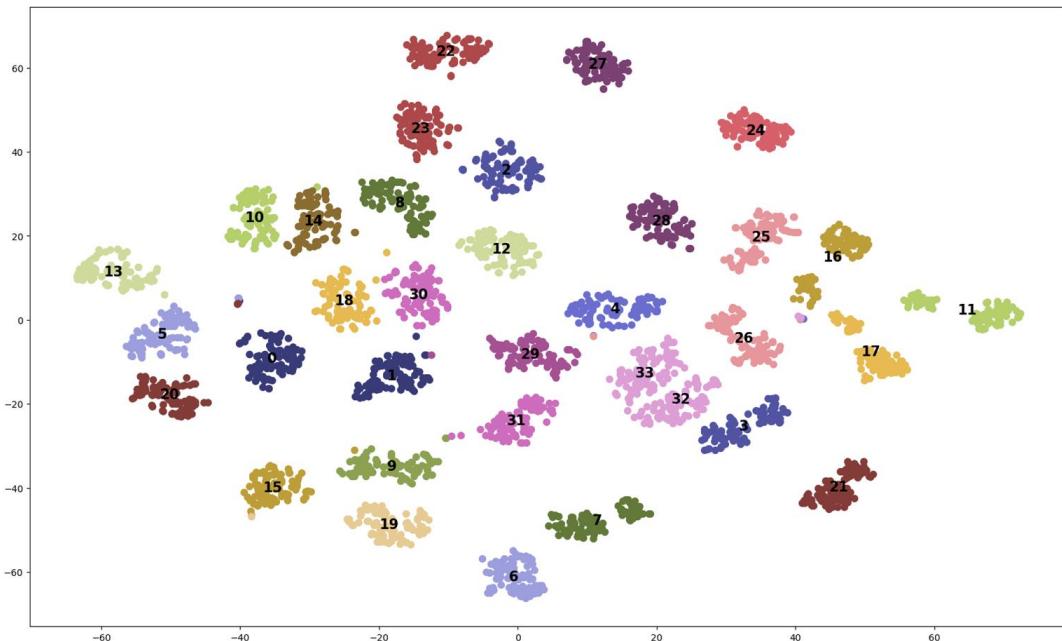
EDA를 통해 차량의 class를 예측하기 위해서는 로고와 그릴이 중요하다는 것을 확인하였습니다.

하지만 정확한 예측을 위해서는 포그 램프, 보닛, 헤드라이트 등 여러 특징들을 함께 보는 것이 중요하다는 것도 함께 확인하였습니다.

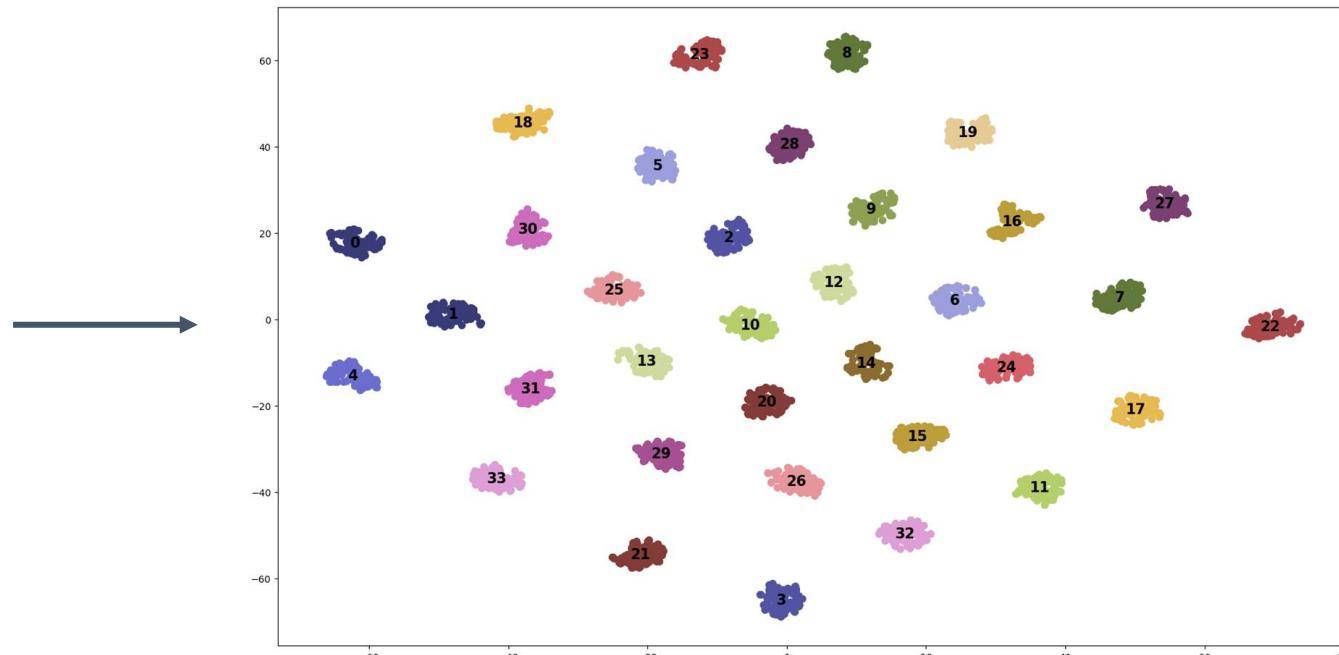
그래서 이를 중점적으로 augmentation 기법들을 설계 하였고 최종적으로는 Test에서

Validation처럼 class 간의 경계를 명확하게 하고, 군집화를 잘하게 하는 것을 목적으로 하였습니다.

