

수업 안내

2023. 04. 12

김 현 용

충북대학교 산업인공지능학과

강의 일정

주차	날짜	발표 주제	비고
1	3/08	[강의] 오리엔테이션 / 조 편성	비대면수업
2	3/15	[강의] Project #1: CNN을 이용한 불량 검출	대면수업(1)
3	3/22	조별 토의 및 멘토링, [강의] OpenCV 기본 명령어	비대면수업
4	3/29	조별 토의 및 멘토링, [강의] Numpy와 Matplotlib 시각화	비대면수업
5	4/05	Project #1 주제발표(5): 한희주, 명성구, 권진관, 백정흠, 신건철	대면수업(2)
6	4/12	Project #1 주제발표(3): 박영제, 김현기, 원윤재, 조별 토의 및 멘토링	비대면수업
7	4/19	프로젝트 최종점검(사전발표)	비대면수업
8	4/26	Project #1 발표평가	대면수업(3)
9	5/03	[학과행사] 가디언별 토의 → 장소, 시간 별도 통보 (22~23학번 통합)	대면수업(4)
10	5/10	[강의] Project #2: YOLO를 이용한 객체 검출	비대면수업
11	5/17	조별 토의 및 멘토링, [강의] CUDA 및 YOLO 환경구성, Numpy와 Pytorch, 전과정 시연	비대면수업
12	5/24	Project #2 주제발표(11): 이선명, 김홍열, 임강혁, 안병승, 안성인, 송동건, 이재익, 이정현, 장현우, 한병엽, 이진우	대면수업(5)
13	5/31	조별 토의 및 멘토링, [강의] YOLOv8과 객체분할	비대면수업
14	6/07	프로젝트 최종점검(사전발표): 테스트 데이터 공개 → 검출결과 제출	비대면수업
15	6/14	Project #2 발표평가	대면수업(6)

구분	시간		수업 내용
강의	19:00~19:10	10'	수업 안내 / 출석체크 (중간고사 답안 미제출자 확인)
	19:10~20:00	50'	주제발표 (3명)
조별 활동	20:00~20:30	30'	조별 토의 / 휴식
조별 발표	20:30~21:00	30'	프로젝트 진행상황 점검 (팀장, 팀원 모두) Q & A
마무리	21:00 ~21:10	10'	과제 안내

□ 학과 공지사항

- 애로사항 발생 시에는 가디언, 지도교수, 센터장 등과 상담 후 결정
- 포트폴리오 깃허브 관리 부실 → 수업/업무 시 작성한 코드, 작품, 문서 정리

개인별 발표 주제 선정

□ 주제발표 방법

- 10분 발표 + 5분 질의응답
- 발표 주제 및 순서

Project #1 (4/5)			Project #2 (5/24)		
#	성명	주제	#	성명	주제
1	김현기	데이터 증량	1	이선명	주석
2	백정흠	데이터 증량	2	임강혁	YOLO 데이터 증량
3	신건철	데이터 증량	3	한병엽	YOLO 데이터 증량
4	명성구	CNN구현(tensorflow)	4	김홍열	YOLO 사용법
5	권진관	CNN구현(tensorflow)	5	안병승	YOLO 사용법
6	원윤재	CNN구현(pytorch)	6	이재익	YOLO 사용법
7	한희주	학습기법	7	송동건	YOLO 변종
8	박영제	학습기법	8	이정현	YOLO 변종
			9	안성인	평가지표
			10	이진우	객체분할
			11	장현우	TensorRT

□ 평가방법

항목	내용	점수
충실도	• 논문, 인터넷, 책자 등 활용 가능 • 출처를 명시할 것	40
전달력	• 이해하기 쉽도록 발표 자료 작성 및 설명력	30
이해도	• 질의응답에 대한 답변 능력	30

▣ 조별 토의 및 멘토링

- (조별) 프로젝트 진행상황 발표 → 간단한 자료, 결과물 등 줌공유 활용 가능
- (개인별) 프로젝트를 위해 수행하고 있는 내용 발표
- (Q&A) 수행 중 애로사항, 성공 비법 등 공유 (조-교수자-학생)

조	조원	프로젝트 진행상황	비고
1	박영제, 이정현 , 임강혁		
2	이진우, 송동건		
3	원윤재, 김현기		
4	권진관 , 안병승		
5	이재익 , 백정흠		
6	안성인 , 한병엽		
7	장현우 , 명성구		
8	김홍열 , 이선명		
9	신건철 , 한희주		

