2022 데이터 크리에이터 캠프

Data Creator Camp

부부젤라

과학기술정보통신부 NIA 한국지능정보사회진흥원

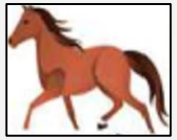
Index

EDA

Data

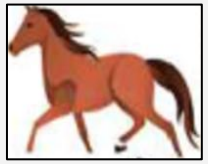
AI Hub

스케치



일러스트

픽토그램



데이터셋에
일러스트만
포함되어 있음

20 Classes

데이터셋에
클래스는 20개가 있음

검은 배경

실사 이미지

잘린 이미지

배경 있는 이미지

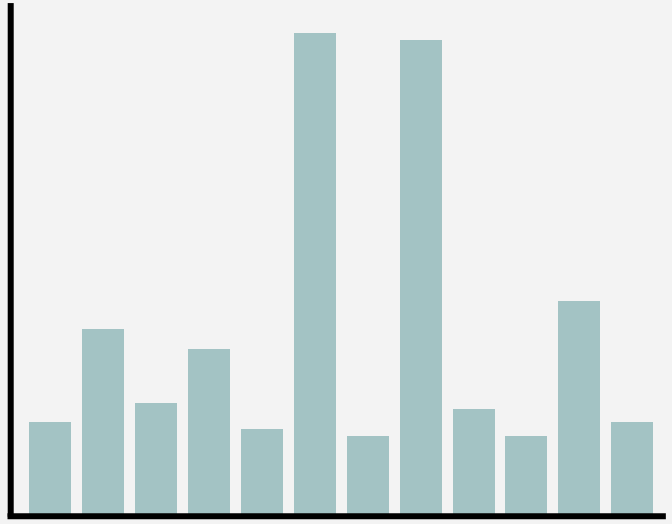
회색 배경

비슷하거나 동일한 이미지

EDA

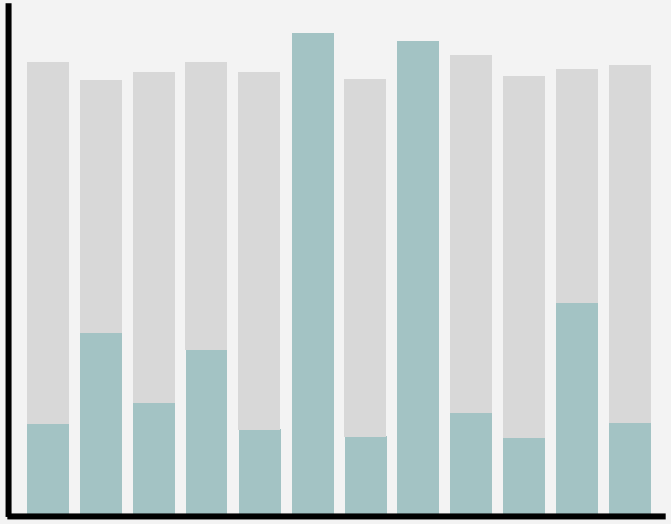
Data Distribution

Original Data Distribution



- 데이터 불균형이 존재
- 다수 클래스에 비슷한 이미지 상당 수 존재

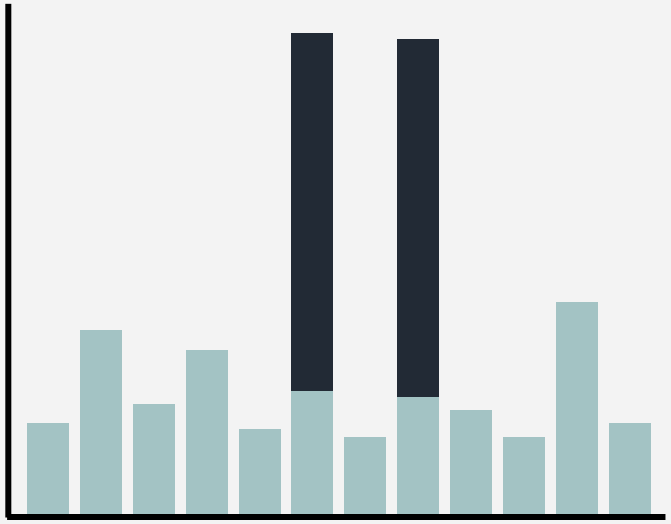
Oversampling



원본
추가된 데이터

- 소수 클래스 Transformation
- 소수 클래스 GAN Augmentation

Undersampling



원본
감소된 데이터

- 다수 클래스 Sampling 적용
- 비슷한 이미지 제거

Unsupervised Classification Illustration or Real image

Objective

K-Means Clustering

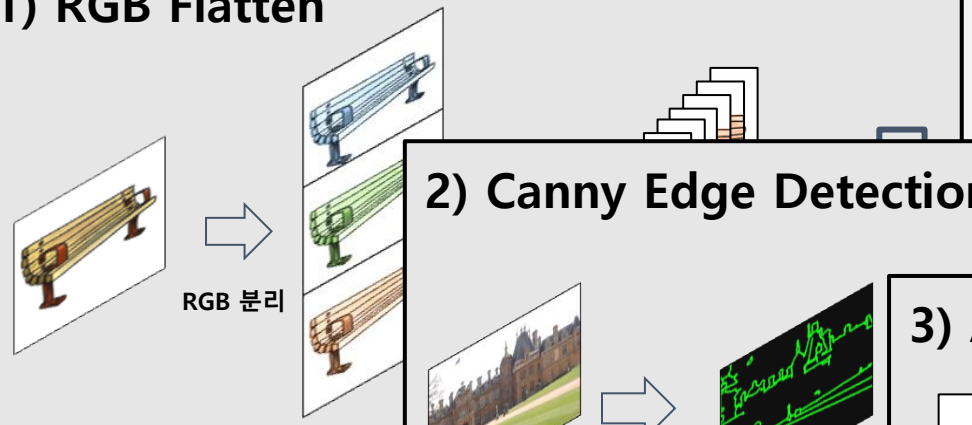
Unsupervised Learning

실사 이미지 제거

Unsupervised Classification Illustration or Real image

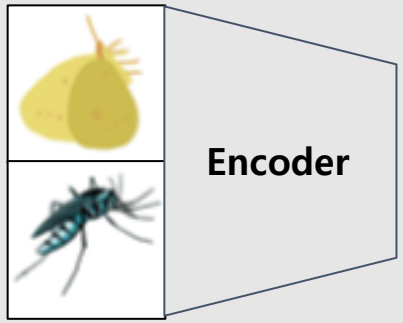
Methods

1) RGB Flatten

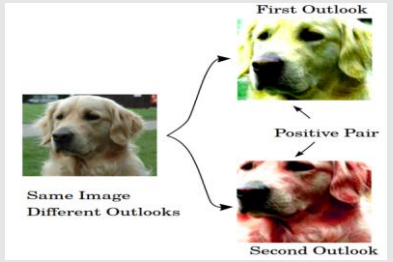


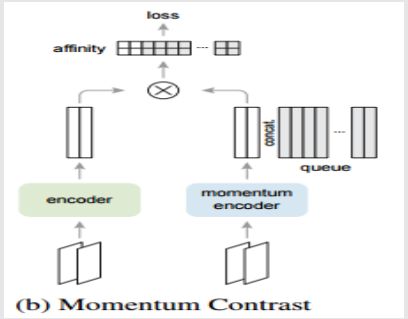
2) Canny Edge Detection

3) AutoEncoder



4) Pre-training Model

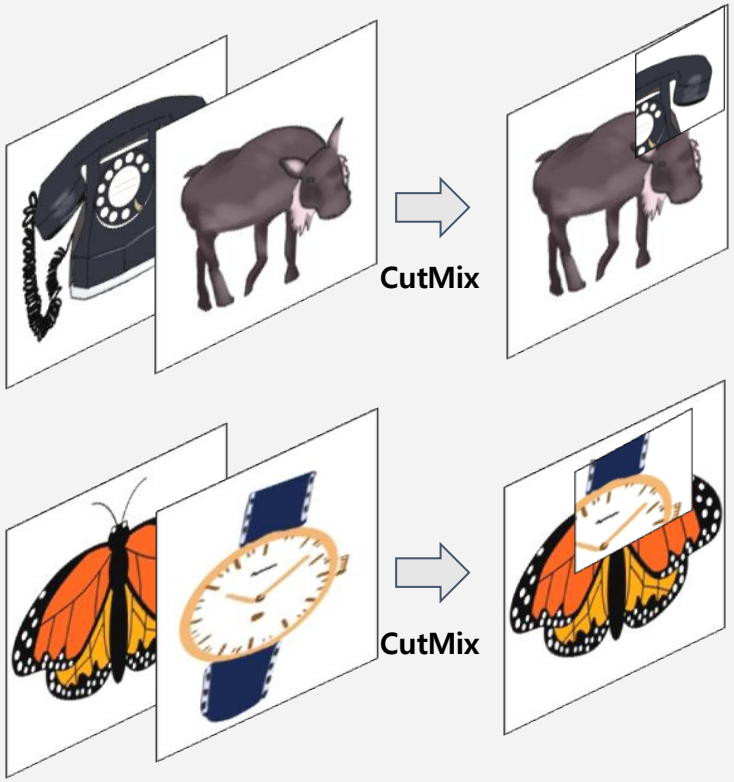




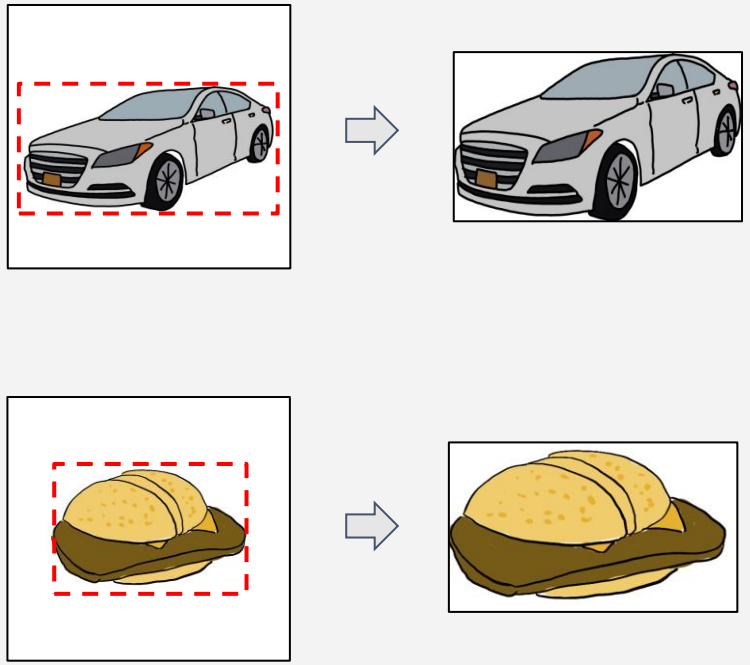
Supervised Classification 20 Class of Illustration Image

