

프로젝트 #1 결과 발표

[CNN을 이용한 Wafer 불량 검출]

2022. 6. 15

충북대학교 산업인공지능학과

[20-5조] 강운구, 김병근

수행방법 및 기여도

수행방법

- 각자의 장.단점을 고려하여, 준비 영역과 실행 영역으로 분장 함
- 회사는 다르지만, 분장한 내용에 맞게 과제를 수행 함

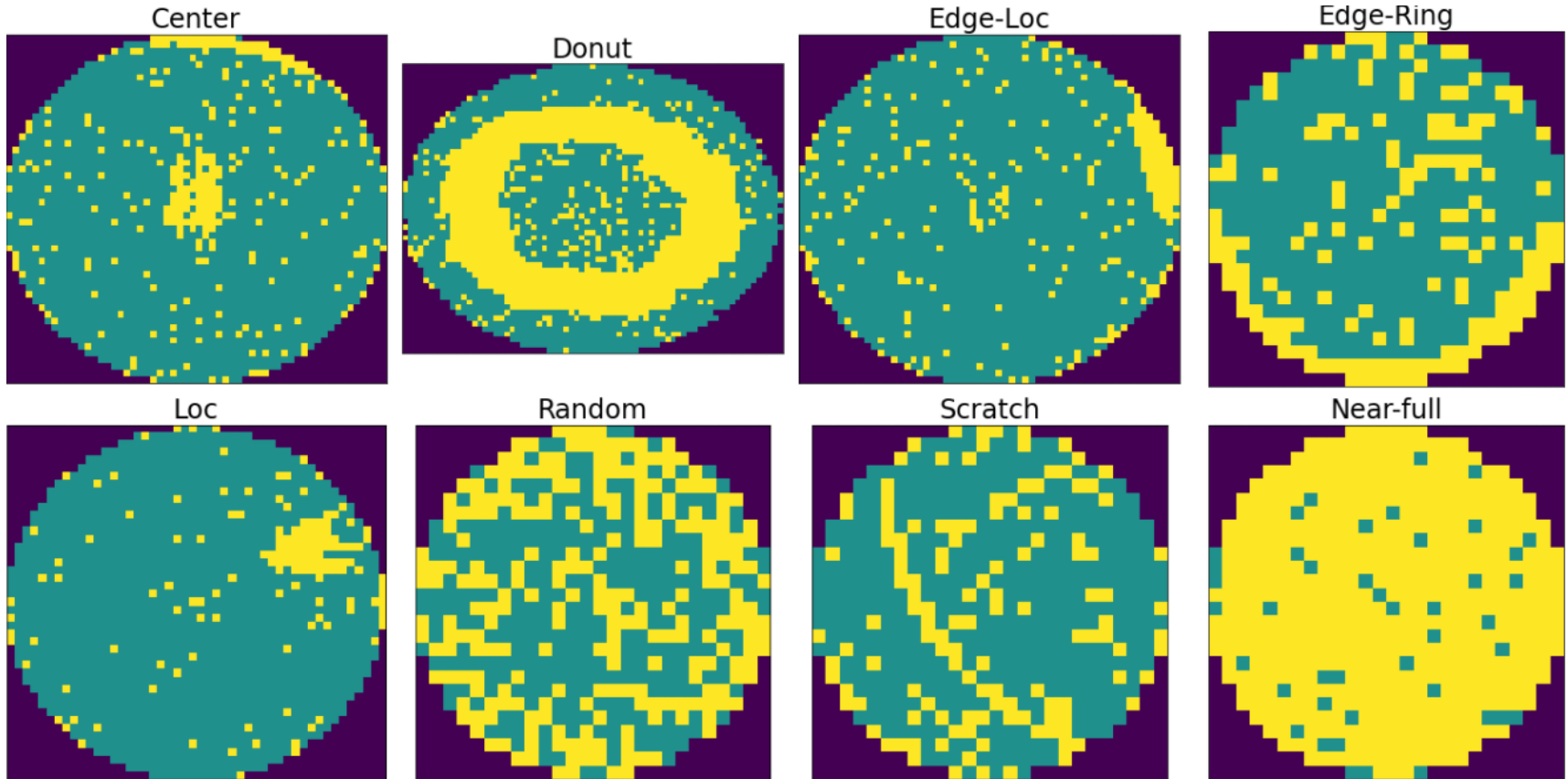
업무분장 및 기여도

이름	비중	수행내용	비고
강윤구	50%	<ul style="list-style-type: none">• 데이터 셋 확보 및 증량• 자료 작성	
김병근	50%	<ul style="list-style-type: none">• 코딩/학습• 결과 발표	