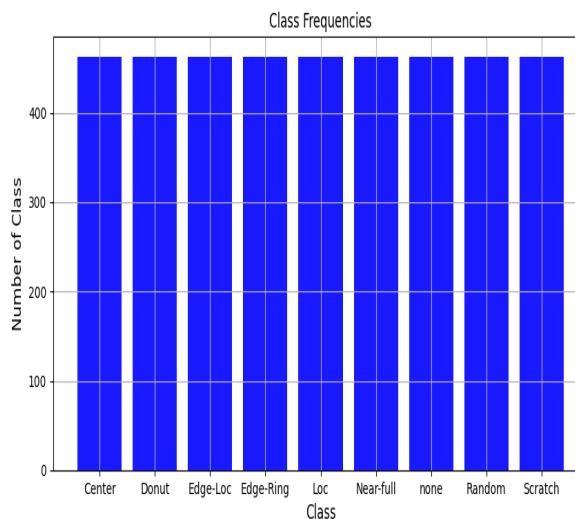
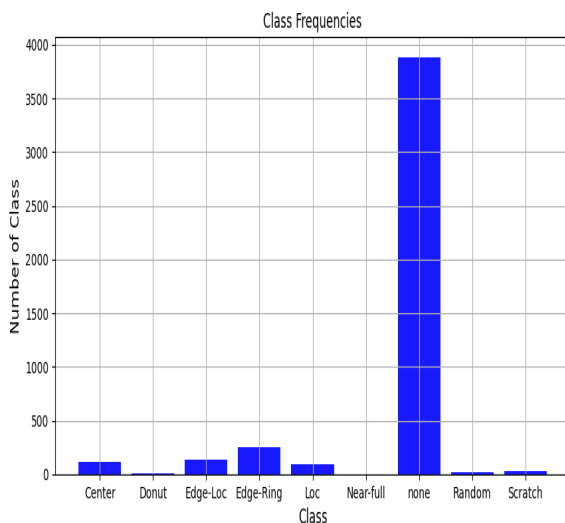
과제물

구분	세부 내용	비고
조별	<ul style="list-style-type: none"> • 데이터셋 다운로드 및 데이터 포맷 확인 (Kaggle 참고) • 딥러닝 환경 구축 : H/W, S/W • 신경망 구현 연습 (참고문헌 참고) • 데이터 증량 (OpenCV, 딥러닝 프레임워크, alimentation 모듈) - 중간발표 양식 배포 • 논문 모델 구현 및 학습, 전이학습(VGG-16) 등 비교/분석 • 프로젝트 #1 발표평가 	<p>-</p> <p>~3/22</p> <p>~3/29</p> <p>~4/12</p> <p>~4/19</p> <p>~4/26</p>
개인별	<ul style="list-style-type: none"> • 주제발표 준비 	~4/24

데이터셋 분할

- 원래 데이터는 클래스별로 0.1% ~ 85.2%의 극심한 **불균형 분포**를 보임
- train : val : test = 17,982 : 5,129 : 3,847 (약 20%)

데이터 분포 (테스트셋)



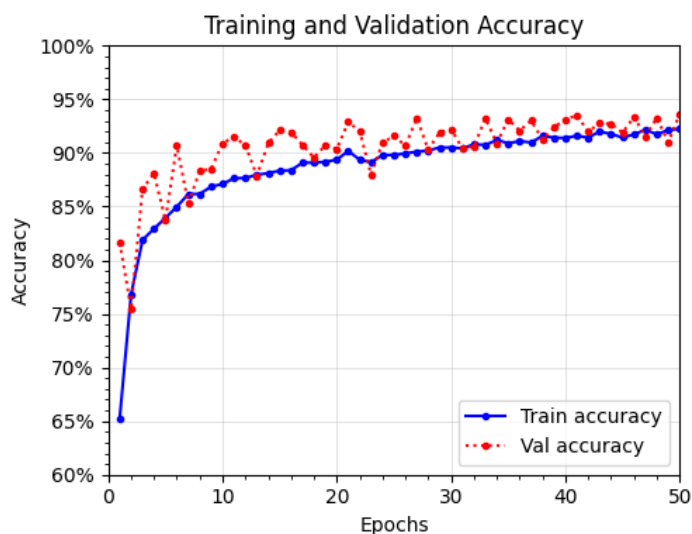
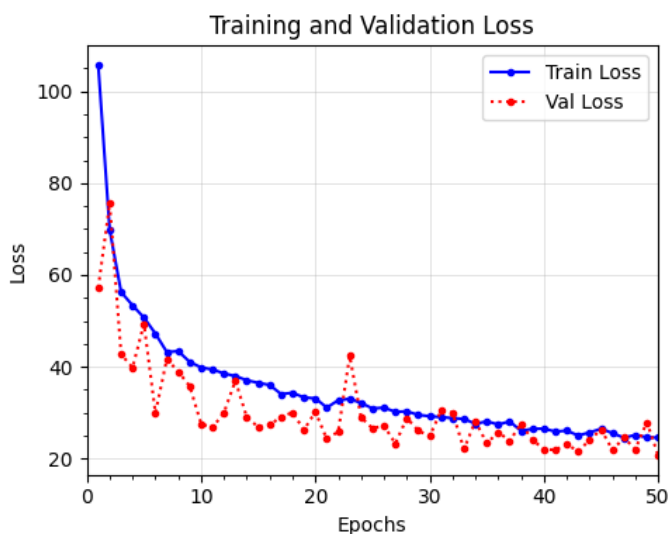
학습 결과 (1) – imbalanced data