Validation(before train)



Validation(after train)



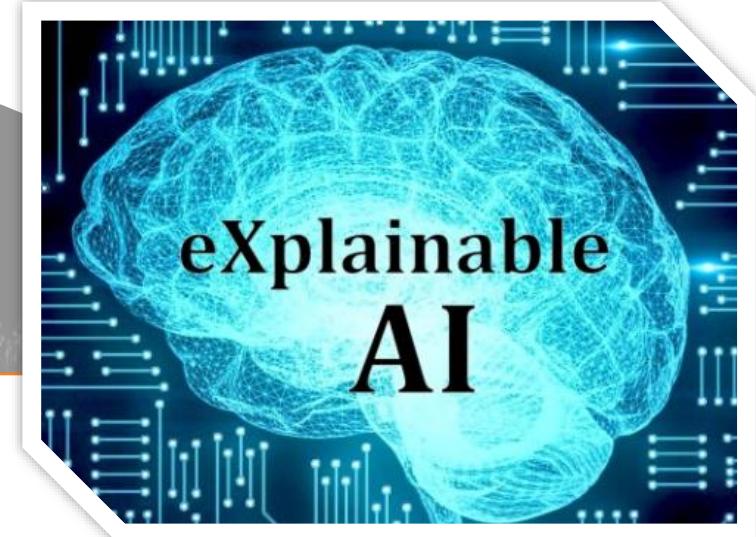
# Contents

- I. Intro
- II. EDA
- ▶ III. 합성 데이터 활용
- IV. 모델검증
- V. 실험관리

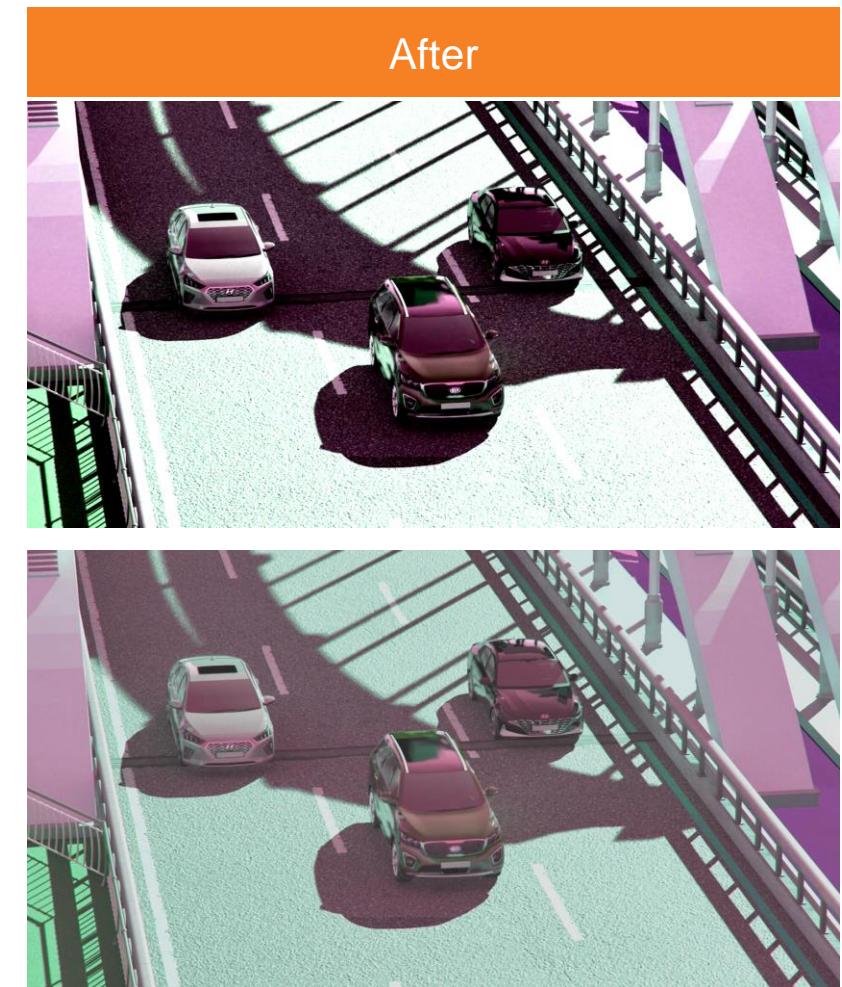
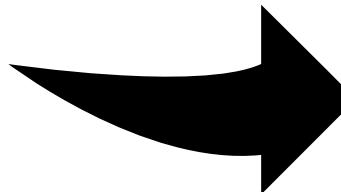
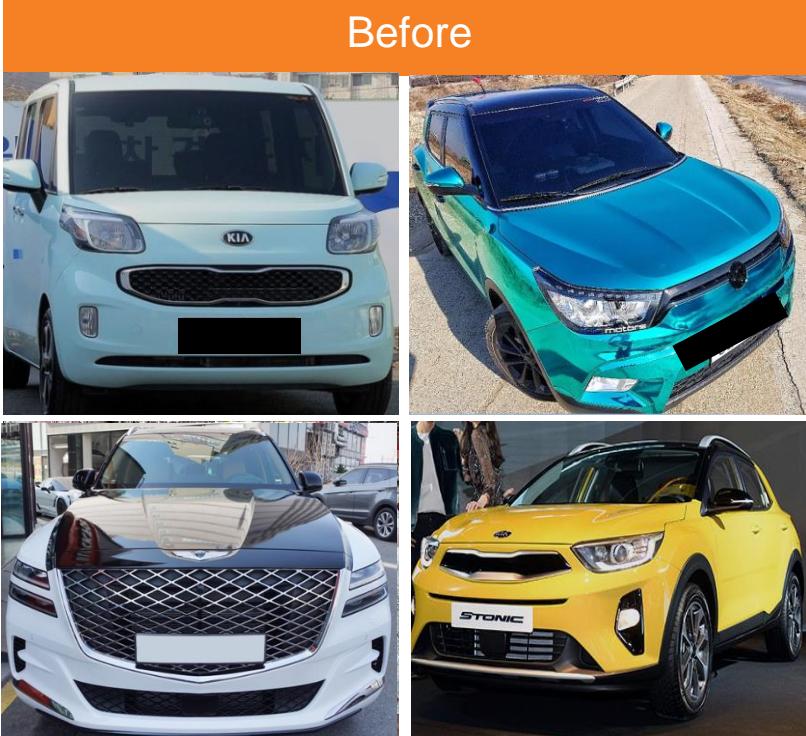
# [ 합성 데이터 활용 증강 ]

합성 이미지의 특성을 잘 이용하는 것이 이번 대회의 주요 목적이라고 판단했습니다.

그래서 Data Augmentation을 통해 Synthetic 이미지의 표현력을 다양하게 만들었습니다.



## [color jitter]



현실 세계에서는 다양한 색깔의 차 이미지가 존재 합니다.

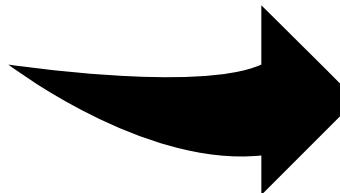
Train 데이터셋에는 위와 같은 색깔의 차들이 존재하지 않고,  
투톤으로 튜닝한 자동차의 많은 조합도 고려해 주어야 하므로,  
color jitter 를 이용해 Hue(색상), Saturation(채도),  
Lightness(명도)을 임의로 변경 하였습니다.

