

Benchmark Results of Project #1

2024. 04. 08

김 현 용

충북대학교 산업인공지능학과

Benchmark Results

- Using the original (imbalanced) dataset
 - After performing data augmentation
 - Using a balanced dataset
 - from EfficientNet_B0 transfer learning
-

■ 데이터셋 분할

- 원래 데이터는 클래스별로 0.1% ~ 85.2%의 극심한 불균형 분포를 보임
- train : val : test = 65% : 20% : 15% = 112,421 : 32,085 : 28,444

■ 학습 결과

```
Python 3.10.9 (tags/v3.10.9:1dd9be6, Dec 6 2022, 20:01:21) [MSC v.1934 64 bit (AMD64)] on win32
>>> runfile('D:\WW52-Python\WW6-Cap_cnn_wafer\WWwm_train.py', wdir='D:\WW52-Python\WW6-Cap_cnn_wafer')
총 데이터 개수 - train : val = 112,421 : 32,085
```

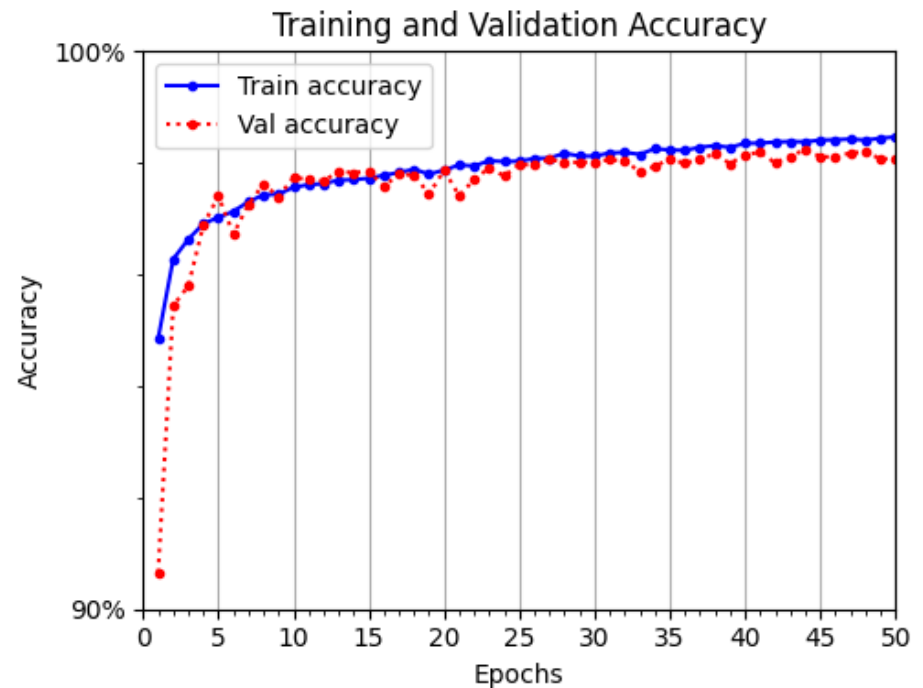
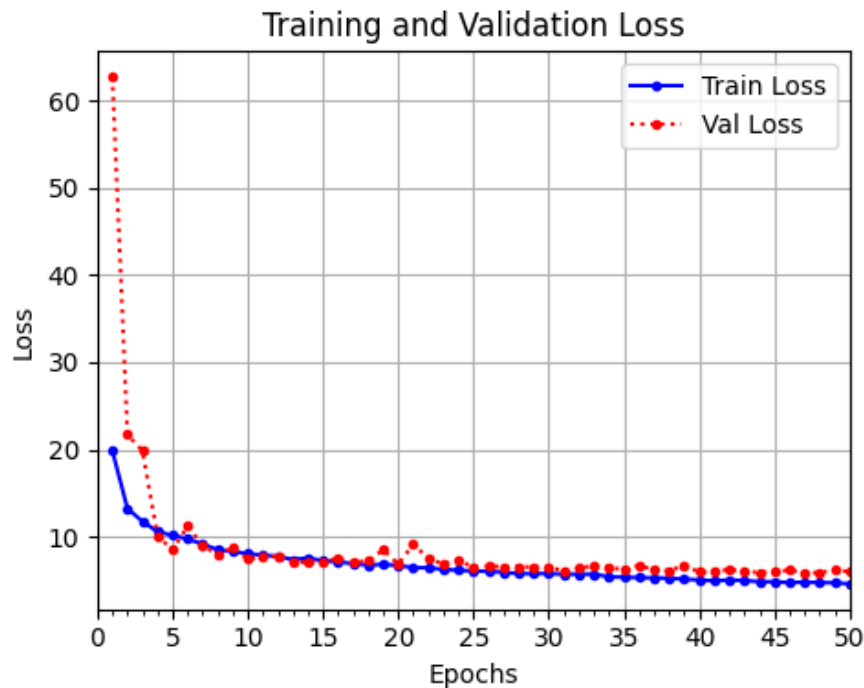
Epoch	Best	Acc/trn	Acc/val	Time/e	Elapsed hrs	Exp. end date (hrs)
1	+	94.85%	90.66%	8.43m	0.14h	2023-04-02-Sun 10:37 (7.0h)
2	+	96.28%	95.44%	7.91m	0.27h	2023-04-02-Sun 10:24 (6.8h)
3	+	96.63%	95.79%	8.39m	0.41h	2023-04-02-Sun 10:27 (6.9h)
4	+	96.91%	96.89%	8.07m	0.55h	2023-04-02-Sun 10:25 (6.8h)
44	+	98.38%	98.24%	7.95m	5.89h	2023-04-02-Sun 10:17 (6.7h)
45	-	98.41%	98.11%	7.92m	6.02h	2023-04-02-Sun 10:17 (6.7h)
46	-	98.42%	98.11%	7.91m	6.15h	2023-04-02-Sun 10:17 (6.7h)
47	-	98.43%	98.18%	7.88m	6.28h	2023-04-02-Sun 10:16 (6.7h)
48	-	98.41%	98.20%	7.93m	6.42h	2023-04-02-Sun 10:16 (6.7h)
49	-	98.44%	98.08%	7.91m	6.55h	2023-04-02-Sun 10:16 (6.7h)
50	-	98.47%	98.07%	7.91m	6.68h	2023-04-02-Sun 10:16 (6.7h)