학습 결과

- (정확도) train: val: test = 98.24% : 98.64% : **98.48%**
- (transforms) train : val : test = ○ : X : X
- 학습조건 : batch size 112, Adam(lr=0.001)50 epoch, 2hr 소요

데이터 transform 정의

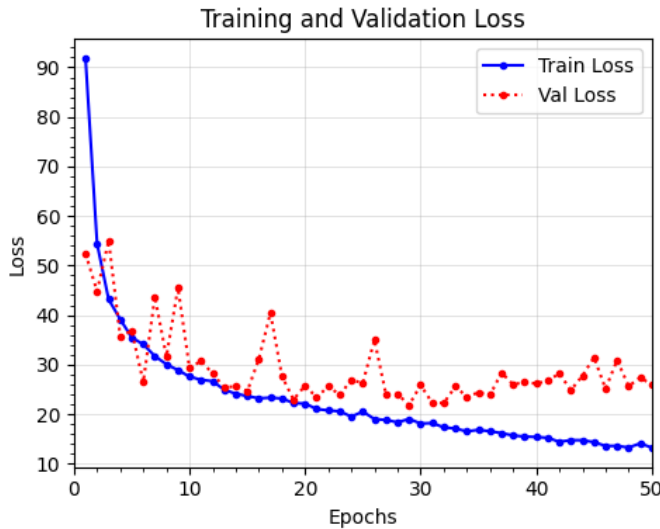
```
Resize(224),
RandomRotation(45, fill=1),
RandomHorizontalFlip(),
RandomVerticalFlip(),
RandomResizedCrop(224, scale=(0.9, 1.0), ratio=(0.8, 1.25)),
RandomAffine(scale=(0.8, 1.2), shear=5, fill=1)
```



▣ 데이터셋 분할

- 클래스별로 under/over-sampling을 통해 동일한 개수로 만들
- train : val : test = 18,000 : 5,535 : **4,158** (20%)

▣ 학습 결과 : (Accuracy) train : val : test = **95.69%** : 93.53% : 93.55%



학습 결과 (3) – 비교

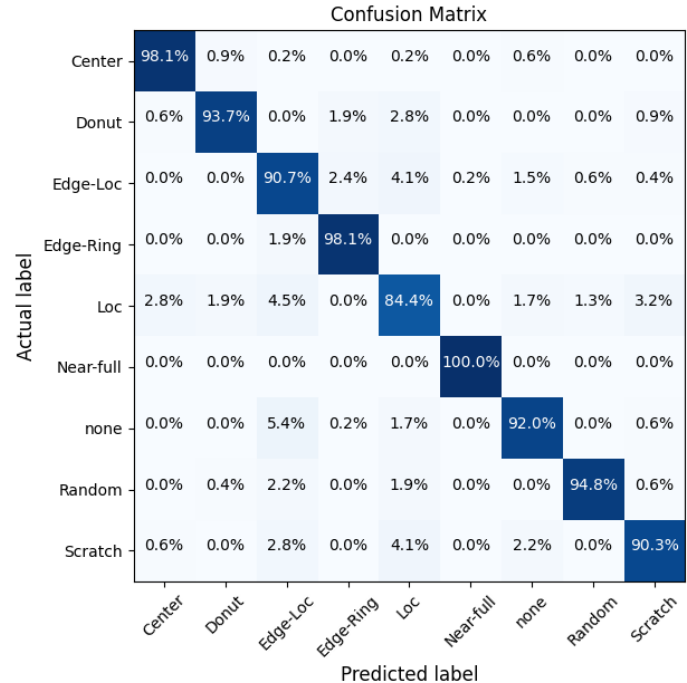
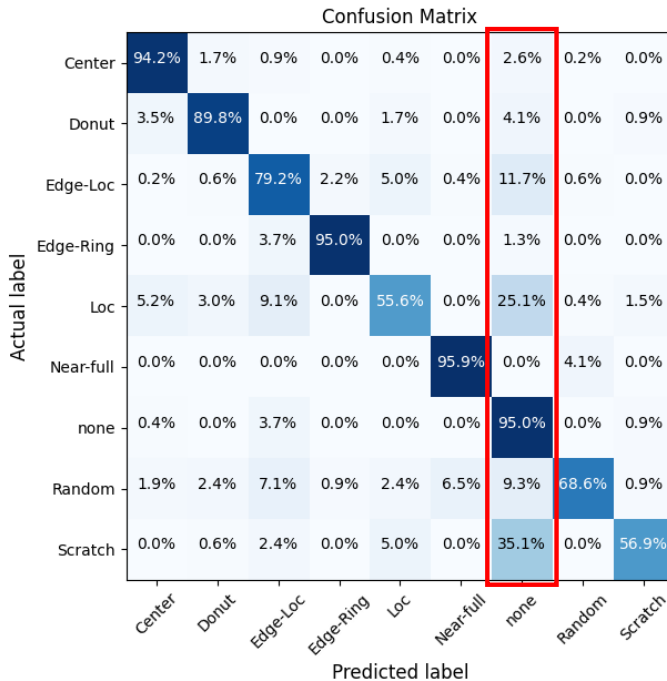
▣ 학습 결과 비교