데이터셋

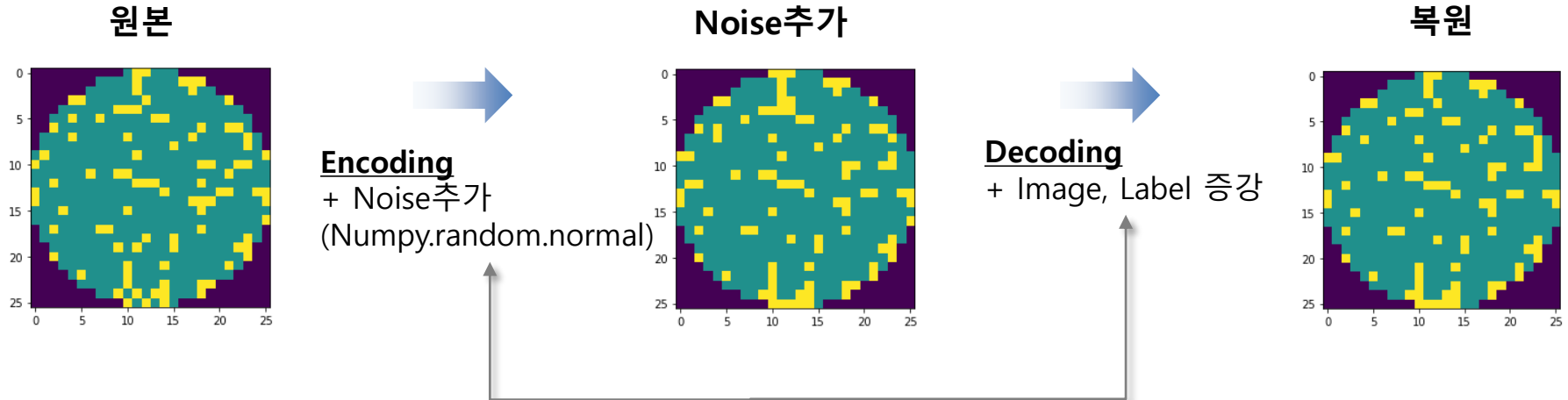
불량 유형(Failure Class)

: 9개의 Class(OK:1, NG:8)와 다양한 해상도 존재



데이터셋

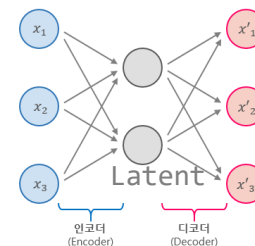
Data augmentation "AutoEncoder"



※ AutoEncoder model 학습

Layer (type)	Output Shape	Param #
input_5 (InputLayer)	[(None, 26, 26, 3)]	0
conv2d_5 (Conv2D)	(None, 26, 26, 64)	1792
max_pooling2d_2 (MaxPooling2)	(None, 13, 13, 64)	0
conv2d_transpose_4 (Conv2DTr	(None, 13, 13, 64)	36928
up_sampling2d_2 (UpSampling2	(None, 26, 26, 64)	0
conv2d_transpose_5 (Conv2DTr	(None, 26, 26, 3)	1731
Total params: 40,451		
Trainable params: 40,451		
Non-trainable params: 0		