# [Brightness + contrast]

Before

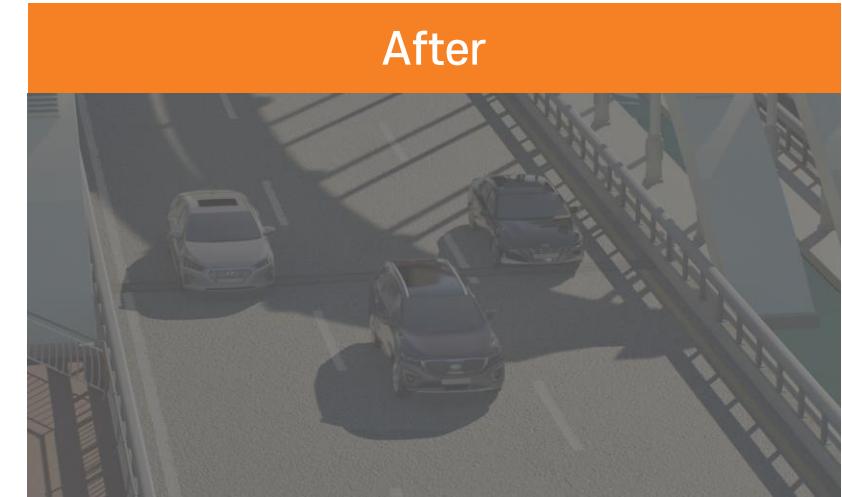


실제 도로 상황에서는 날씨, 시간, 구름  
등의 영향으로 인해 밝기와 대조에  
무한한 가능성이 존재하기 때문에,  
합성 이미지에서 도로 및 차량의 밝기와  
대조의 표현력이  
실제보다 부족할 수 있습니다.



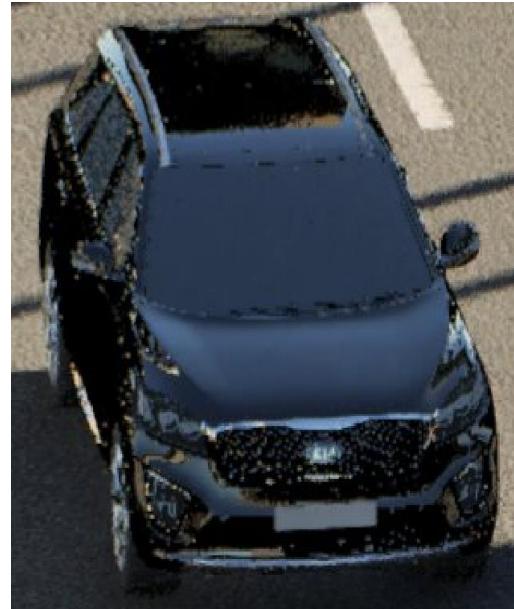
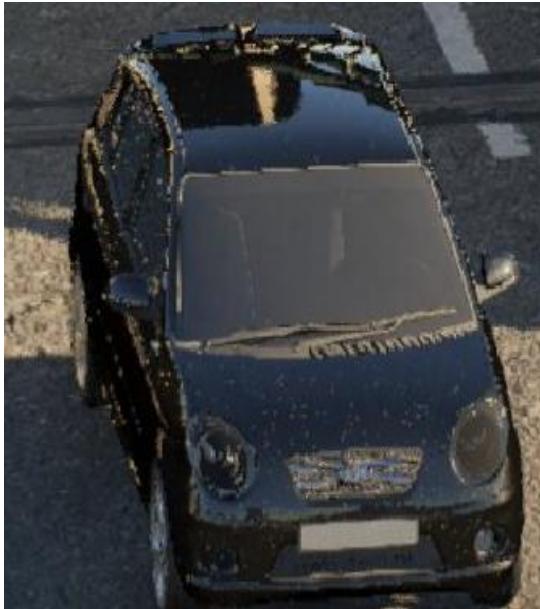
현실 세계의 변화를 시뮬레이션하여  
모델이 다양한 조명 상황에서 차량을  
더 정확하게 탐지하도록 학습 가능하게  
했습니다.

After



# [Equalize Transform]

Before



After

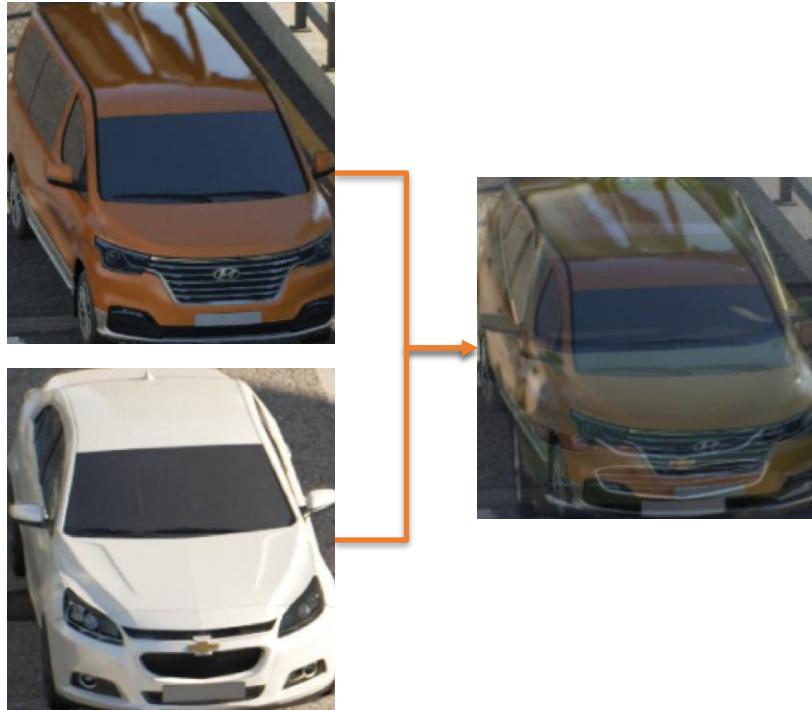


1. 합성 이미지에서의 차량은 실제 이미지의 차량보다 간단한 텍스처와 패턴으로 표현될 수 있고, 현실에서는 차량의 브랜드, 모델, 년도에 따라 차량의 텍스처와 패턴이 크게 다릅니다.
2. 합성 이미지는 때때로 원본 이미지에 비해 노이즈가 많을 수 있습니다.

현실 세계의 이미지의 세부 텍스처와 패턴을 강조하여, 모델이 다양한 차량의 특징을 더 잘 인식하도록 학습 가능하도록 했습니다.