- 불균형 데이터셋 학습/테스트 VS 균일 데이터셋 학습/테스트

No.	Defect Class	Imbalanced dataset			Balanced dataset		
		Precision	Recall	F1-score	Precision	Recall	F1-score
0	Center	89.32	94.16	91.68	95.97	98.05	97.00
1	Donut	91.41	89.83	90.61	96.65	93.82	95.16
2	Edge-Loc	74.69	79.22	76.89	84.14	90.69	87.29
3	Edge-Ring	96.91	95.02	95.96	95.57	98.05	96.79
4	Loc	79.32	55.63	65.40	84.97	84.42	84.69
5	Near-full	93.26	95.89	94.56	99.78	100.00	99.89
6	none	51.59	95.02	66.87	93.82	91.99	92.90
7	Random	92.69	68.61	78.85	97.99	94.81	96.37
8	Scratch	93.26	56.93	70.70	93.92	90.26	92.05
Average		84.72	81.15	82.89	93.65	93.57	93.57

■ 학습결과 비교

- Confusion matrix 비교 (actual label 기준 비율)



전이 학습(1) – EfficientNetB0

■ EfficientNet_B0

```
(classifier): Sequential(
  (0): Dropout(p=0.2, inplace=True)
  (1): Linear(in_features=1280, out_features=1000, bias=True)
)
```

■ pretrained model 불러오기

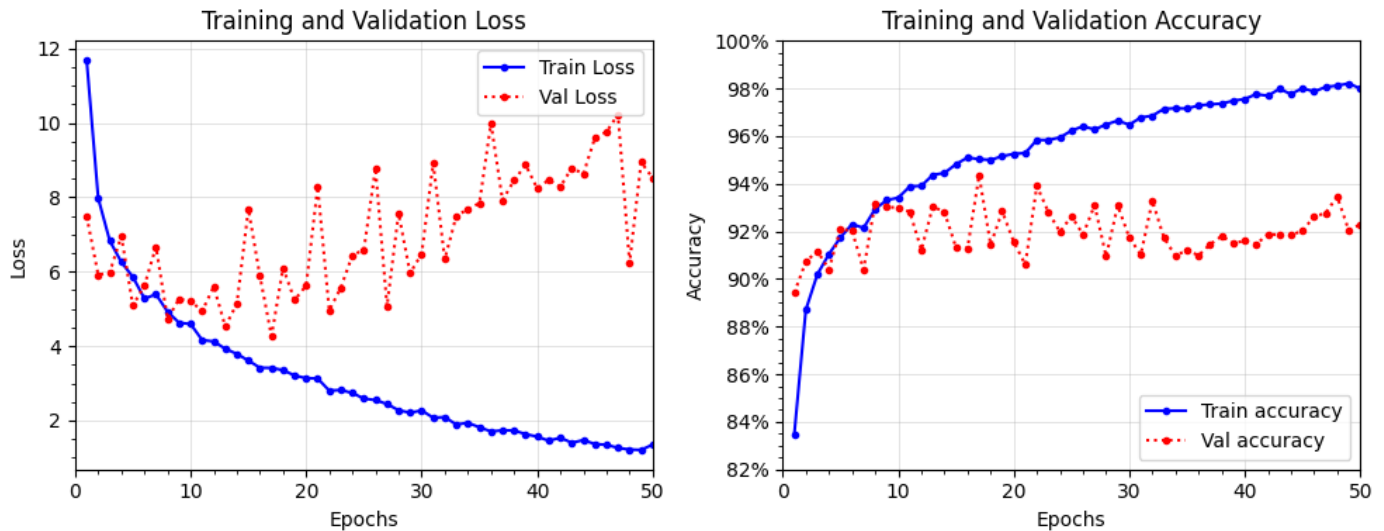
```
model = torchvision.models.efficientnet_b0(weights='DEFAULT')

if freeze_convnet: # ConvNet as fixed feature extractor vs. Finetuning
    for param in model.parameters():
        param.requires_grad = False # 파라미터 고정

# 새로 생성된 모듈의 매개변수는 기본적으로 requires_grad=True 임
num_in_features = model.classifier[1].in_features
model.classifier[1] = nn.Linear(in_features=num_in_features,
                                out_features=num_class)
```