<https://excelsior-cjh.tistory.com/187>



인공지능 신경망의 일종
입력:은닉:출력층으로 구성

데이터셋

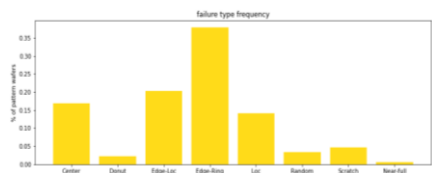
데이터 구성

해상 도: 26*26의 데이터를 가장 데이터가 많은 Nono class기준으로 데이터 증강

학습 비: Train:Test=8:2

<Class 별 데이터 구성>

26*26 size 추출

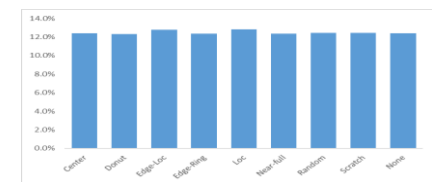


Center 90 (0.6%)	Donut 1 (0.0%)	Edge-Loc 296 (2.1%)	Edge-Ring 31 (0.2%)	Loc 297 (2.1%)	Near-full 16 (0.1%)	Random 74 (0.5%)	Scratch 72 (0.5%)	None 13,489 (93.9%)
------------------------	----------------------	---------------------------	---------------------------	----------------------	---------------------------	------------------------	-------------------------	---------------------------

양품 Class 수량 수준으로!
각 Class별 약 +13,400ea

"AutoEncoder"

<증강 후>



Center 13,500 (12.4%)	Donut 13,402 (12.3%)	Edge-Loc 13,912 (12.8%)	Edge-Ring 13,454 (12.4%)	Loc 13,959 (12.8%)	Near-full 13,424 (12.3%)	Random 13,542 (12.5%)	Scratch 13,536 (12.4%)	None 13,489 (12.4%)
-----------------------------	----------------------------	-------------------------------	--------------------------------	--------------------------	--------------------------------	-----------------------------	------------------------------	---------------------------