Data Preprocessing

Cutmix



Object Crop

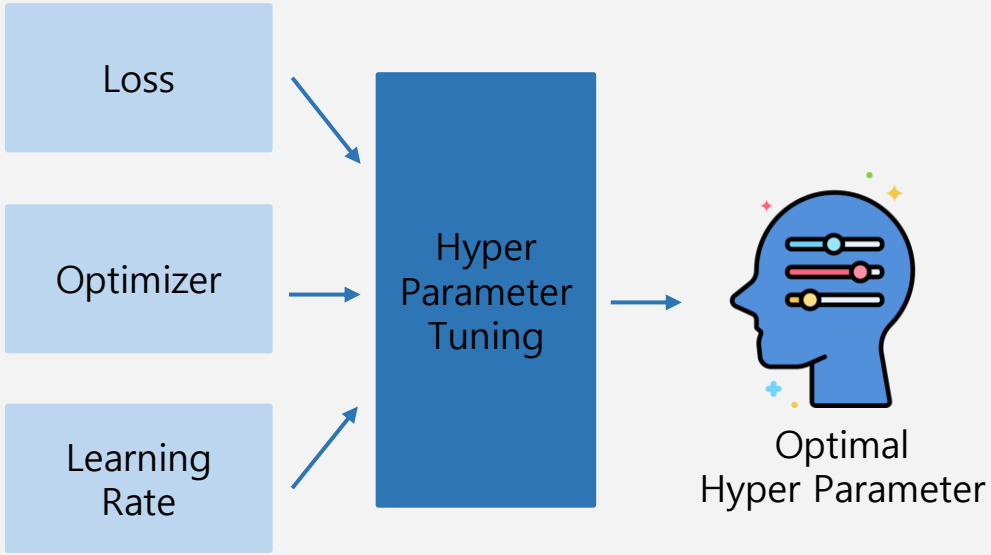


Transformation

Supervised Classification 20 Class of Illustration Image

Model & Parameters

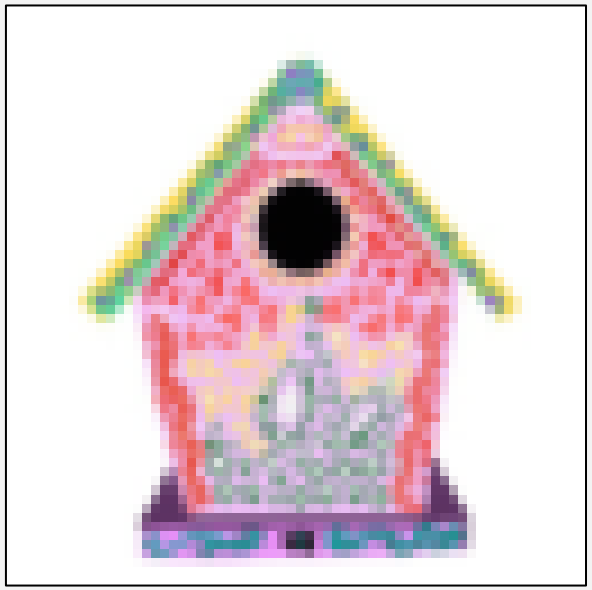
Model Search



Hyper Parameter Tuning

Class Imbalance Oversampling

GAN Augmentation



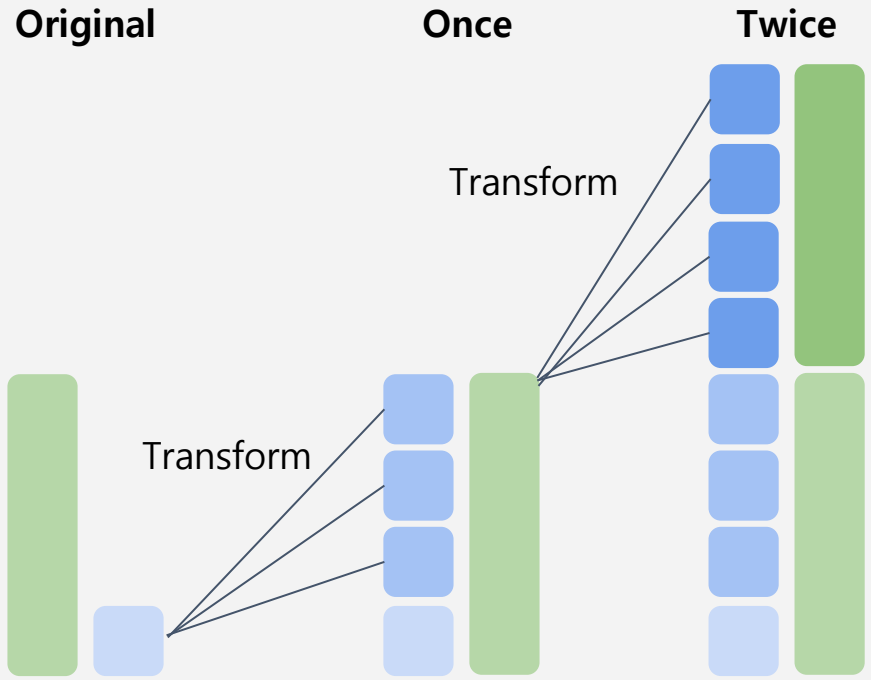
Original

600 epoch

1400 epoch

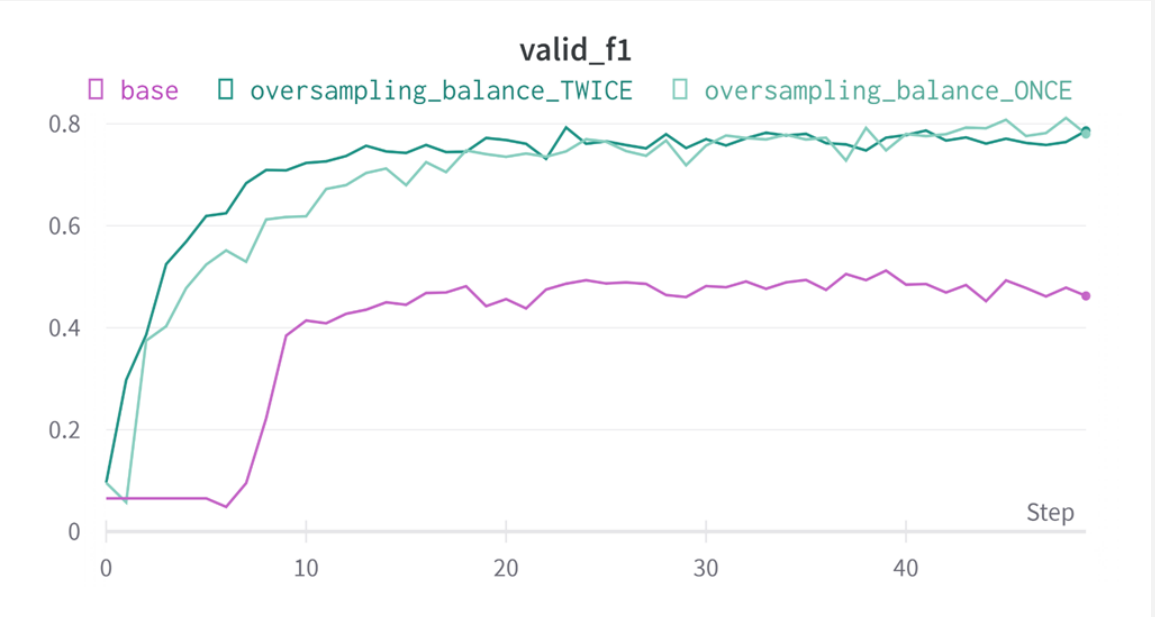
Class Imbalance Oversampling

Transformation with minority class



특정 class에만 transform을 적용하여
오버 샘플링을 했을 때에는 성능이 꽤 준수하게 오름

전반적으로 불균형을 해결하는 것이 성능 향상에 도움이 되었음



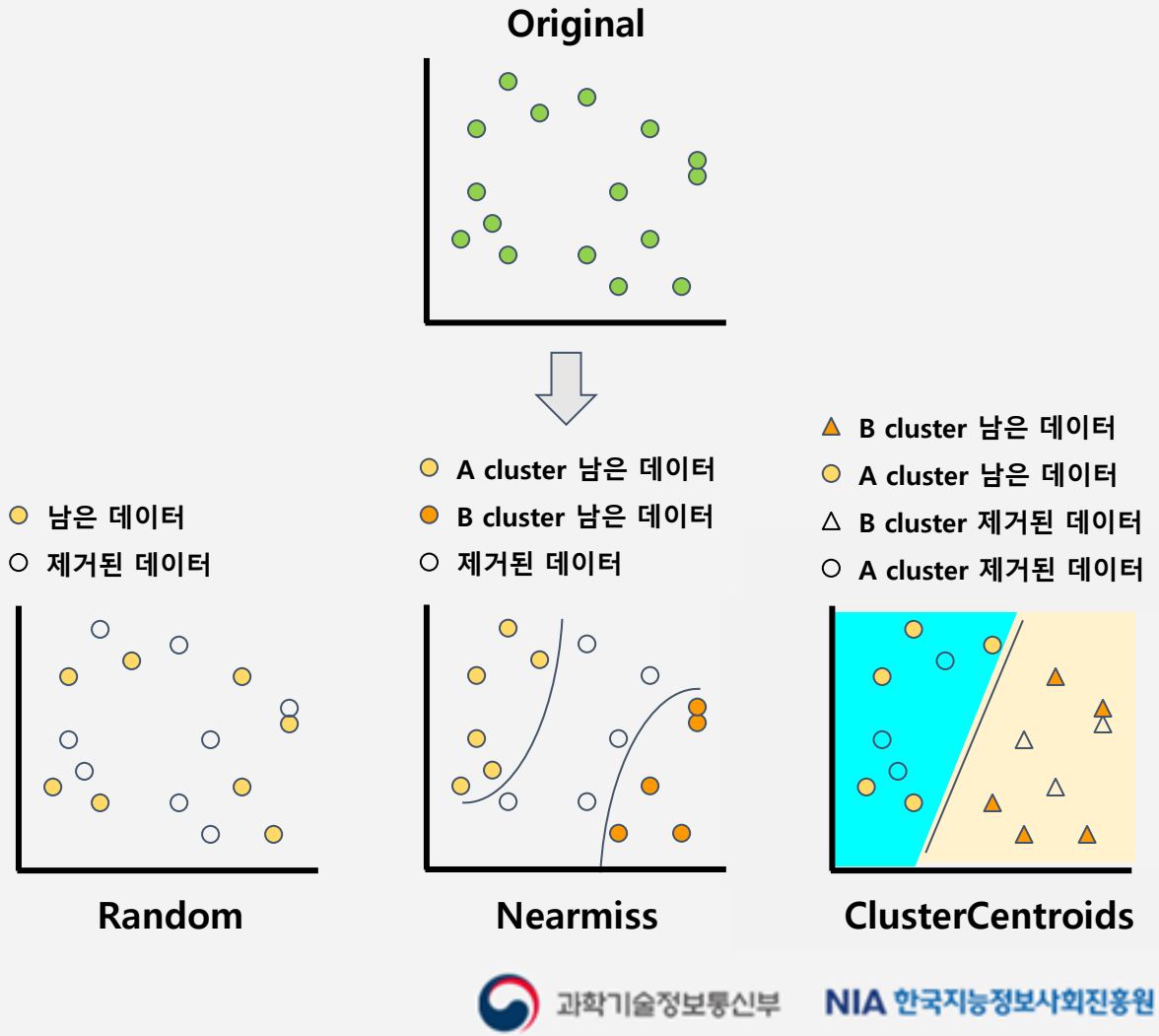
	base	once	twice
F1	0.4623	0.7801	0.7863

Class Imbalance Undersampling

Undersampling

방법	이미지 개수	F1
Original	25,000 개	0.6709
Random	3,600 개	0.1951
ClusterCentroids	3,600 개	0.2864
Nearmiss	3,600 개	0.3707

- 언더샘플링을 사용하지 않았을 때, 가장 좋은 성능을 냈음