# [Mixup]

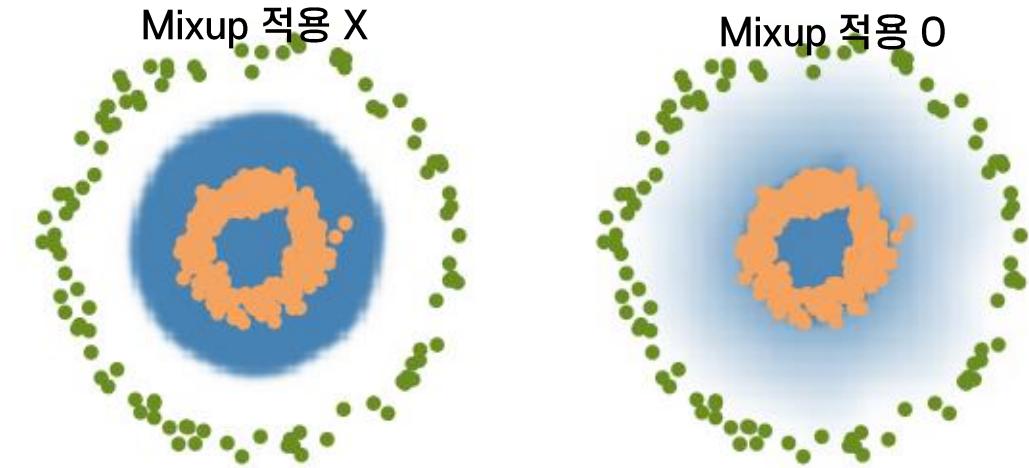
## 이미지 간 분류 성능 개선



데이터는 각 클래스간 500개씩으로 상당히 제한적입니다.

Mixup은 두 가지 이미지를 선형적으로 결합하여  
클래스 간 구분을 더 잘하게 할 수 있습니다.

## Decision Boundary 확장



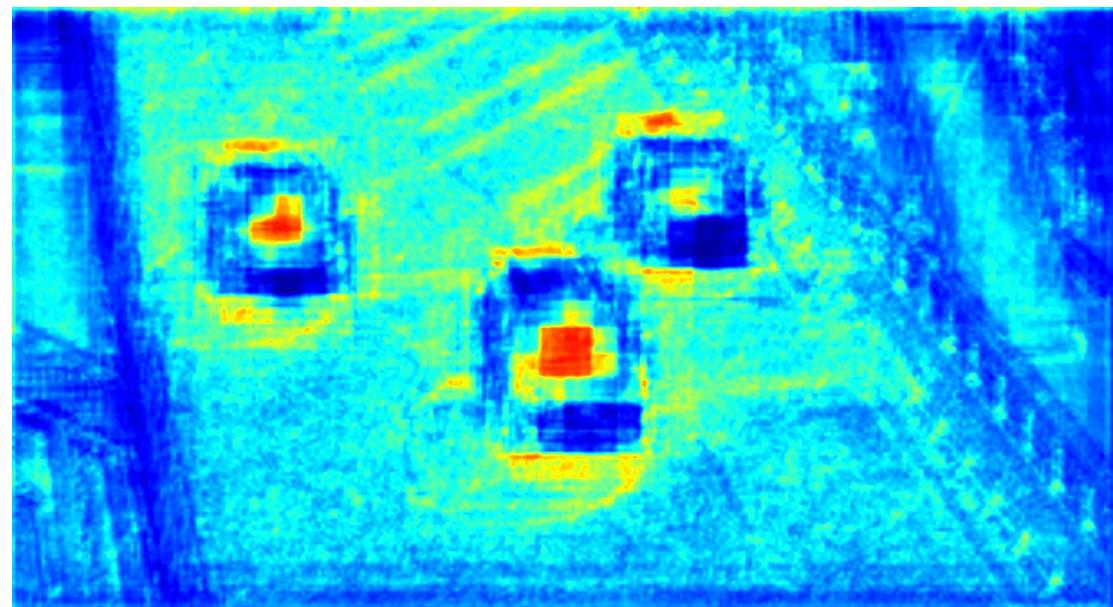
위 그림은 두 클래스에 대한 분류 상황을 나타냅니다.

파란색 범위가 두 클래스 간 경계선

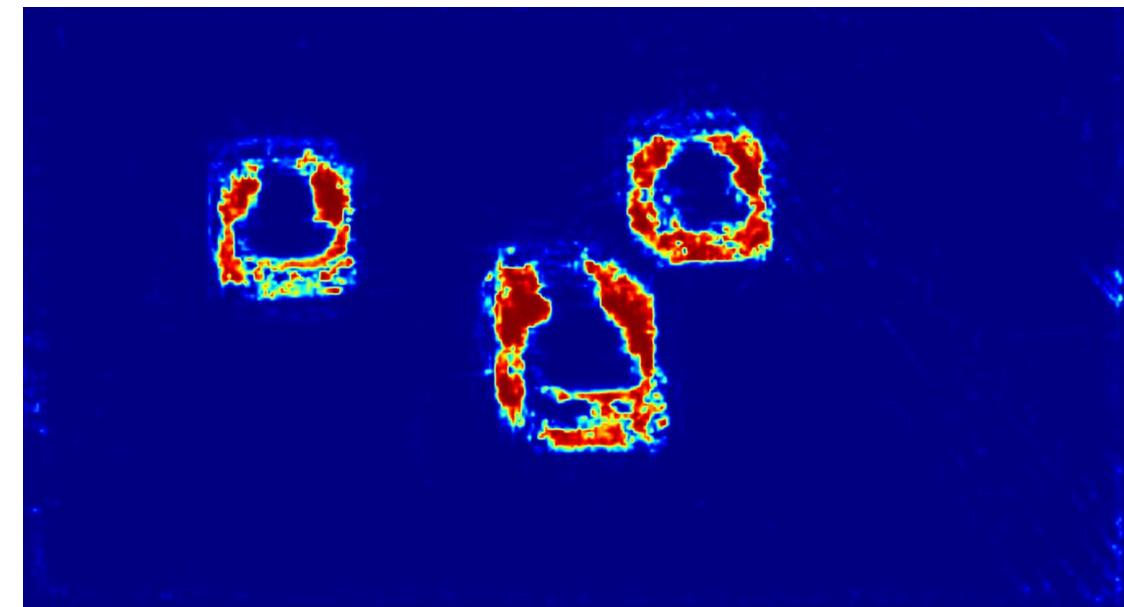
Mixup이 적용되지 않을 경우, 파란색 위치에 위치하는 데이터들은 한쪽클래스(초록색 or 주황색)으로 과도하게 확신하는 경향을 갖게 됩니다. 하지만, Mixup을 적용하게 되면, 중간에 위치하는 데이터에 대해 과도하게 확신하지 않아 성능이 향상될 수 있습니다.