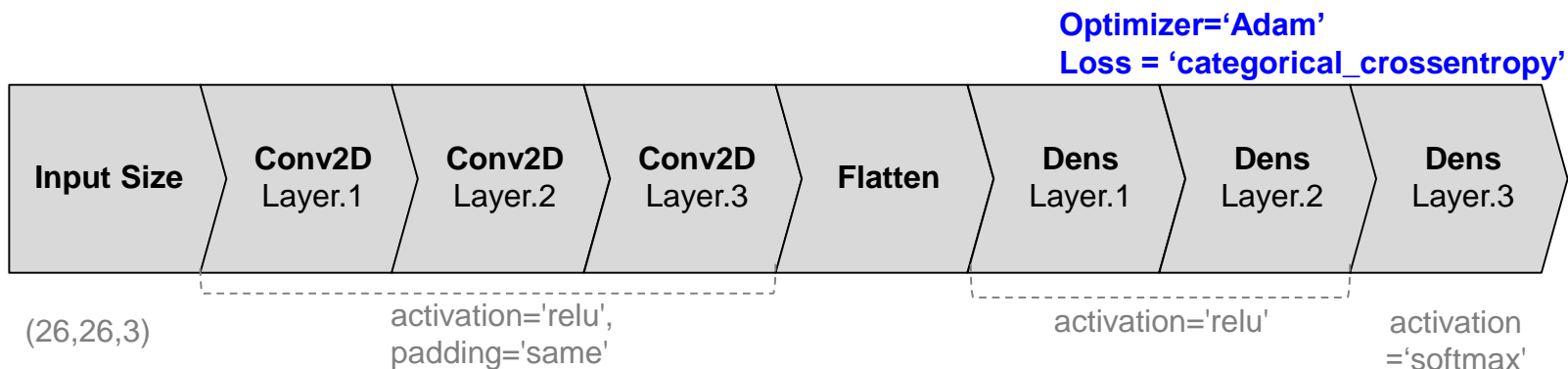
Train
80%

Test
20%

CNN 구조

연구방향: Case 별 성능 비교

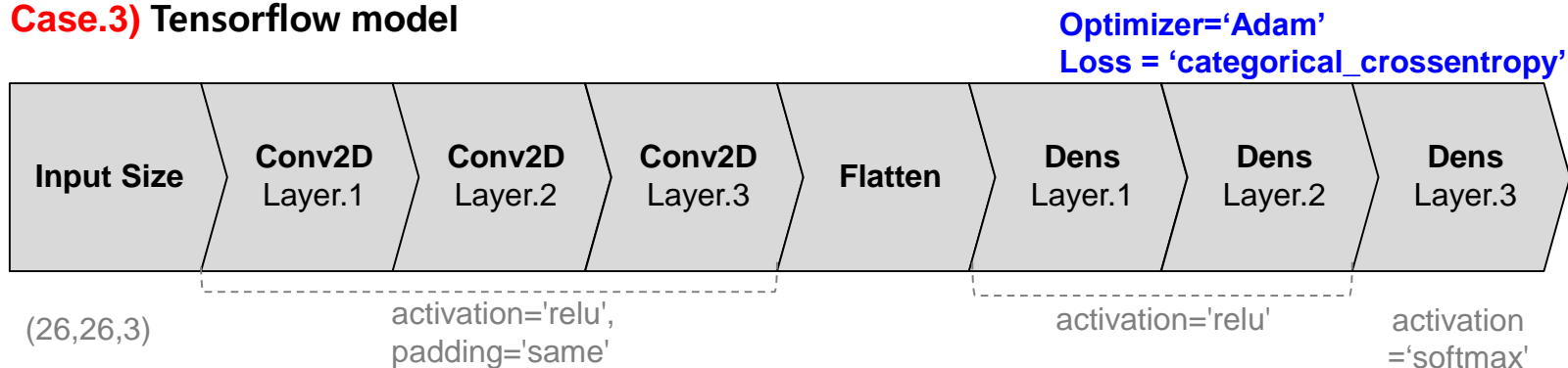
Case.1) Tensorflow 데이터 불균형 상태로 학습 → 과적합(Overfitting) 여부 확인



Data augmentation 후

Case.2) Scikit-Learn의 교차검증 API(cross_val_score)

Case.3) Tensorflow model



CNN 구조

주요 코드 및 실행 결과

- 프레임워크(tensorflow, scikit-learn, vscode)

Tensorflow model

```
def create_model():
    input_shape = (26, 26, 3)
    input_tensor = Input(input_shape)

    conv_1 = layers.Conv2D(16, (3,3), activation='relu', padding='same')(input_tensor)
    conv_2 = layers.Conv2D(64, (3,3), activation='relu', padding='same')(conv_1)
    conv_3 = layers.Conv2D(128, (3,3), activation='relu', padding='same')(conv_2)

    flat = layers.Flatten()(conv_3)

    dense_1 = layers.Dense(512, activation='relu')(flat)
    dense_2 = layers.Dense(128, activation='relu')(dense_1)
    output_tensor = layers.Dense(9, activation='softmax')(dense_2)

    model = models.Model(input_tensor, output_tensor)
    model.compile(optimizer='Adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])

    return model
```

Layer (type)	Output Shape	Param #
=====		
input_12 (InputLayer)	[(None, 26, 26, 3)]	0
conv2d_21 (Conv2D)	(None, 26, 26, 16)	448
conv2d_22 (Conv2D)	(None, 26, 26, 64)	9280
conv2d_23 (Conv2D)	(None, 26, 26, 128)	73856
flatten_6 (Flatten)	(None, 86528)	0
dense_18 (Dense)	(None, 512)	44302848
dense_19 (Dense)	(None, 128)	65664
dense_20 (Dense)	(None, 9)	1161
=====		
Total params: 44,453,257		
Trainable params: 44,453,257		
Non-trainable params: 0		

학습 방법 및 결과

딥러닝 학습 조건

구분	PC 사양
CPU	AMD Ryzen7 2700X 8-Core 3.70Ghz
RAM	64GB
GPU	NVIDIA GeForceGT 1030 2GB

실험 결과

Case.1) 데이터셋의 불균형을 그대로 학습(overfitting)

INPUT

```
Center : 90  
Donut : 1  
Edge-Loc : 296  
Edge-Ring : 31  
Loc : 297  
Near-full : 16  
Random : 74  
Scratch : 72  
none : 13489
```

