

Document Type Classification

Content

01. 팀원 소개

02. 대회 소개

03. Data Description

04. Modeling

05. 결과

06. 경진대회 진행 소감

01

팀원 소개

팀원 소개

이름	이현진	권혁찬	김태한	문정의	김소현
역할	조장	조원	조원	조원	조원
전공	산업경영공학과	선박기관시스템 공학과	응용통계학 인공지능 (석사)	전자계산학과	컴퓨터공학과 (석사)
관심사	Computer Vision, NL, 금융 데이터 및 물류 산업 데이터	기계 시스템 고장진단, 시계열 예측, LLM, MLops	Computer Vision, NLP, 추천 시스템 OTT, 문화산업 (책, 웹툰, 영화 등등), 쇼핑 등	ML, 강화학습	Computer Vision, NLP, 최적화, 경량화

02

대회 소개

macro f1 score

		True Class	
		Positive	Negative
Predicated Class	Positive	TP	FP
	Negative	FN	TN

precision : $TP / (TP + FP)$

recall : $TP / (TP + FN)$

f1 score :

$2(\text{precision} * \text{recall}) / (\text{precision} + \text{recall})$

하나의 class에 대한 f1 score를 평균
= macro f1 score

03

Data Description

데이터 설명

1	계좌번호	train 100개 모든 데이터가 흰 백지(또는 이면지)에 자필 이름 / 은행명 / 계좌번호로 이루어져있음 (순서는 랜덤) 이름 : 성 이외의 이름은 비식별화 계좌번호 : 마지막 숫자열의 일부or전부 비식별화
2	건강보험 임신/출산 진료비 지원신청서	train 46개 모두 신청문서 양식을 띄나, 양식이 완전히 일치하지 않음 개인정보는 비식별화 공통적으로 '국민건강보험공단 이사장' 단어 포함 공통적으로 '건강보험 임신·출산 진료비' 단어 포함
3	자동차 계기판	train 100개 다양한 차종의 계기판 대부분 원형적 image feature
4	입·퇴원 확인서	train 100개 개인 의원명은 비식별화 양식 전부 다름
5	진단서	train 100개 양식과 워터마크는 전부 다름 개인정보 비식별화 양식 제목 '진단서' 통일
6	자동차운전 면허증	train 100개 증 사진은 모두 같으나, 스캔여부/ 배경이나 빛반짐, 각도 등 촬영적 요소에서 차이가 있음
7	진료비 계산서·영수 증	train 100개 양식이 모두 다르며, 인쇄 방향 또한 서로 상이한 경우가 있음 개인정보 비식별화
8	진료확인서	train 100개 제목이 '진료확인서' or '통원확인서'로 구성 (통일 x)
9	주민등록증	train 100개 운전면허증과 비슷한 특징, but 주민등록증과 운전면허증의 양식 차이만 존재
10	여권	train 100개 데이터마다 스캔여부, 촬영방식이 모두 다름

11	진료비 납입 확인서	train 100개 '진료비 납입 확인서' or '진료비(약제비) 납입 확인서'로 양식 제목이 나뉨 기타 내용양식은 서로 상이함 모두 세로로 출력되었으나, 스캔여부가 상이
12	약제비 영수증	train 100개 POS기에서 뽑은 영수증, 약봉투에 적힌 영수증, A4 영수증 등 가장 데이터마다 양식이 상이함 약봉투에 적힌 영수증은 약 봉투 전체가 스캔된 데이터와 영수증 부분만 따로 찍은 데이터도 존재 특히, 영수증만 따로 찍은 데이터는 배경이 전체 데이터의 절반이상이므로 이런 데이터에 대한 전처리의 유무 또한 중요해보임
13	처방전	train 100개 A4 세로로 출력된 영수증 데이터로 통일 그러나 스캔 여부, 문서의 손상(접힘, 구겨짐 등)이 심함 양식 제목이 '처방전'으로 통일되어있음
14	이력서	train 74개 A4 세로로 출력된 데이터로 통일 양식 제목이 '처방전'으로 통일되어있으나, 제목 부분의 서식이 제각각 데이터 중 일부는 이력서 양식의 둘 째 장인 데이터도 존재 (양식 제목이 없음)
15	소견서	train 49개 A4 세로로 출력된 데이터로 동일 양식제목이 '소견서' '진료소견서' 등 통일되어있지 않음 이외에 의료기관과 같은 정보가 포함되어 있으나, 비식별화 양식에 '병명'을 적는 부분 존재
16	자동차 등록증	train 100개 A4 세로로 출력된 데이터이나, 일부 데이터가 회전되어있음 양식 제목이 '자동차 등록증'으로 통일, 양식제목의 font도 통일 다만 자세한 양식은 관찰시에 따라 다른 것으로 보임
17	자동차 번호판	train 100개 자동차의 정면에서 찍은 데이터 차종은 상이하지만, 1개의 데이터를 제외한 번호판이 모두 신식 번호판 (흰색 번호판) 일부 데이터는 앞 3자리 숫자 번호가 채워진 데이터 (흰색 번호판)