- 언더샘플링을 통해 데이터 불균형 문제는 해결할 수 있으나, 이미지 개수가 급격하게 감소되어 모델 성능이 급격하게 감소함



Class Imbalance Undersampling

Discarding Similar Images

23,554 → 23,549

완전히 동일한
이미지 존재

Class Imbalance Undersampling

Discarding Similar Images

23,554 → 23,549

완전히 동일한
이미지 존재

Transform을 통해
생성할 수 있는 이미지 존재

23,554 → 22,207

Class Imbalance Undersampling

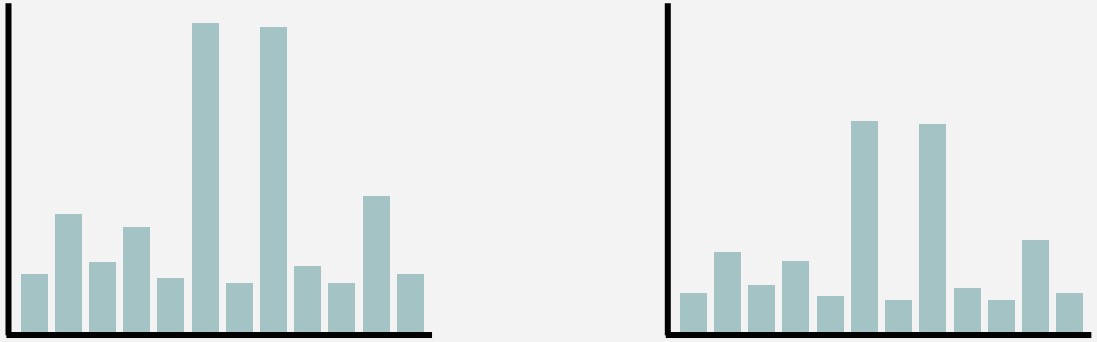
Discarding Similar Images

23,554 → 23,549

완전히 동일한
이미지 존재

Transform을 통해
생성할 수 있는 이미지 존재

23,554 → 22,207



학습할 때 데이터셋에서 Transform을 진행할 텐데,
굳이 비슷한 이미지가 존재해야할까?

Transform을 통해 만들 수 있는 비슷한 이미지를 줄여서
데이터 불균형 문제를 해결해보자

Class Imbalance which way we use?

Comparing Performance

GAN — Not estimated

Transform Minority — 0.8717

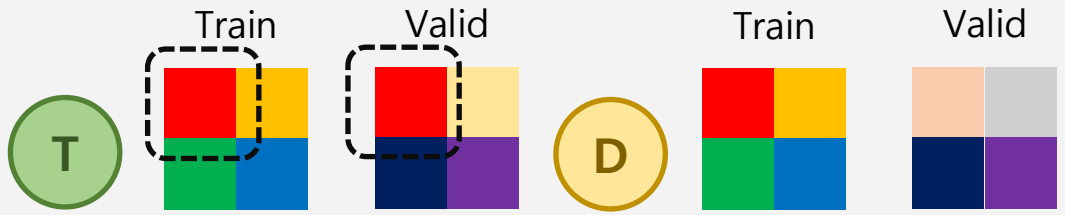
Under sampling — 0.3709

Discard Sim Images — 0.7931

Unfair

1 Inconsistent valid dataset

2 Cheating



Class Imbalance which way we use?