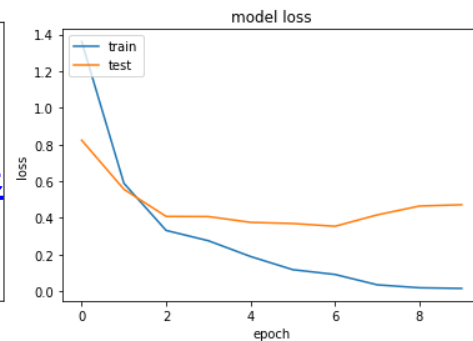
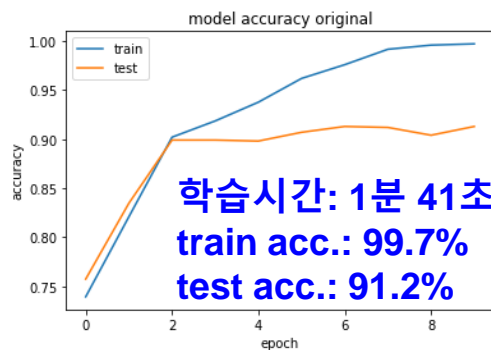
PARAMETERS

Epoch: 10
Batch size: 100

활성화함수: relu, softmax
Padding = same

Optimizer: adam

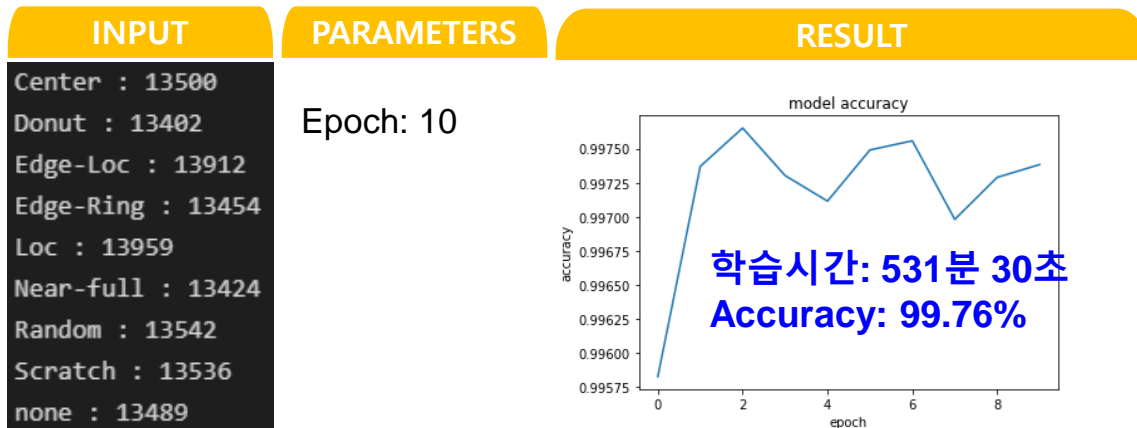
RESULT



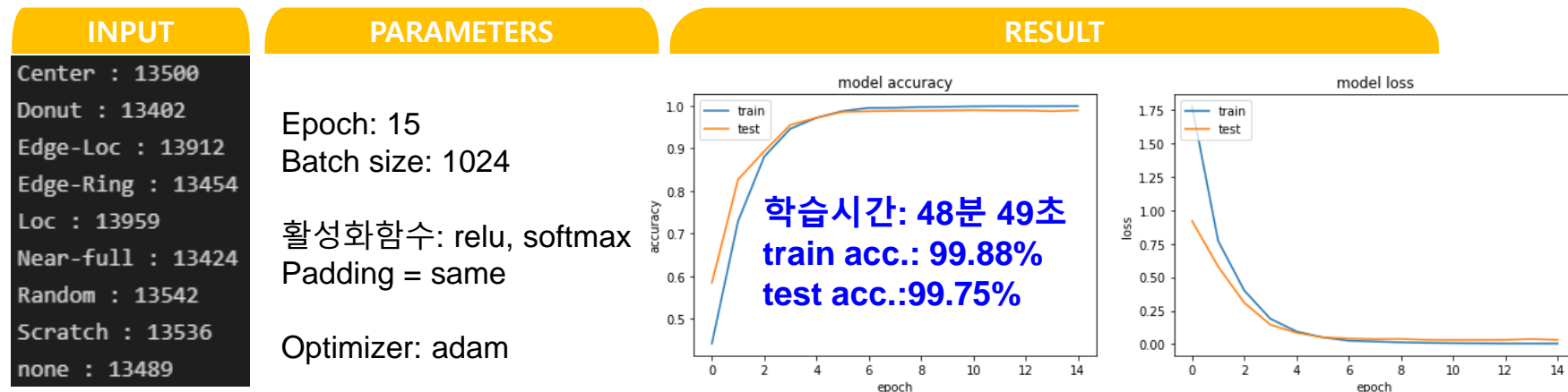
학습 방법 및 결과

실험 결과

Case.2) Scikit-Learn의 교차검증 API(cross_val_score)



Case.3) Keras model

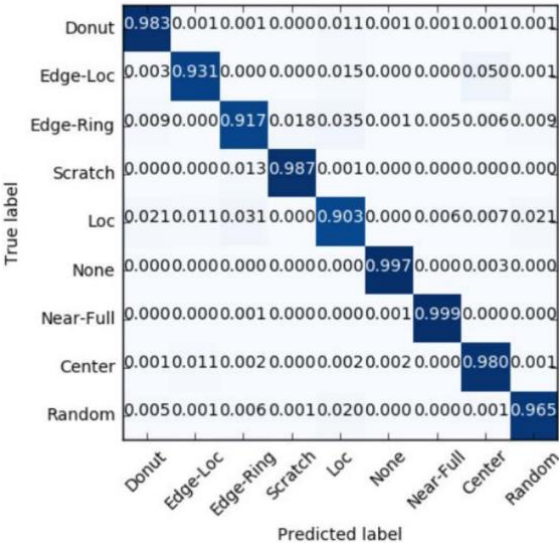


결과 및 토의

분류 성능

- Confusion matrix

(오류를 잡지 못해, 확인 못 함)



OVERALL PERFORMANCE COMPARISON OF VARIOUS CLASSIFIERS (%)

논문

Classifier	Training Acc	Validation Acc	Testing Acc	Precision	Recall	F1-Score
CNN-WDI	98.9	96.4	96.2	96.2	96.2	96.2
CNN-D	97.6	95.5	95.2	95.2	95.2	95.2
CNN-BN	99.4	95.6	95.6	95.6	95.6	95.6
CNN-SD	98.6	94.7	94.8	94.8	94.8	94.8