Data Augmentation strategy - (1)

1. Augmentation type	<ul style="list-style-type: none">- <u>Offline Augmentations</u> 방법 사용
2. Details	<ul style="list-style-type: none">- 학습 데이터의 수 자체가 많지 않고 테스트에 Flip 되거나 mixup된 데이터 등 다양한 데이터가 많음- 이를 augmented data를 많이 만들어서 robust하게 학습하도록 유도- 그외에도 <u>224x224로의 Resize</u>와 baseline 코드의 Noramlize 사용
3. Options	<ul style="list-style-type: none">- <u>Albumentation</u> 사용- HorizontalFlip , VerticalFlip, ShiftScaleRotate,- Grayscale, ColorJitter, Blur, MedianBlur, Spatter, Defocus, ZoomBlur,- OpticalDistortion 2장, Perspective 2장 , Rotate 2장

Data Augmentation strategy - (2)

1. Augmentation type	<ul style="list-style-type: none">- <u>Offline + Online Augmentations</u> 방법 사용
2. Details	<ul style="list-style-type: none">- 테스트 데이터와 유사한 학습 데이터를 Albumentation을 사용하여 증강하였음- 데이터 갯수는 <u>class당 약 300개 데이터로 생성</u>하였음- Albumentation적용 옵션은 VerticalFlip, Rotate, HorizontalFlip, RandomBrightnessContrast, GaussianBlur, PixelDropout, ShiftScaleRotate, ShiftScaleRotate 사용- <u>Augraphy</u> 옵션을 사용하여 신규 생성되는 이미지 데이터의 50%에 적용
3. Options	<p><u>Albumentation적용 / Online Augmentations</u></p> <ul style="list-style-type: none">- AdvancedBlur, ColorJitter, GaussNoise, OpticalDistortion, Affine, HorizontalFlip, ElasticTransform

Data Augmentation strategy - (3)

1. Augmentation type	<ul style="list-style-type: none">- <u>Offline Augmentations</u> 방법 사용
2. Details	<ul style="list-style-type: none">- 학습 데이터의 수 자체가 많지 않아 메모리에 부담이 될 것 같지 않아 데이터를 시각화하기 위해 <u>offline augmentation</u> 사용- <u>데이터를 12배 증강</u> 후, <u>augmentation</u>을 통해 서로 다른 데이터로 변환- 추가적으로 <u>256x256로의 Resize</u>와 <u>Normalize</u> 사용. (Online)
3. Options	<p><u>Albumentation</u> 적용</p> <ul style="list-style-type: none">- HorizontalFlip, VerticalFlip, RandomRotate90- ShiftScaleRotate, RandomBrightnessContrast, GauissNoise- Oneof(Motion Blur, Medianblur, Blur)- Oneof(OpticalDistortion, GridDistortion, PiecewiseAffine)- HueSaturationValue

Data Augmentation strategy - (4)