Comparing Performance

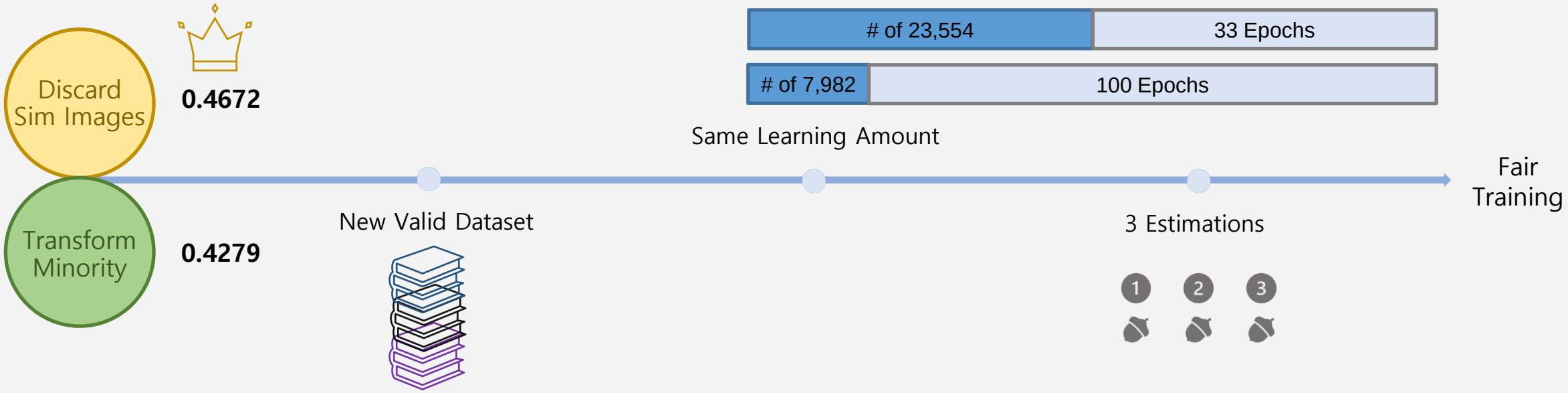
New Valid Dataset

Used for not training, only validation

- 비슷한 데이터가 있을 가능성이 작음
- 동일한 검증 데이터셋 구성

Class Imbalance which way we use?

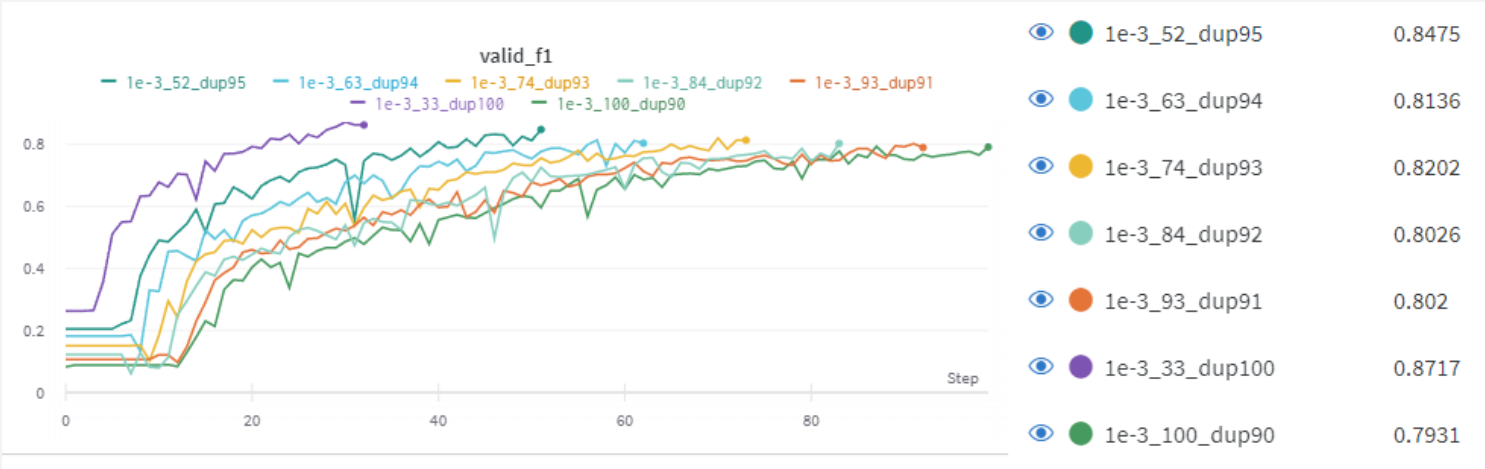
Comparing Performance



Class Imbalance

Discarding Similar Images

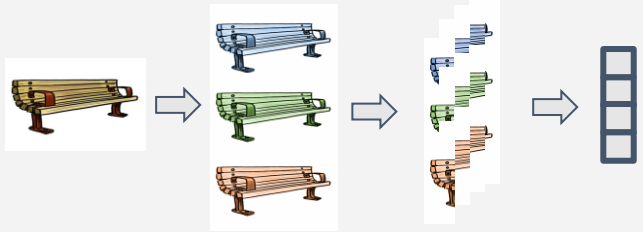
dup	# of images	valid f1	test f1(not schetch)
0.90	7978	0.7931	0.4672
0.91	8605	0.8020	0.4841
0.92	9498	0.8026	0.4831
0.93	10750	0.8202	0.4824
0.94	12648	0.8136	0.4590
0.95	15487	0.8475	0.4430
0.96	19168	x	x
0.97	22207	x	x
0.98	23354	x	x
0.99	23549	x	x
1.00	23549	0.8717	0.4279



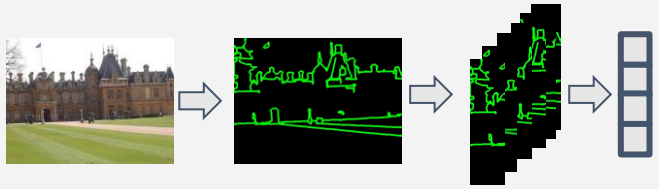
Unsupervised Classification Illustration or Real image

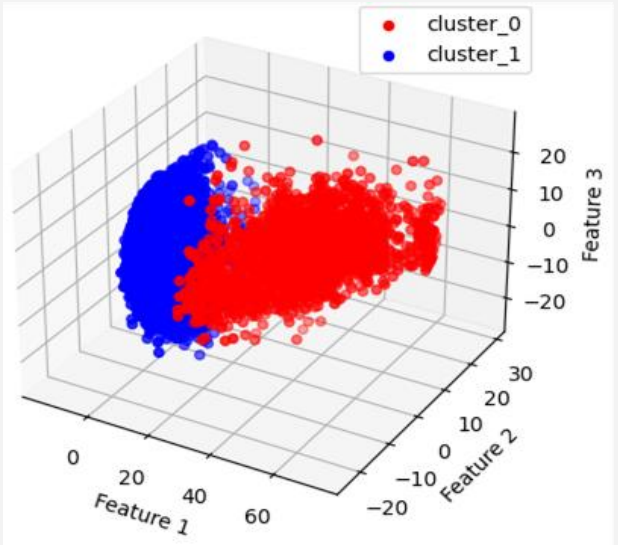
RGB Flatten & Edge Detection

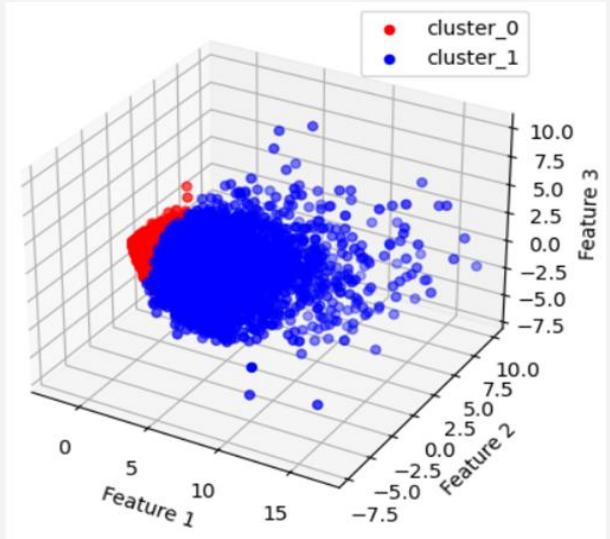
RGB Flatten



Canny Edge Detection







자체적으로 실사와 일러스트 이미지를 구분하여
비지도 클러스터링의 성능을 측정할 수 있도록 함
(해당 정보를 학습에는 일절 사용하지 않았음)