## [Mixup]

Before Mixup



After Mixup



Grad CAM으로 Train 이미지 비교 결과

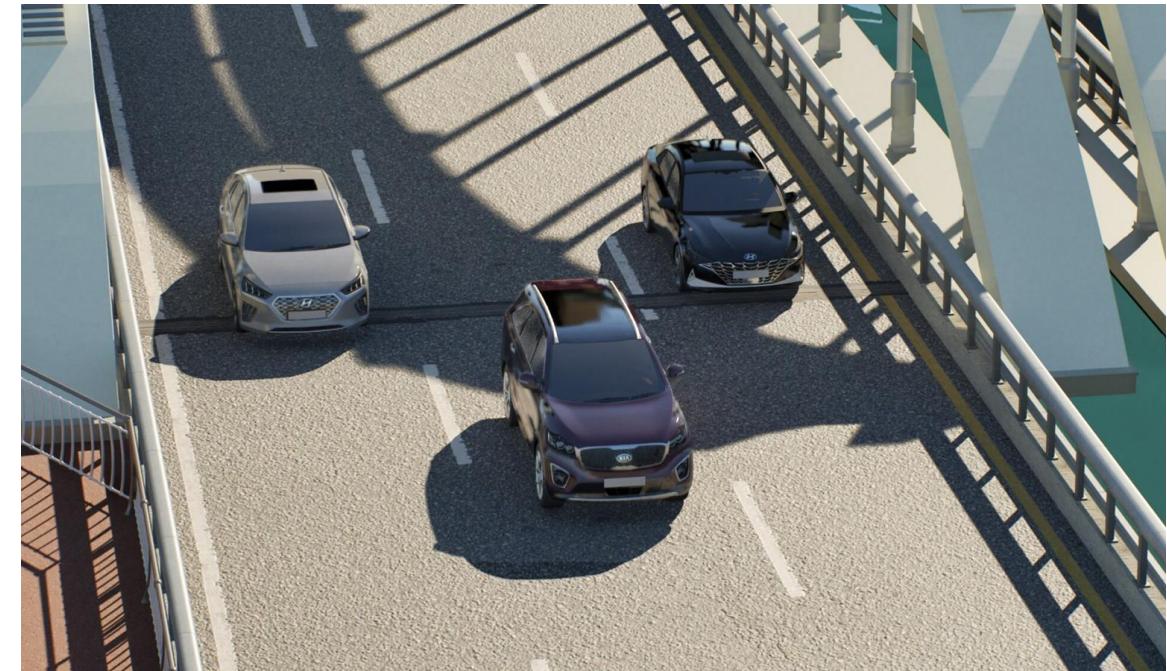
Mixup을 사용할 경우 차 본연에 더 집중하는 것을 확인할 수 있습니다.

# [Custom Cutout]

Cutout01 필요한 이유(1) - 가려짐  
차량이 가려짐



가려지는 차량이 없음



실제 세계에서는 다른 차량 등에 의해 운전 차량이 부분적으로 가려질 수 있습니다. (왼쪽 이미지)  
그러나 합성 이미지(오른쪽 이미지)는 가려지지 않은 완전한 차량만 존재하는 문제가 있습니다.

## [Custom Cutout]

Cutout0이 필요한 이유(2) - 차량 간 차이

실제 학습 이미지



2번 class의 이름 : 쉐보레 스파크 2016~2021

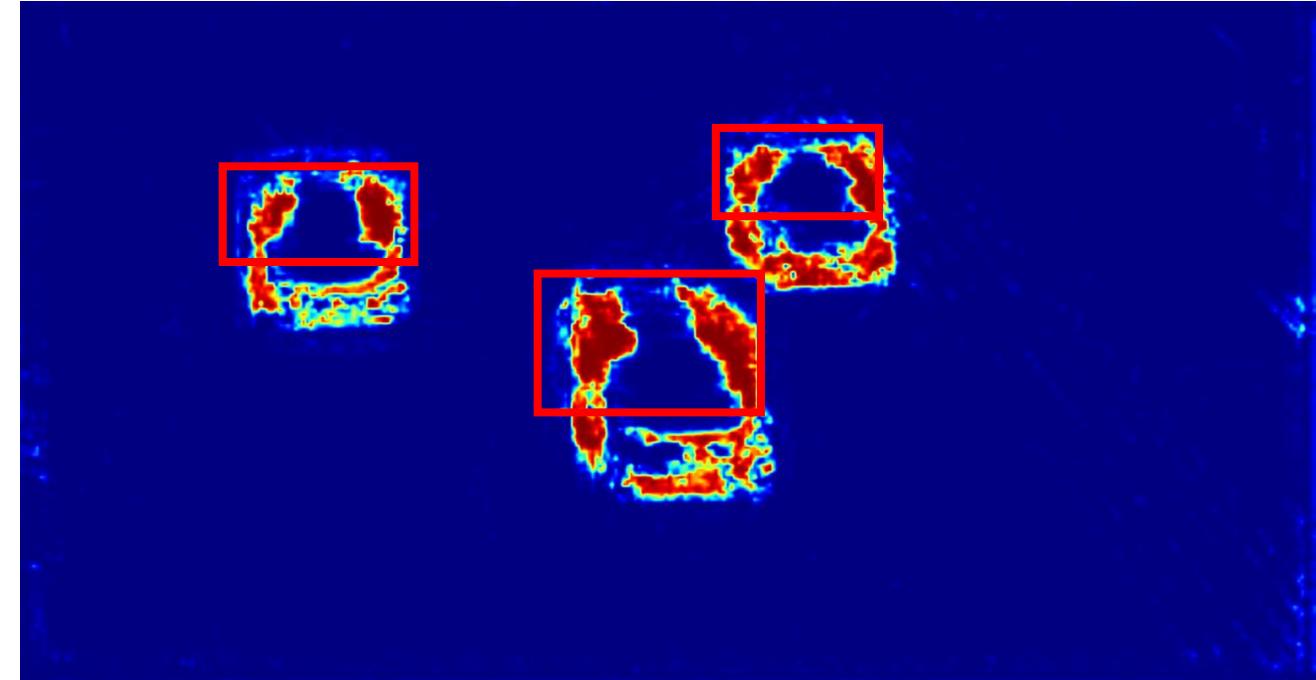


2번 class의 이미지와 2번 class의 이름 간의 매치가 되지 않았습니다.