1. Augmentation type	<ul style="list-style-type: none">- <u>Online Augmentations</u> 방법 사용
2. Details	<ul style="list-style-type: none">- 기존에 주어진 데이터를 프로그램에서 불러온 뒤 Albumentation을 사용하여 augmentation- <u>실험한 모델들에 따라 다른 Resize 적용</u>
3. Options	<div><u>Albumentation</u>사용</div> <ul style="list-style-type: none">- Resize, RandomRotate90, ShiftScaleRotate- Flip, GaussNoise, RandomBrightnessContrast, HueSaturationValue- Normalize <div><u>선택 적용</u></div> <ul style="list-style-type: none">- Blur, MotionBlur, MedianBlur- OpticalDistortion, GridDistortion, PiecewiseAffine- CLAHE, Sharpen, Emboss

04

Modeling

Model Select

Hugging Face 모델

Model Stats	efficientnet_b5 (sw_in12k_ft_in1k)	tf_efficientnet_b6 (ns_jft_in1k)	tf_efficientnet_b7 (ns_jft_in1k)	비고
Params (M)	30.4	43.0	66.3	
GMACs	9.6	19.4	38.3	
Activations (M)	93.6	167.4	289.9	
Image size	448 x 448	528 x 528	600 x 600	

Model Stats	efficientnet_b5	CoAtNet	convnext_large	tf_efficientnetv2_m	HRNet
Params (M)	52.6	27.4	200.1	54.1	77.5
Image size	416 * 416	384 * 384	224 * 224	224*224	224*224
비고	다른 competition 등에서 활용도가 높고 성능도 좋은 모델 선택	데이터셋이 크지 않아 상대적으로 작은 모델로 학습 시도	상대적으로 큰 CNN 모델을 선택하여 학습, 잘 구분하지 못했던 클래스에 대해 더 나은 classification 성능 기대	실험의 성능이 제일 좋았던 <u>efficientnet</u> 의 상위버전인 v2 선택	<u>해상도가 영향</u> 이 있는지 알기 위해 화질에 따른 feature를 추가하는 network 선택

* vit_base_patch16_clip_384 모델을 적용하려고 하였으나, 학습이 되지 않아서 제외함

Various Models with Trial and Error

김태한

Model	Optimizer	Loss	Augmentation	Add. condition	Local F1 (macro)	Public Score
EfficientNet_v2_m	Adam (1e-3)	Cross Entropy	16 Offline Augmentations <ul style="list-style-type: none">HorizontalFlipVerticalFlipShiftScaleRotateGrayscaleColorJitterBlurMedianBlurSpatterDefocusZoomBlurOpticalDistortion 2장Perspective 2장Rotate 2장	<ul style="list-style-type: none">이미지 하나당 16개의 augmentation.Transfer Learning.	0.7874	미제출
EfficientNet_v2_m	Adam (1e-3)	Cross Entropy		<ul style="list-style-type: none">이미지 하나당 16개의 augmentation.Fine-Tuning.	0.9958	0.8110
Two Stage: Two EfficientNet_v2_ms	Adam (1e-3)	Cross Entropy		<ul style="list-style-type: none">이미지 하나당 16개의 augmentation.Fine-Tuning.첫 번째 EfficientNet_v2_m으로 <u>자동차계기판, 자동차번호판, 문서를 분류.</u>두 번째 EfficientNet_v2_m으로 나머지 <u>15가지의 문서를 분류.</u>	1st: 1.0000 2nd: 0.9967	0.6538
EfficientNet_v2_m	Adam (1e-3)	Focal Loss + Cross Entropy		<ul style="list-style-type: none">이미지 하나당 16개의 augmentation.Fine-Tuning.	0.9819	0.7789