	RGB Flatten	Canny Edge Detection
Precision	0.795	0.336
Recall	0.985	0.842
F1	0.880	0.480

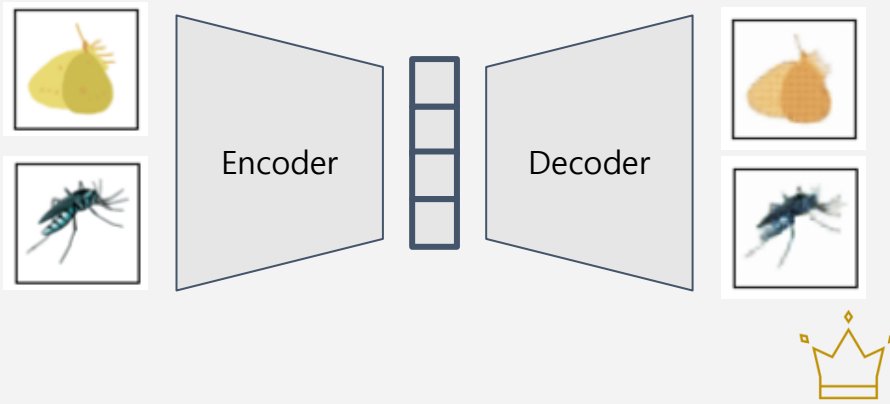
RGB Flatten은 98%의 실사 이미지를 분류했지만,
일러스트 이미지의 20%도 실사 이미지로 분류

Edge Dectection은 84%의 실사 이미지를 분류했지만,
일러스트 이미지도 다수 실사 이미지로 분류

이미지 정보를 Flatten하여 공간 정보를 손실한다는
문제가 있었고 특히 Edge Detection의 경우 검출 자체의
성능도 좋지 않아 Edge 정보를 정확히 담지 못함

Unsupervised Classification Illustration or Real image

AutoEncoder



이미지를 그대로 쓰거나 변환 했을때 보다는
모델을 통해 얻어진 인코딩이 가장 높은 성능을 냈음

Encoder Decoder 레이어가 너무 깊거나 Channel의
개수가 너무 크면 오히려 제 성능을 내지 못했음

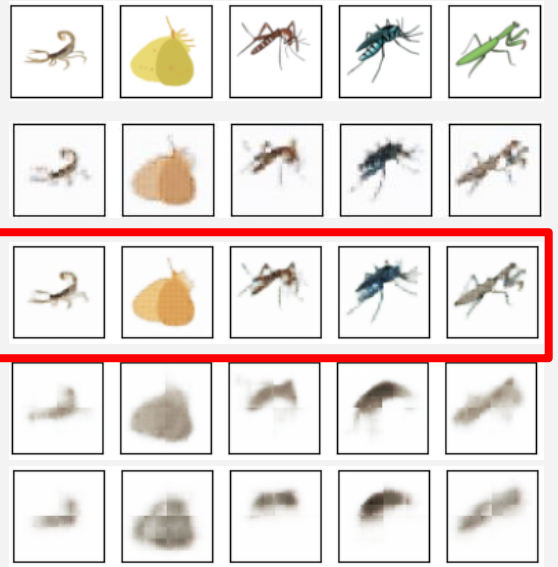
실사 이미지는 99% 분류하지만 일러스트 이미지도
17% 같이 분류한다는 한계가 있음

자체적으로 4가지의 모델을 실험

Enc / Dec	Precision	Recall	F1
CNN 2L / CNN 2L (big channel)	0.791	0.985	0.878
CNN 2L / CNN 2L (small channel)	0.831	0.990	0.904
CNN 4L / CNN 4L	디코딩 결과가 너무 부실하여 따로 평가하지 않음		
CNN 8L / CNN 4L			

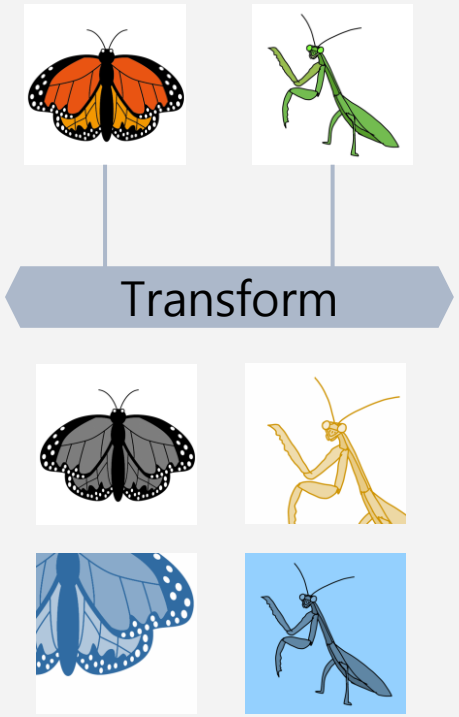
*n-L : n개의 Layer로 구성되었다는 의미

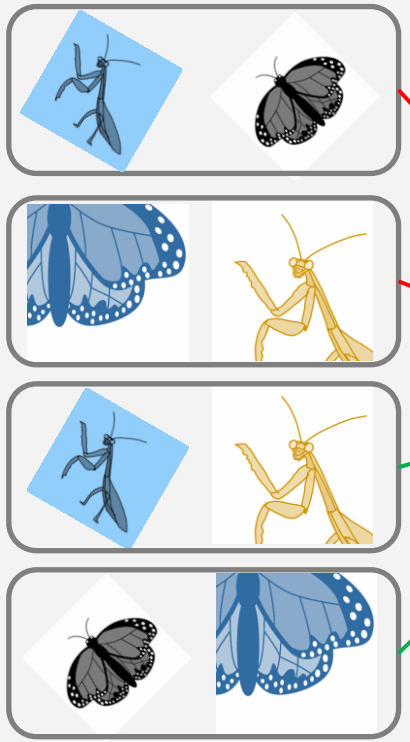
디코딩 결과

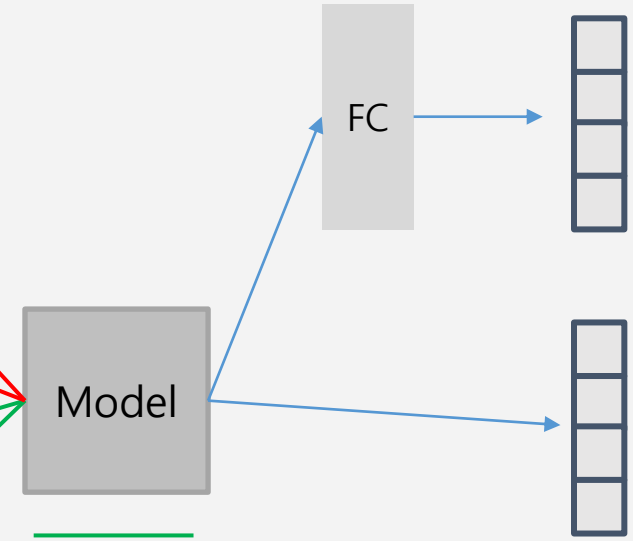


Unsupervised Classification Illustration or Real image

Pre-training







Constrative Learning

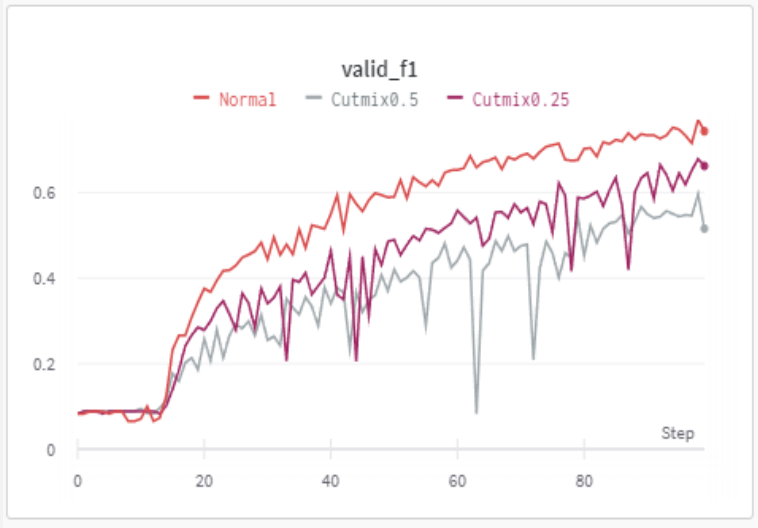
	Model	Model + FC
Precision	0.130	0.122
Recall	0.857	0.758
Found	12660	11973