Test에 실제로 어떤 차량이 나올지 모르기 때문에 모든 모델 연식의 차량을 다 맞출 수 있게 고안하려고 하였습니다.

# [Custom Cutout]

Cutout이 필요한 이유(3) - Mixup 단점

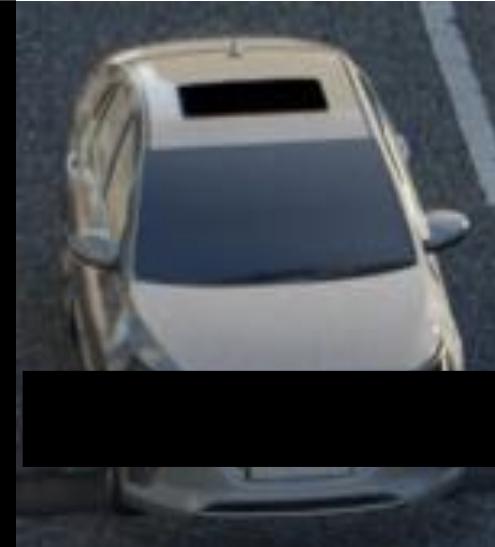


Mixup을 적용하고 난 뒤의 Grad CAM으로 시각화

→ EDA에서 로고, 라이트, 포그램프가 중요하다고 이야기를 하였지만,  
Mixup만으로는 로고, 라이트, 포그램프 쪽에 덜 집중하는 경향을 보입니다.  
저희는 이것이 분류하는데 안좋은 영향을 주었을 것이라고 판단하였습니다.

## [Custom Cutout]

Custom Cutout 적용방식



위와 같은 문제 해결을 위해 Custom Cutout을 제작하여 적용하였습니다.

Custom Cutout은 Bounding Box 안에서 확률적으로 위 그림들 중 하나의 방식을 선택하여 적용합니다.

## [Custom Cutout V1]



물체 간 가려짐의 문제

왼쪽, 오른쪽에 Cutout 적용하여 다른 곳에 집중하도록 합니다.

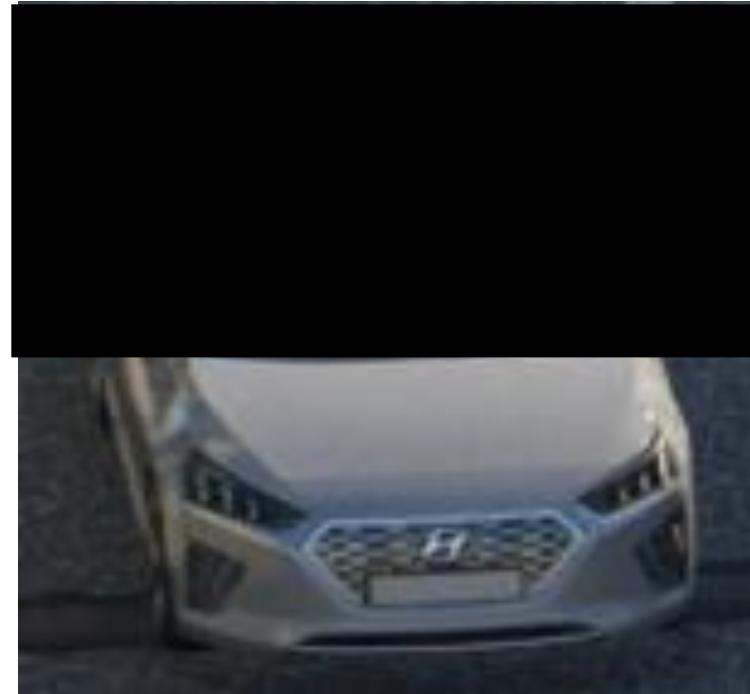
## [Custom Cutout V2]



차량 간 차이 존재

로고, 라이트, 라디에이터 그릴에 대한 집중도를 줄이도록 Cutout 적용합니다.

## [Custom Cutout V3]



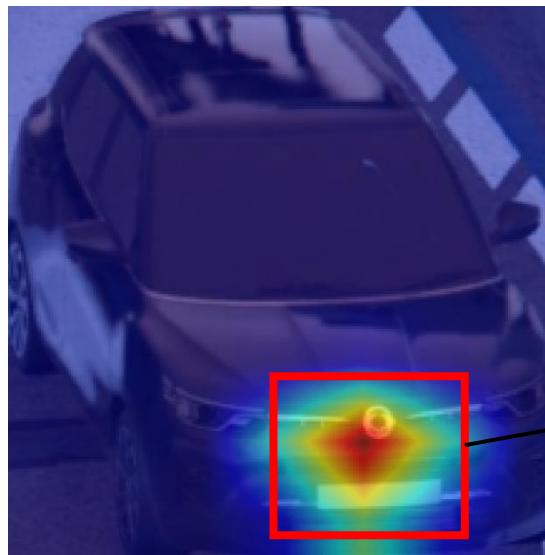
로고, 라이트, 포그램프에 덜 집중하는 문제

위쪽의 이미지에 Cutout을 적용하여  
차량의 프론트 범퍼, 보닛, DRL, 라디에이터 그릴, 헤드램프, 포그램프에 집중을 유도합니다.

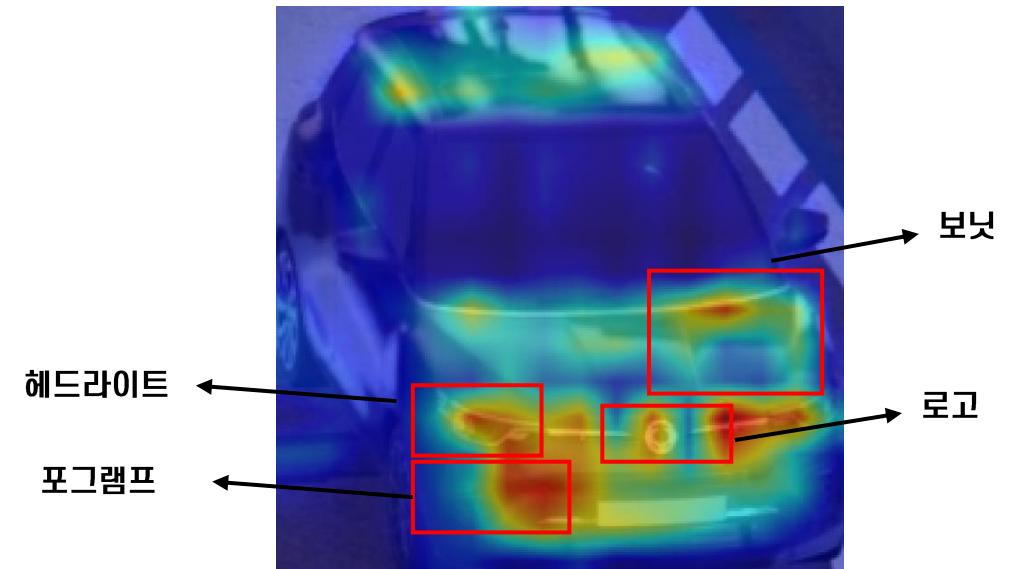
# [Custom Cutout]