VGG-16	82.3	80.0	80.1	80.3	80.1	79.9
ANN	95.9	95.9	72.0	95.2	95.9	95.4
SVM	91.3	91.0	32.6	87.5	91.0	88.0

Note: Boldface numbers denote the highest values of different performance measures and Acc accuracy

프로젝트

Classifier	Train Acc.	Vaild Acc.	Test Acc.	Precision	Recall	F1-Score
#1 결과	99.88%	99.75%				

결과 및 토의

1) 처음 알게 됨

- .pickle 파일: Loading속도 높음

- Image file도 배열로 변환 되지만, 미리 배열로 저장한 pkl 파일이 있다는 것을 알게 됨

- . Kaggle site에 대해 알게 됨

2) 연구결과 고찰

- .데이터 증강을 통한 class별 데이터 수의 균형 필요: 데이터 불균형은 무조건 문제 임.

- . Labeling, 표준화된 해상도의 데이터와 분류 Class별 수량의 균형을 맞춘다면,
알고리즘 성능의 차이는 있으나, 만족할 만 한 정확도를 낼 수 있다고 확인 됨

3) 느낀점

- .강윤구:

- .김병근: 김현용교수님의 딥러닝 관련 전문성에 감사하였습니다. 도규원교수님 수고 많으셨습니다.

- 정말 괜찮은 수업 과정이 었으며, 2학년 1학기에 배웠으면 논문에 상당히 도움이 되었다고 생각합니다

감사합니다