Various Models with Trial and Error

문정의

Model	Optimizer	Loss	Augmentation	Add. condition	Local F1 (macro)	Public Score
efficientnet_b5	Adam (1e-5)	CrossEntropy	offline - VerticalFlip, Rotate, HorizontalFlip, RandomBrightnessContrast, GaussianBlur, PixelDropout, ShiftScaleRotate, ShiftScaleRotate, Augraphy online - AdvancedBlur, ColorJitter, GaussNoise, OpticalDistortion, Affine, HorizontalFlip, ElasticTransform	- 이미지 데이터 4배 증가 - epochs : 30 - StratifiedKFold : 5fold	1.0000	0.8821
tf_efficientnet_b6					1.0000	0.9008
tf_efficientnet_b7					1.0000	0.8866
efficientnet_b5	Adam (1e-5)	CrossEntropy	- 상동	- 이미지 데이터 3배 증가 - 잘못된 이미지 라벨링 수정 (6건)	1.0000	0.9135
tf_efficientnet_b6					1.0000	0.9285
tf_efficientnet_b7					0.9910	0.9133
Ensemble						0.9303
efficientnet_b5	Adam (1e-5)	CrossEntropy	- 상동	- 상동	1.0000	0.9168
tf_efficientnet_b6					1.0000	0.9168
tf_efficientnet_b7					1.0000	0.9152
Ensemble						0.9303

Various Models with Trial and Error

이현진

image size = 256 x 256

Model	Optimizer	Loss	Augmentation	Add. condition	Local F1 (macro)	Public Score
ResNet50	Adam (1e-4)	CrossEntropy	-	data size 증강 (3배)	0.9556	0.5085
EfficientNet_v2_s	Adam (1e-4)	CrossEntropy	Flip, RandomRotate90, Blur, ShiftScaleRotate, RandomBrightnessContrast, GaussNoise	data size 증강 (6배) + labeling 이상한 데이터 삭제 (9건)	0.9812	0.6234
EfficientNet_v2_m	Adam (1e-4)	CrossEntropy	이전 Augmentation + CLAHE/Sharpen/Emboss	+ 상동 + data size 증강 (10배)	0.9926	0.7402
EfficientNet_v2_s	Adam (1e-4)	Focal loss	이전 Augmentation + HueSaturationValue	상동 + data size 증강 (10배)	0.9857	0.8214
ResNet50	Adam (CosineAnnealingWarmRe starts, 1e-4 ~ 1e-6)	Focal loss	상동	상동	0.9898	0.8205
tf_EfficientNet_v2_n s	Adam (CosineAnnealingWarmRe starts, 1e-4 ~ 1e-6)	Focal loss	이전 Augmentation + MixUP	상동 + data size 증강 (12배)	0.9921	0.7781

Various Models with Trial and Error

김소현

Model	Optimizer	Loss	Augmentation	Add. condition	Local F1 (macro)	Public Score
efficientnet_b5	Adam	Cross Entropy	Resize, Normalize	<ul style="list-style-type: none">img_size = 416*416	0.9872	0.3304
efficientnet_b5	Adam (scheduler)		Rotate, Flip, Blur, Distortion, Resize, Normalize	<ul style="list-style-type: none">img_size = 416*416	0.9646	0.7239
CoAtNet			Rotate, Flip, Blur, Distortion, Resize, Normalize	<ul style="list-style-type: none">img_size = 384*384	0.8465	0.6416
CoAtNet				<ul style="list-style-type: none">img_size = 384*384Oversampling	0.9735	0.6408
CoAtNet			Rotate, Flip, Blur, Distortion, HueSaturation, CLAHE, Resize, Normalize	<ul style="list-style-type: none">img_size = 384*384	0.9672	0.7567
ConvNeXt				<ul style="list-style-type: none">img_size = 256*256	0.7892	0.6551
tf_efficientnetv2_m				<ul style="list-style-type: none">img_size = 224*224	0.9918	0.8649
tf_efficientnetv2_m				<ul style="list-style-type: none">img_size = 224*224Oversampling애매한 labeling 수정	0.9941	0.8688
HRNet				<ul style="list-style-type: none">img_size = 224*224애매한 labeling 수정	0.5311	0.4001

강사님께 받은 피드백 및 의견을 정리해주세요.

Question	Q1. 문서 분류 CV와 관련된 참고할만한 document가 있는지
Answer	<p>**기본 baseline은 Layout LM V3(MS의 모델)**로 잡아도 됨</p> <p>접근 방식</p> <ul style="list-style-type: none">● pre-trained 가져와서 전이 학습● 직접 학습● Few-shot <ul style="list-style-type: none">- calibration 이후, OCR를 수행하는 방식도 있음- header / body / footer 등 따로따로 잘라 학습할 수도 있음- ViT-LAION-2B도 참조- pre-trained가 많이 되었느냐가 중요하다. 이 때, data가 한국어 문서이므로, 한국어 데이터에도 robust한 pre-trained 모델을 사용하는 것이 결과가 안정적으로 나올 것이다.