Custom Cutout 적용 전/후 Grad CAM 결과

Custom Cutout X



Custom Cutout O



→ Custom Cutout 적용 시 헤드라이트, 포그램프, 그릴, 보닛, 로고 등 다양한 부위를 보고 클래스를 예측합니다.

# Contents

- I. Intro
- II. EDA
- III. 합성 데이터 활용
- ▶ IV. 모델검증
- V. 실험관리

## [모델검증 전략]

사용 가능한 이미지는 모두 Synthetic 데이터,  
하지만 Test set은 현실 이미지라는 간극 존재합니다.  
또한, Train data에 사용된 Augmentation을 Validation Set에 적용한다면  
대부분 잘 예측하게 되어, 모델의 성능을 제대로 평가할 수 없다고 판단하였습니다.  
따라서 저희는, 현실 세계에서 나타날 수 있는 Augmentation을  
Validation 셋에만 따로 적용하였습니다.  
모델이 이를 잘 구별할 수 있다면,  
현실 세계에서 나타나는 다양한 현상들을 잘 반영할 수 있으며,  
모델의 강인함에 대한 신뢰 가능하다고 판단하였습니다.

## [모델검증 전략]

Fog



EDA를 통해 Train 이미지 대부분이 날씨가 맑은 것을 확인하였습니다.

→ 하지만, 현실 세계에서는 안개, 비와 같은 다양한 날씨 조건이 존재합니다.

실제 세계에서 다양한 날씨 조건에 대한 모델의 적응성을 평가하는데 중요하므로,

모델이 학습하지 못했던 Fog와 같은 조건을 Validation Set에 적용하였습니다.

모델의 일반화 능력을 강화하고 새로운 상황에 대한 검증 신뢰도 상승이 가능합니다.

## [모델검증 전략]

Blur



EDA를 통해 Train 이미지는 선명하다는 특성을 발견하였습니다.

→ 이는, 멈춰 있는 자동차가 촬영 될 때의 특성입니다.

움직임으로 인한 모션 Blur를 포함해서 카메라 초점이 흐릿하거나 하는  
현실 세계에서 다양한 상황에 대한 모델의 적응성을 평가하는데 중요하다고 판단하였습니다.

# Contents

- I. Intro
- II. EDA
- III. 합성 데이터 활용
- IV. 모델검증
- ▶ V. 실험관리

# [실험관리]

Model	Backbone	Depth	Augmentation	mAp
Cascade R-CNN	SwinT	-	Mixup, Cutout	0.89
Libra R-CNN	Resnest	200	Mixup, Cutout, AutoAugment, PhotoMetricDistortion	0.89
Faster R-CNN	ResNeXt	101	Mixup	0.91
Faster R-CNN	ResNeSt	200	Mixup	0.93
Faster R-CNN	ResNeSt	101	Mixup, Cutout	0.95
Cascade R-CNN	ResNeSt	200	Mixup, Cutout, AutoAugment, PhotoMetricDistortion	0.98
...	...	...	...	...

→ Resnet계열의 backbone이 가장 성능이 좋았음

→ Mixup과 자체 제작한 CustomCutout을 적용하였을 때 가장 큰 성능 향상이 있었습니다.

## 1. 모델

- mmdetection 라이브러리에서 사용할 수 있는 거의 대부분의 모델을 실험하였으며 최종적으로 Cascade R-CNN을 선정하였습니다.

## 2. Backbone

- Backbone모델은 transformer과 CNN계열을 실험을 해보았습니다. 그 결과 CNN의 성능이 더 좋았고 그 중에서도 resnet 계열의 모델이 가장 우수하였습니다.
- 최종적으로 ResNeSt 200을 backbone 모델로 선정하였습니다.

## 3. Augmentation

- 다양한 Augmentation 기법들을 실험하였습니다.
- Custom Cutout은 차량의 다양한 부위를 하나씩 가려보면서 실험을 하였고 version 3까지 upgrade 하여 최적의 Cutout 방법을 찾아내었습니다.