Pretraining을 거쳐 얻은 모델의 인코딩을 가지고서는 오히려 실사와 일러스트를 더 구분하기 어려워졌음

특히, 이전 방법들은 3천개 정도를 실사로 구분한데 비해 모델의 인코딩은 전체 데이터의 절반 정도인 1.2만여개를 실사로 구분하였음

Supervised Classification 20 Class of Illustration Image

Experiments : Cutmix & Crop

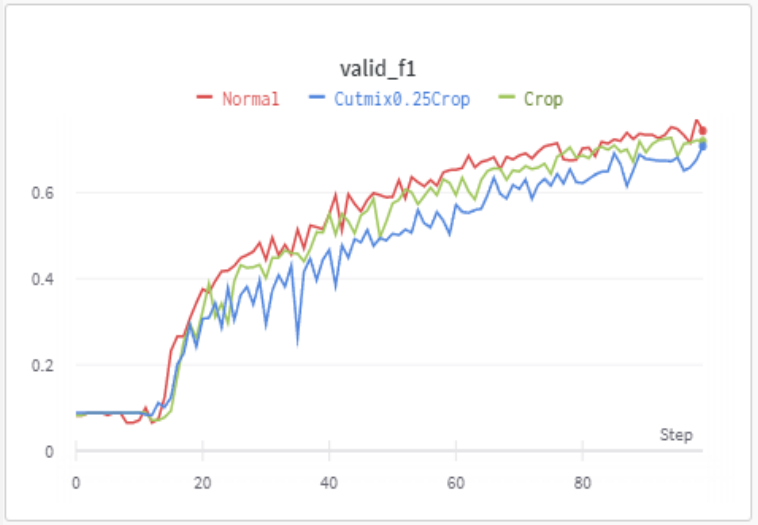


	Normal	Cutmix 0.25	Cutmix 0.5
F1	0.770	0.678	0.596

Cutmix는 오히려 성능을 하락시킴
Cutmix 발생 확률이 높아질 수록
성능이 더 감소하는 모습을 보임

	Cutmix	Cutmix Crop
F1	0.678	0.708

Cutmix와 Crop을 같이 적용할 경우
Object가 아닌 배경을 Mix할 가능성을 줄여
성능이 증가하는 것으로 보임



	Normal	Crop	Cutmix Crop
F1	0.770	0.678	0.596

Crop을 적용한 것보다
적용하지 않았을 때 성능이 더 좋음

Supervised Classification 20 Class of Illustration Image

Experiments : Transform

방법	F1	결과
No Transform	0.5901	기준점
Horizon	0.6843	↑
Vertical	0.5604	
Rotate	0.6096	↑
Perspective	0.7306	↑
Crop	0.2141	
Eras	0.6071	↑
ColorJitter	0.7021	↑
Center Crop	0.2016	

Single Transformation

	F1
No Transform	0.5901
Horizon	0.7427
Rotate	
Perspective	
Vertical	0.5336
Affine	
Eras	

	F1
Horizon	0.7735
Rotate	
Perspective	
ColorJitter	
Vertical	0.6951
Eras	
Perspective	

Multiple Transformation

Supervised Classification 20 Class of Illustration Image

Model Search

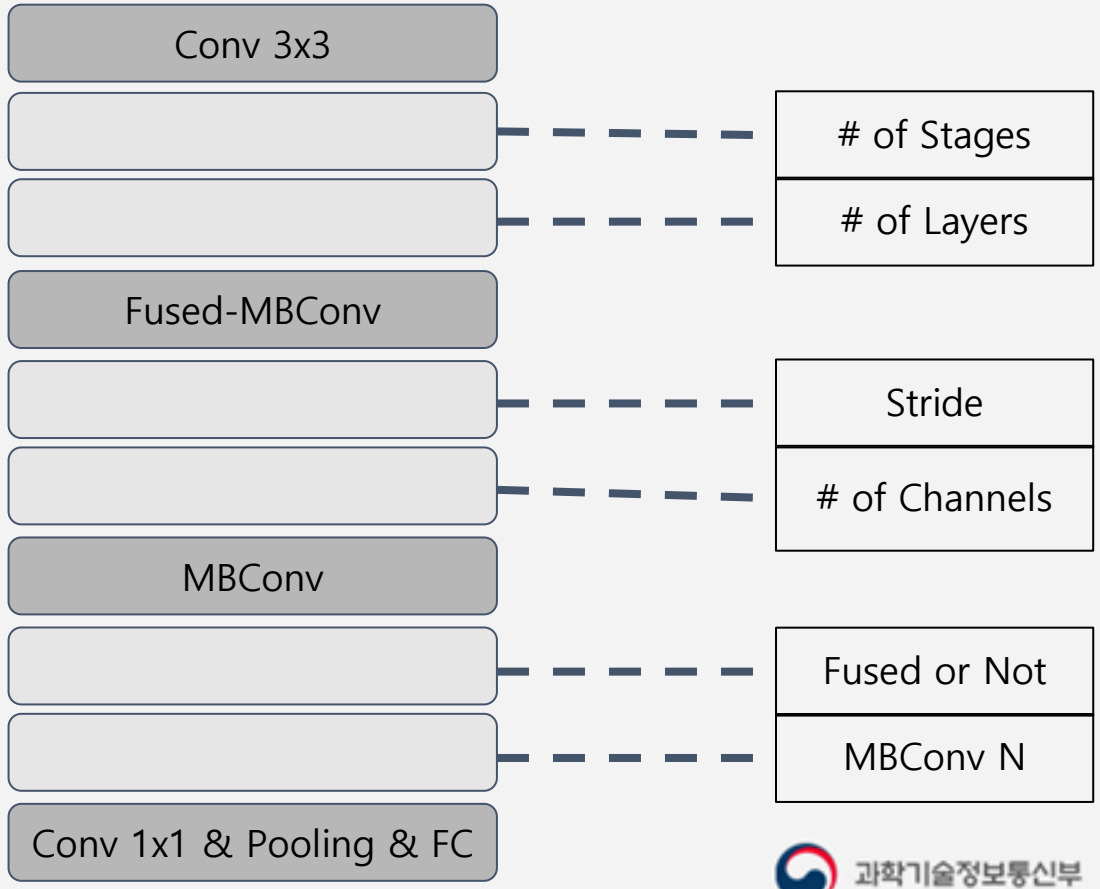
EfficientNetV1

Stage i	Operator $\hat{F}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224 × 224	32	1
2	MBConv1, k3x3	112 × 112	16	1
3	MBConv6, k3x3	112 × 112	24	2
4	MBConv6, k5x5	56 × 56	40	2
5	MBConv6, k3x3	28 × 28	80	3
6	MBConv6, k5x5	14 × 14	112	3
7	MBConv6, k5x5	14 × 14	192	4
8	MBConv6, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

EfficientNetV2

Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

Custom Model



Supervised Classification 20 Class of Illustration Image

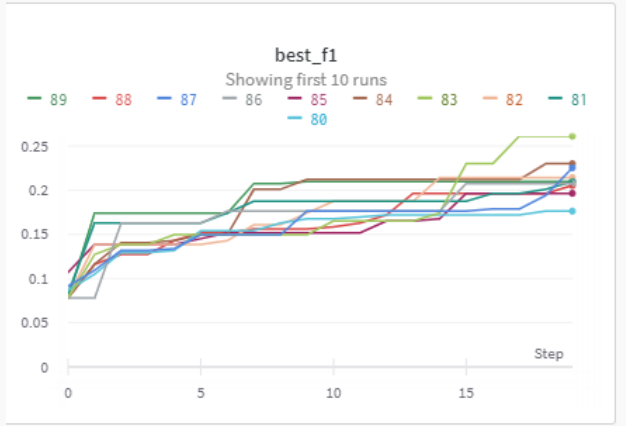
Model Search

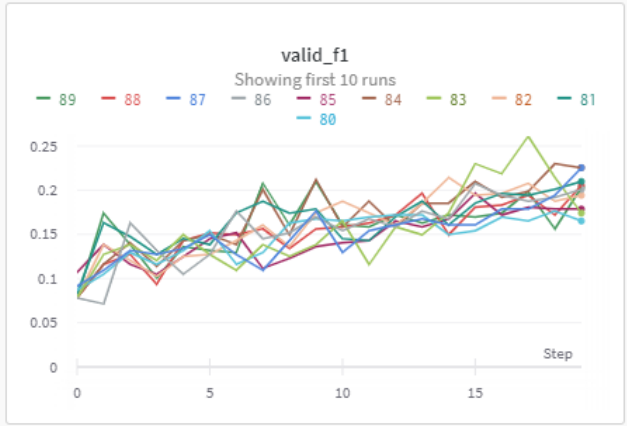
Optuna Parameter Searching

Trial	90
Lr	3e-3
Optimizer	SGD
Epochs	20
Duplicated	0.91
Sampling	around 4000 (balanced)