강사님께 받은 피드백 및 의견을 정리해주세요.

Question	Q2. 한국어로 된 pretrained가 없다면, 영어로 된 걸 가져와서 fine-tuning을 하는 게 나을지, 밑바닥부터 학습하는 것이 나을지
Answer	<p>가능한 접근법: Transfer Learning, 밑바닥부터 직접 학습, Pre-trained 등 Few Shot 등</p> <p>document 상대로 trained된 게 좋다 - 한국어 Pre-trained LayoutLM 모델들, 영어 Pre-trained LayoutLM 모델들 찾아볼 필요가 있다.</p> <p>text적 특징이 중요한 게 아니면 OCR은 필요하지 않다. 그냥 이미지적 특징이 중요하면 ResNet보다 더 큰 모델, 데이터를 학습한 pre-trained를 가져오면 성능이 좋아질 수 있다.</p> <p>Document 관련 OCR, Input이 Image, Output이 Text인 방식의 Transfer가 가장 적합하다고 고려. 텍스트적 특징이 중요한지 텍스트를 포함한 포맷(문서 양식)이 중요한지 확인이 필요</p> <p>그럼 문서 양식(세금신고서를 예로 들면 종합소득세, 재산세 신고서 등의 포맷)을 학습하는 것인가?</p> <p>문서의 이미지적 feature를 학습하는 것이므로 양식을 학습하는 것이라 볼 수 있습니다.</p> <p>문서 이미지 자체에 대한 EDA가 필요</p> <p>만약 OCR를 하는 중이면 calibration이 중요하다.</p> <p>하지만 classification이라면 양식의 특징이 잘 드러나게 만드는 것이 중요하다. 즉 양식에서 이미지의 특징을 활용한다. 따라서 augmentation 등을 적극적으로 활용해야 한다.</p>

강사님께 받은 피드백 및 의견을 정리해주세요.

Question	Q3. OCR 관련 library는 어떤 것이 있는가?
Answer	<p>Python Tesseract가 가장 유명하다.</p> <p>OCR task의 경우에는, 어떤 library를 사용하는가가 중요한게 아니라, input data인 이미지 내의 text가 얼마나 calibration이 잘 되어있는가가 중요하기 때문에, 위의 library를 사용해도 꽤 안정적인 성능을 낼 수 있을 것이다.</p> <p>결론 :</p> <ol style="list-style-type: none">1. 일단 데이터를 확인하여, 각 class 별 이미지를 구분하는 feature가 (1)이미지 내의 text인지, 혹은 (2)문서 양식 그자체인지를 확인 후, OCR을 사용해야하는지/Image Feature만 뽑아내면 되는지 확인해야한다.2. OCR을 이용하게 된다면, (1)문서 내 이미지의 Calibration을 어떻게 수행할 것인지, (2)또 OCR을 이용해 추출한 text data에 대한 처리는 어떻게 할 것인지(NLP?) 확인해야한다. (c.f.) 문서 전체를 넣게 될 경우, text가 작아져 이 경우, 이미지의 어떤 부분을 OCR로 활용해야하는지도 결정해야할 수 있다.3. Image Feature를 이용하게 된다면, (1)어떤 CV module을 이용해서 학습을 할 것인지, (2)사전학습 모델 중 어떤 모델을 사용할 것인지를 확인해서 진행해야 한다. (3)또한 data augmentation이 중요하다. 사전학습 모델은 torch vision, huggingface 등에서 확인할 수 있다.

강사님께 받은 피드백 및 의견을 정리해주세요.