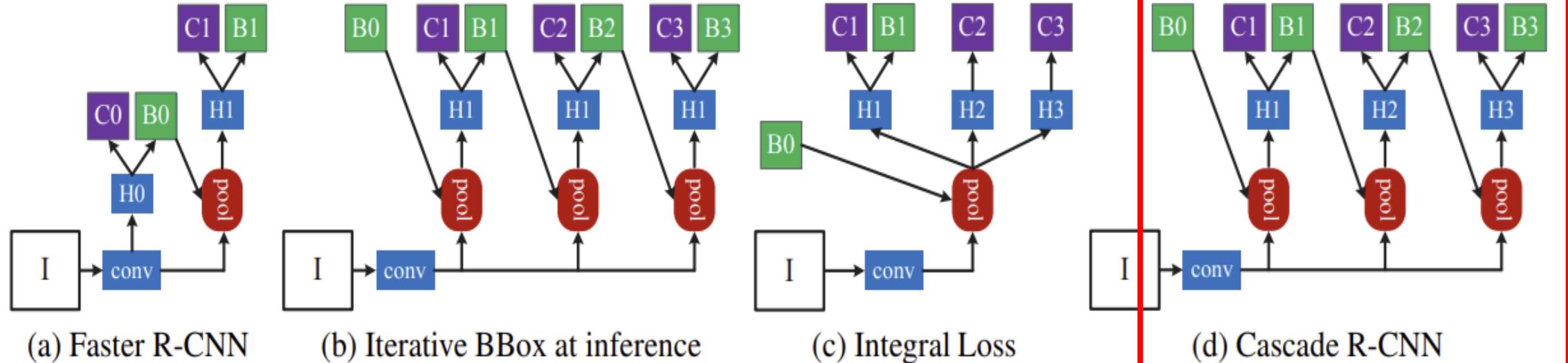


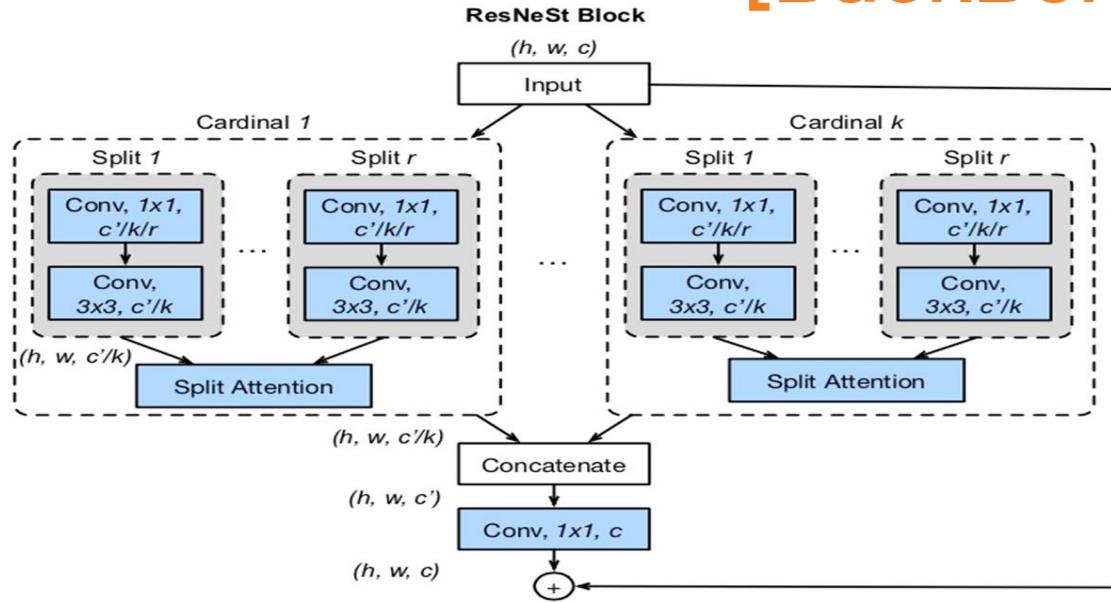
Figure 3. The architectures of different frameworks. “I” is input image, “conv” backbone convolutions, “pool” region-wise feature extraction, “H” network head, “B” bounding box, and “C” classification. “B0” is proposals in all architectures.

저희는 최종적으로 Cascade R-CNN 모델을 선정하였습니다.

Cascade R-CNN은 기존의 Faster R-CNN이 높은 IoU threshold를 가져갈 때, Positive Sample(실제 객체)가 Negative Sample(배경) 수보다 부족하여 성능이 떨어지는 것을 개선한 모델입니다.

Head 부분을 Cascade로 연결하고, 이 때 점진적으로 높은 IoU Threshold를 적용하여 다양한 IoU threshold에 대해 최적화를 할 수 있게 합니다.

# [BackBone : ResNeSt]



실험을 통해 BackBone은 ResNeSt 모델을 선정하였습니다.

## 차량 속성의 다양성 학습 :

다양한 차량 모델은 서로 다른 외관, 크기, 색상 등을 가지고 있습니다.

이러한 특성들은 이미지의 다양한 채널에 정보를 제공하는데,

**ResNeSt의 Split-Attention 메커니즘**을 통해 이런 다양한 채널 간 상호작용을 효과적으로 작용하며 차량 속성의 다양성을 잘 학습할 수 있습니다.

## 다양한 도로 상황 처리 :

다양한 차량 유형, 다양한 도로 상황, 다양한 조명 조건과 같은 여러 환경이 존재합니다. 이런 다양성 때문에 모델이 학습 데이터에 overfitting될 가능성이 높습니다.

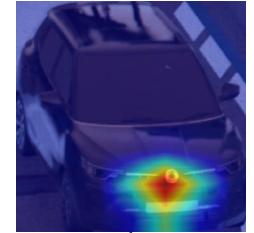
**ResNeSt의 Stochastic Depth 메커니즘**은 이런 overfitting을 방지하고 모델의 일반화 능력을 향상시키는 데 도움이 됩니다.

	Method	Backbone	mAP %
Prior Work	Faster-RCNN [47]	ResNet101 [22]	37.3
		ResNeXt101 [7, 63]	40.1
		SE-ResNet101 [27]	41.9
Our Results	Faster-RCNN+DCN [13]	ResNet101 [7]	42.1
	Cascade-RCNN [4]	ResNet101	42.8
		ResNet50 [60]	39.25
Our Results	Faster-RCNN [47]	ResNet101 [60]	41.37
		ResNeSt50 (ours)	42.33
		ResNeSt101 (ours)	<b>44.72</b>
		ResNet50 [60]	42.52
Our Results	Cascade-RCNN [4]	ResNet101 [60]	44.03
		ResNeSt50 (ours)	45.41
		ResNeSt101 (ours)	<b>47.50</b>
	Cascade-RCNN [4]	ResNeSt200 (ours)	49.03

Object Detection 영역에서 ResNeSt는 Faster R-CNN 및 Cascade R-CNN에서 mAP가 약 3% 높음

# [실험관리 : CustomCutout]

Custom Cutout version.1 : 물체 간 가려짐에 대응하기 위한 Cutout을 적용

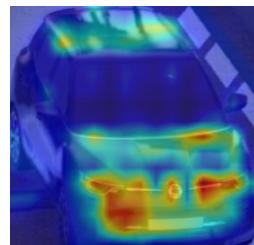


GradCAM으로 Version마다 변화를 관찰함

Custom Cutout version.2 : version.1에서 헤드라이트 및 로고에 대한 집중도를 덜도록 Cutout을 적용



Custom Cutout version.3 : version.2를 개선하여 최종적 가장 다양한 부위를 Cutout하여 version 3을 선택 함.



마치며...

먼저, 좋은 대회로 좋은 데이터를 경험 할 수 있게 기회를 주신  
VISOL과 DACON에 감사드립니다.

쉽게 접할 수 없는 합성 데이터라는 주제를 가지고 대회를 할 수 있어서  
어느 다른 대회보다 의미있는 대회였던 것 같습니다.

미래 지향적으로 저희 팀의 경험과 결과가 앞으로 많은 Resource  
Save와 AI의 발전에 도움이 되었으면 좋겠습니다.

감사합니다 :) !