□ EfficientNet_B0

- 학습 조건 : batch size 24, Adam(lr=0.001), 50 epoch, 3hr 소요
- 학습 결과 : (Accuracy) train : val : test = 98.21% : 94.35% : 93.15%



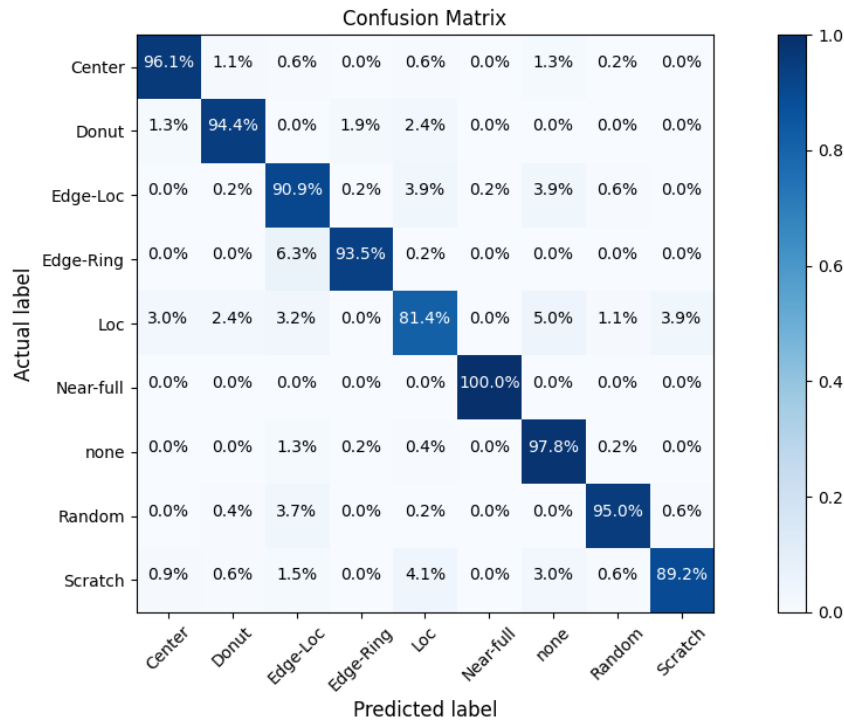
전이 학습(3) – EfficientNetB0

□ 학습결과 비교

- Balanced data
- EfficientNetB0 vs 제안하는 CNN 모델

No.	Defect Class	EfficientNetB0			제안하는 CNN 모델		
		Precision	Recall	F1-score	Precision	Recall	F1-score
0	Center	94.87	96.10	95.48	95.97	98.05	97.00
1	Donut	95.20	94.37	94.78	96.65	93.82	95.16
2	Edge-Loc	84.51	90.91	87.59	84.14	90.69	87.29
3	Edge-Ring	97.52	93.51	95.47	95.57	98.05	96.79
4	Loc	87.24	81.39	84.21	84.97	84.42	84.69
5	Near-full	99.78	100.00	99.89	99.78	100.00	99.89
6	none	88.11	97.84	92.72	93.82	91.99	92.90
7	Random	97.12	95.02	96.06	97.99	94.81	96.37
8	Scratch	95.15	89.18	92.07	93.92	90.26	92.05
Average		93.28	93.15	93.21	93.65	93.57	93.57

Confusion matrix



감사합니다
Q & A