Question	Q4. 모델이 혼동하는 label들에 대해서는 어떻게 처리하는 게 좋은가?
Answer	<ul style="list-style-type: none">- 이미지적인 차이가 잘 드러나도록 전처리(augmentation)를 하는 방법- 만약, 두 label의 이미지적 특징이 거의 같음에도 불구하고 label이 다르게 붙은 경우, 아예 OCR을 이용해서 접근을 하거나, contrastive learning (metric learning, image semantic search, arcface)를 한번 구현해서 아예 label간 멀어지게 학습을 하게 하는 방법을 사용할 수 있다. 원리 : triplet loss (아예 image를 임베딩할 때, 다른 클래스는 다른 영역으로 임베딩한다.)

강사님께 받은 피드백 및 의견을 정리해주세요.

Advice from lecturer






1. Early Stopping	계속해서 iter를 도는 것은 Overfitting을 야기 tolerence 이하로 모델이 개선이 될 때, iter를 계속 도는 것은 자원의 낭비이다. 이러한 관점에서 Early Stopping을 적용
2. Validation	Valid set을 만드는 기준 (혹은 판단 기준) Valid set의 의미 : Test set과 비슷한 데이터를 기준으로 모델의 일반성을 확인해야한다. 클래스가 많을 경우, stratified 옵션을 걸어 두고 random으로 해도 된다. 이후에는 반복된 실험에 대해서 코드의 오류로 valid set이 변경될 가능성이 있으므로 따로 local로 저장해서 사용하는 것도 방법이 된다.
3. Augmentation	online : pytorch의 Dataloader에 data를 올리고, 여기서 augmentation을 처리하는 것 offline : 이미지 파일 자체에 augmentation을 적용해서 file로 저장해서 사용하는 것 데이터를 직접보는게 편하다 → offline EDA를 할 때에 편리를 위해서 offline을 만드는게 좋다. offline Augmentation을 한 이후, 다시 online augmentation을 적용할 경우, 실제로 본 사진에 다시 변형이 가해지므로 EDA의 의미가 희석될 수 있다. 다만, data의 크기가 큰 경우, online augmentation을 이용해서 컴퓨팅 자원을 줄이는 것이 좋다. 이번 대회에서는 데이터의 크기가 크지 않으므로 offline augmentation을 이용해서 확인을 해보는 것을 추천
4. Model	레이어를 쌓아 만든 모델 vs. pre-trained model 당연히 pre-trained model이 좋다.