Consumed Time 33h 20m

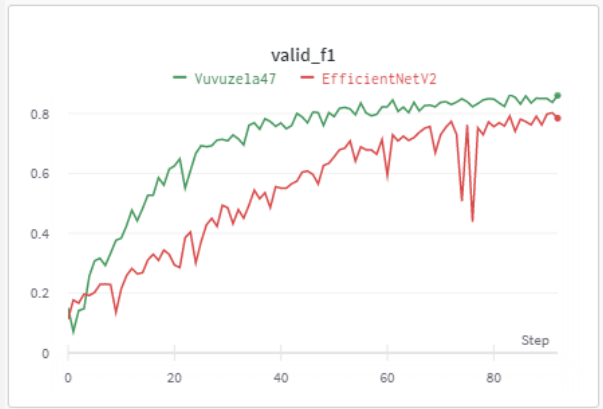
Name (96 visualized)	best_f1	valid_f1
89	0.2098	0.2031
88	0.2054	0.2054
87	0.2254	0.2254
86	0.2076	0.2009
85	0.1964	0.1786
84	0.2299	0.2254





Model	F1
EfficientNet V1 (b0)	0.8020
EfficientNet V2 (s)	0.8027
Candidate 1st	0.8627

93 Epochs Training



과학기술정보통신부 NIA 한국지능정보사회진흥원

https://wandb.ai/sangmandu/automl_search?workspace=user-sangmandu

Supervised Classification 20 Class of Illustration Image

Model Search

Stage	Operator	Stride	#Channels	#Layers
0	Conv 3x3	2	24	1
1	MBConv2, k3x3	1	30	1
2	MBConv4, k3x3	1	42	2
3	MBConv4, k3x3	2	60	4
4	Fuesd MBConv4, k3x3	1	66	4
5	Fuesd MBConv6, k3x3	1	84	4
6	Fuesd MBConv6, k3x3	2	96	6
7	Conv 1x1 & Pooling & FC	-	1792	1

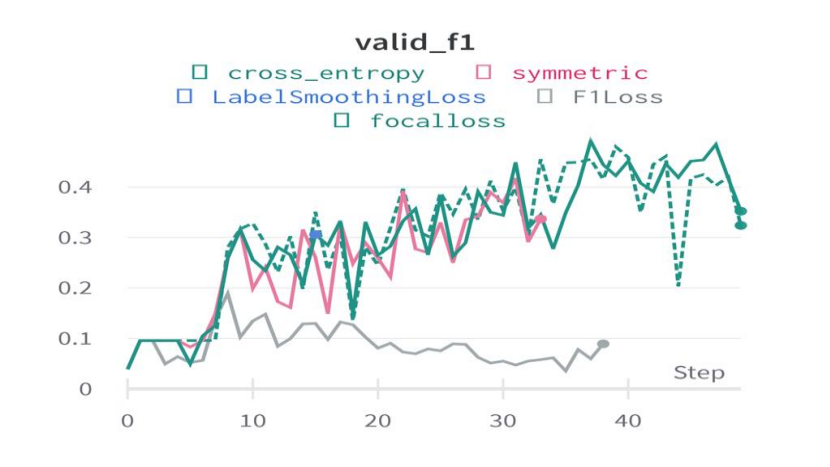
Model	Epochs	F1
Finetuing	93	0.8627
Pre-training + Finetuning	40	0.8843

Vuvuzela 47

Supervised Classification 20 Class of Illustration Image

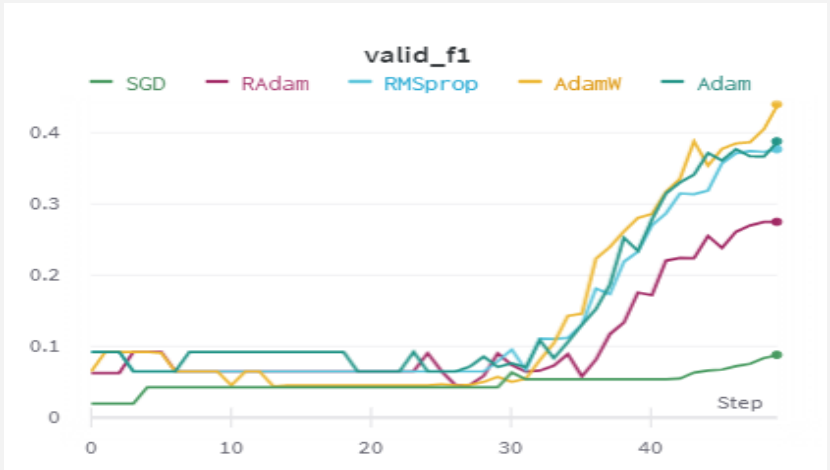
Tuning

Loss



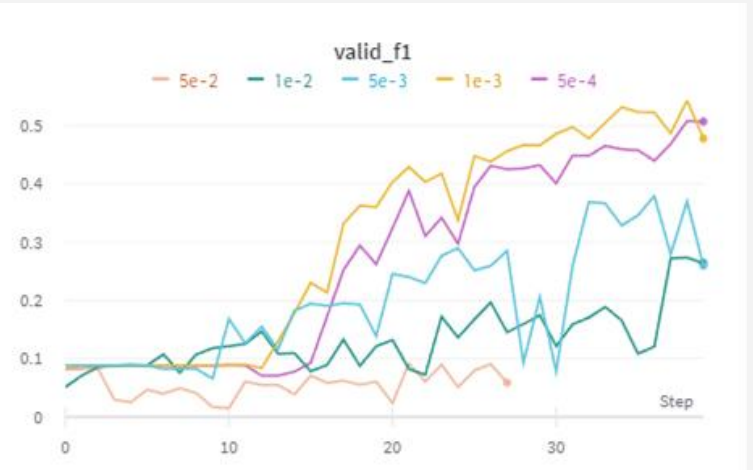
Loss	Cross entropy	symmetric	Label Smoothing Loss	F1 Loss	Focal loss
성능	**0.491**	0.417	0.315	0.188	0.455

Optimizer



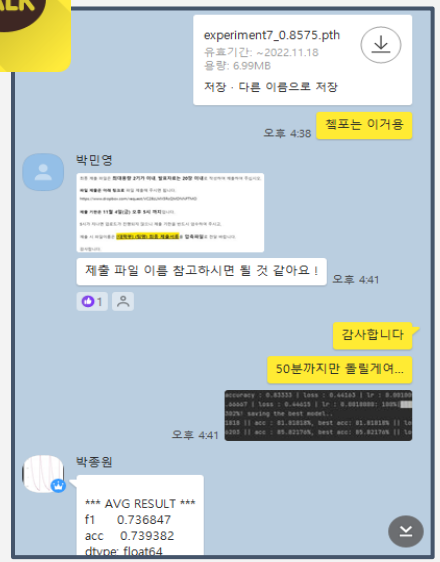
Optimizer	SGD	AdamW	RAdam	RMSprop	Adam
성능	0.088	**0.440**	0.275	0.3769	0.3881

Learning rate

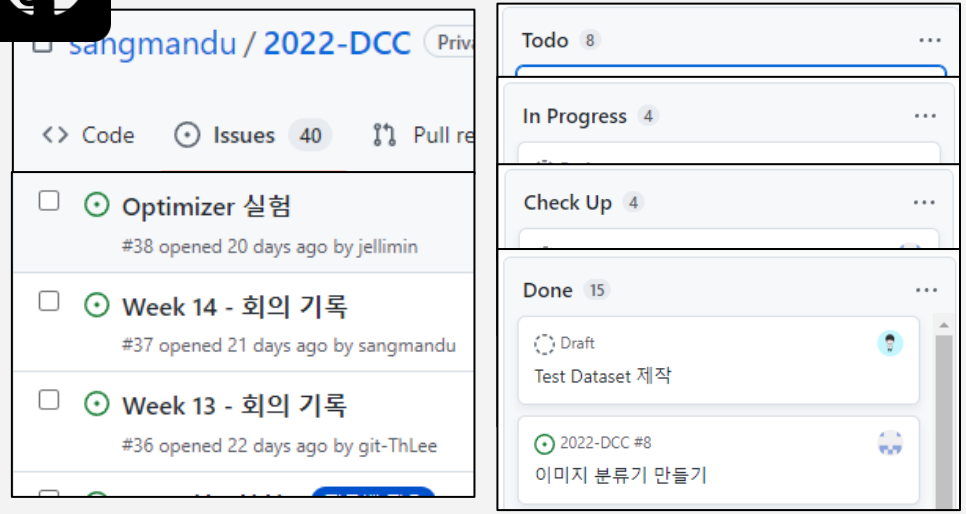


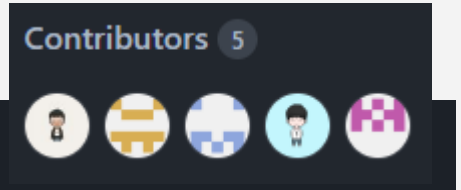
학습률	5e-4	1e-3	5e-3	1e-2	5e-2
성능	0.508	**0.543**	0.379	0.273	0.09