05






결과

최종 순위 및 평가지표 결과

Public 리더보드

7 (-)	CV 3조	    	0.9303	101	7d
----------	-------	---	--------	-----	----

Private 리더보드

7 (-)	CV 3조	    	0.9183	101	4d
----------	-------	---	--------	-----	----

06

경진 대회 진행 소감

그룹 스터디 진행 소감

김태한	<ul style="list-style-type: none">- CV 대회는 처음 참여했는데, 대략적으로만 알던 Classification이나 Segmentation, Augmentations에 대해 새로 배울 수 있었습니다.- torchvision과 Albumentation, augraphy를 통한 augmentation을 배울 수 있었습니다.- torchvision, huggingface, timm을 통한 transfer learning과 fine-tuning을 배울 수 있어서 좋았습니다.- 처음이라 모델 성능을 어떻게 향상시켜야할지에 대한 방향을 제대로 알지 못했습니다. 목표에 대한 큰 그림을 그리고 나서 구체적인 방법을 적용하고 확인하는 작업을 체계적으로 수행하지 못했습니다.- 차후에는 보다 체계적으로 특정 방법을 적용, 결과 확인, 이를 보완하는 방법을 찾는 프로세스를 정립해야겠다고 생각했습니다.
문정의	<ul style="list-style-type: none">- 처음 이미지분류 대회를 진행해서 당황스러웠지만, base model 소스를 기반으로 hugging face에서 모델을 선택하고 데이콘 공개 소스를 이용하여 하나씩 맞추어 가다보니 기본적인 소스 완성(학습, 평가, 추론)을 할 수 있었습니다.- 이후 이 모델의 성능 향상을 위해 StratifiedKFold, 이미지 augmentation(offline, online), 학습 이미지크기, 배치사이즈, learning rate의 patience, epoch, loss함수, 모델 ensemble 등에 대한 다양한 적용을 통해서 성능 향상의 진전이 있었으나, 일정 성능 향상(0.9303) 이후 더 이상 성능향상 진전을 할 수 없어서 어려움을 겪게 되었습니다.- 무언가 다시 점검을 해서 성능 향상을 진행해야 되는데, 학습 시간이 많이 소요됨과 조급함으로 인해 진전이 더 이상 없는 것이 안타깝고, 그래도 이미지분류 모델에 대한 많은 것을 배울 수 있는 시간이 되었습니다.- 그룹스터디 멤버분들과 정보를 공유하며, 함께 다양한 의견을 나눌 수 있어서 너무 좋은 경험이었습니다.
이현진	<ul style="list-style-type: none">- CNN에서 대해서 간단한 지식만을 가지고 있었지만, 강의와 대회를 통해서 많은 것을 알아갈 수 있어서 좋은 경험이 되었습니다.- 대회를 진행하면서 모델의 성능을 향상시키는 방법과 더 효율적인 학습 방법을 고민해보고 내용을 공유하면서 서로 알아가는 insight가 많은 점이 좋았습니다.- 다만 기존에 비해 성능을 크게 개선하지 못했고, 조금 더 성능이 좋았다면 어땠을까 하는 아쉬움은 남습니다.- 이후에 정리 및 복기하면서 부족했던 점을 고쳐나갈 수 있다면 좋겠습니다.
김소현	<ul style="list-style-type: none">- 대회에서 주어진 baseline 코드를 활용하여 여러 hyper parameter와 학습 방법들을 실험해 보면서, 데이터에 맞는 모델과 학습 방식이 어떤 것인지 알 수 있었습니다. Augmentation에 따라 모델의 성능이 크게 차이나게 되는 것도 알게 되었고, 최근 큰 기업들이 연구한 좋은 모델들을 어떤 모양으로 transfer하거나 fine-tuning 할 수 있는지 공부할 수 있는 좋은 기회였습니다.- 여러 모델과 방식을 시도해 보았지만 원하는 성능에 도달하지 못했던 것은 조금 아쉽습니다.- 하지만 개인이 시도한 학습 방식에 대해 팀원들과 서로 공유하고, 강사님의 좋은 멘토링 덕에 후반에라도 감을 잡을 수 있었습니다.- ML 프로젝트에 이어 이 대회에서 배운 방식들을 응용하여 향후 프로젝트와 대회에 적용하고, 좋은 성능을 얻어낼 수 있게 되면 좋겠습니다.
권혁찬	<ul style="list-style-type: none">- 이전에 이미지 도메인에 대한 경험이 전무해서 낯설고 익숙하지 않았으나 강의를 들으면서 조금씩 이해가 되고 어떤 방식으로 이미지 데이터를 다루는지 이해할 수 있었습니다.- 이미지 데이터를 잘 학습하기 위해서는 현실세계에서 다양한 형태로 존재할 수 있는 이미지 데이터를 가정한 데이터 증강 기법이 모델 성능에 가장 영향을 준다는 것을 느꼈습니다.- 모델링의 경우 scratch부터 직접 작성하는 것 보다는 이미 다양한 실험과 논문을 통해 증명된 모델을 사용하는 것이 훨씬 효율적이고 성능이 뛰어나다는 것을 알게 되었고 앞으로의 공부 방향도 이런 점을 고려하여 다양한 모델을 잘 사용할 수 있는 기술을 연습하는 것으로 설정해야함을 느꼈습니다.- 이번 대회는 개인적인 사정으로 리더보드 제출에는 많이 참여를 하지 못해서 팀원에게 진솔한 마음입니다 다음 대회부터는 다시 열심히 할 수 있도록

Inference

<https://scikit-learn.org/>

<https://pytorch.org/hub/>

<https://pytorch.org/docs/stable/index.html>

https://albumentations.ai/docs/api_reference/augmentations/transforms/

<https://dacon.io/codeshare/2373>

<https://dacon.io/codeshare/3658>

https://ko.d2l.ai/chapter_deep-learning-basics/

<https://github.com/Bjarten/early-stopping-pytorch/blob/master/pytorchtools.py>

Q&A

감사합니다.