Cooperation



https://github.com/sangmandu/2022-DCC

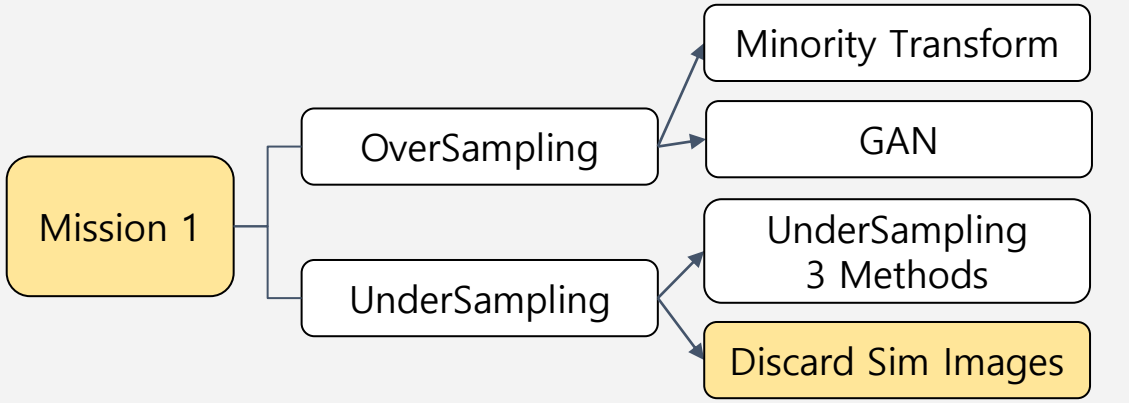


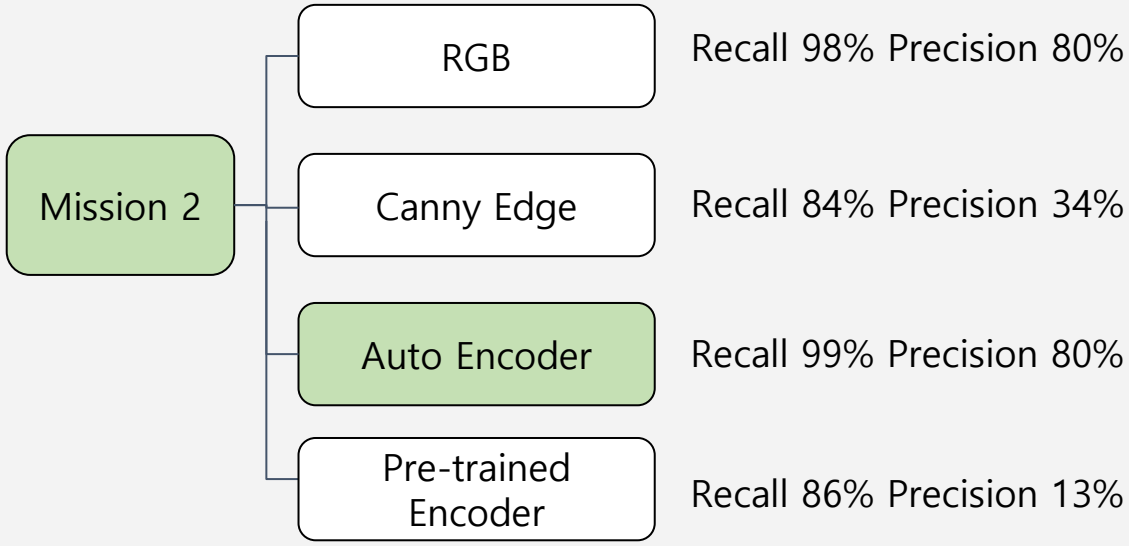


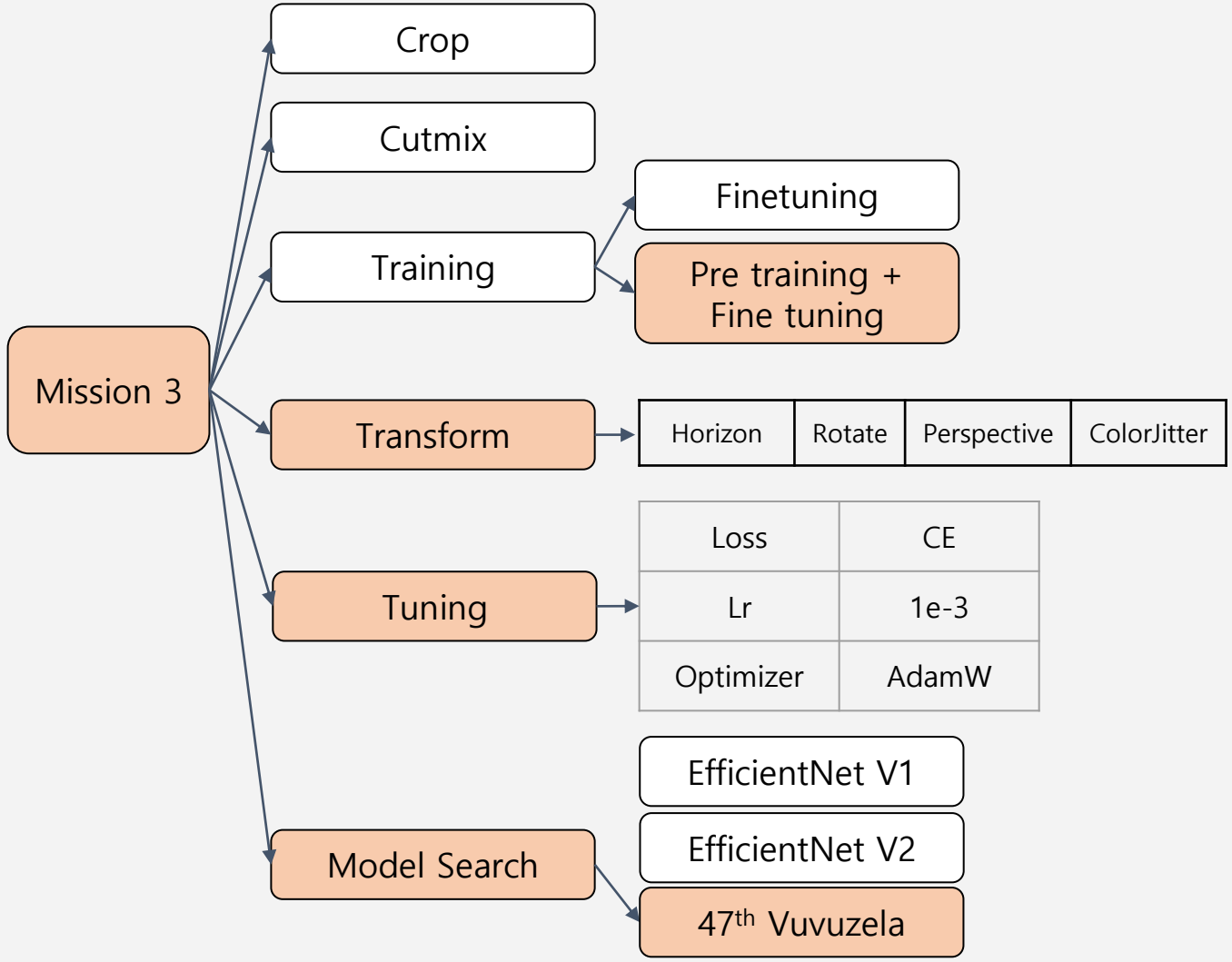
zoom

wandb

Summary


Mission 1
OverSampling
UnderSampling
Minority Transform
GAN
UnderSampling
3 Methods
Discard Sim Images


Mission 2
RGB
Canny Edge
Auto Encoder
Pre-trained
Encoder
Recall 98% Precision 80%
Recall 84% Precision 34%
Recall 99% Precision 80%
Recall 86% Precision 13%


Mission 3
Crop
Cutmix
Training
Finetuning
Pre training +
Fine tuning
Transform
Horizon
Rotate
Perspective
ColorJitter
Tuning
Loss
CE
Lr
1e-3
Optimizer
AdamW
Model Search
EfficientNet V1
EfficientNet V2
47th Vuvuzela

감사합니다

2022 DATA CREATOR CAMP

과학기술정보통신부 NIA 한국지능정보사회진흥원