Best accuracy: (train) 98.38%, (val) 98.24%

Best model: best_model_2404021016.pth

■ 학습 결과

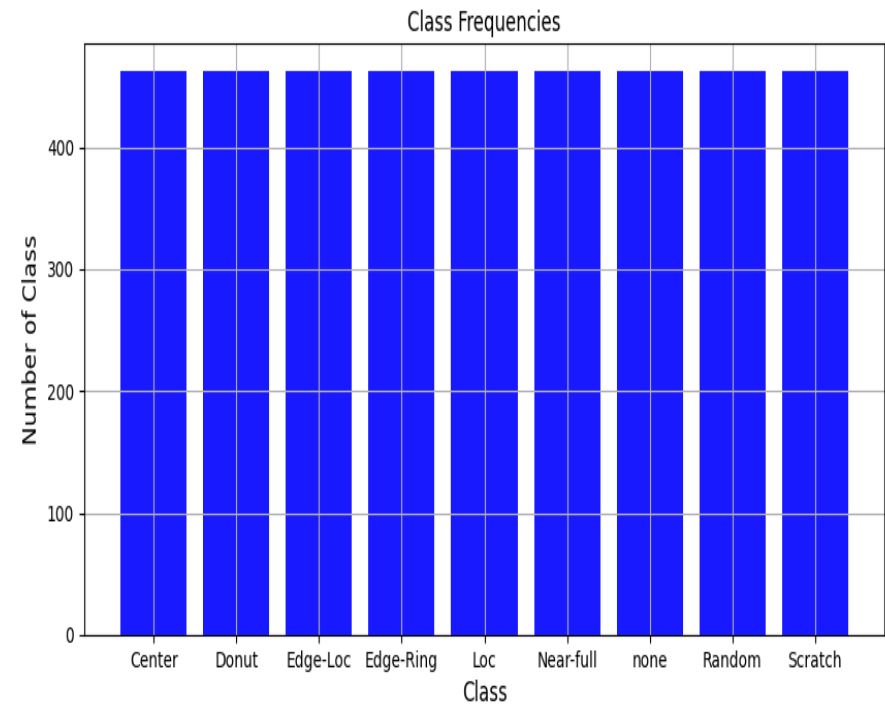
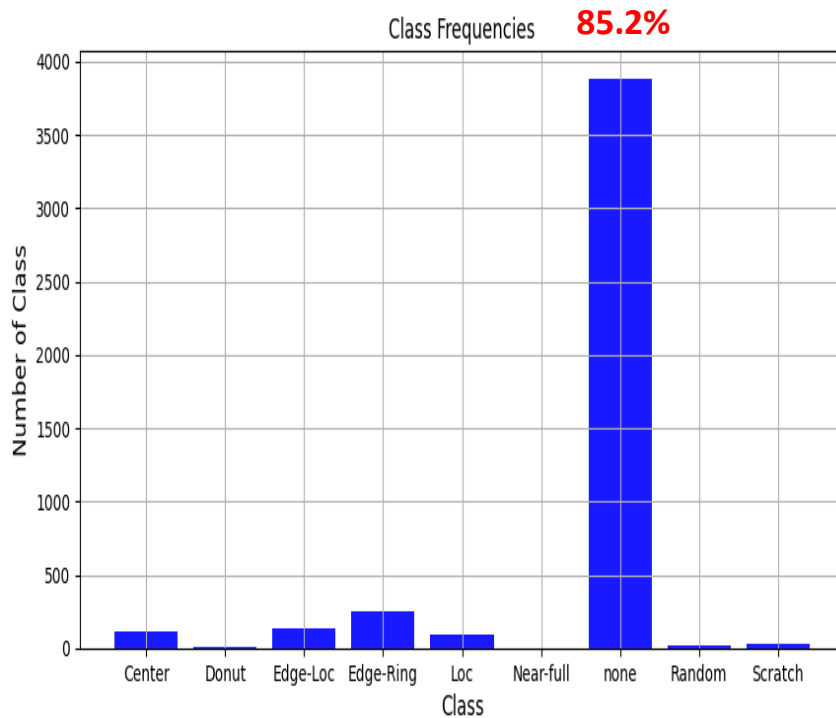
- Best accuracy: (train) 98.38%, (val) **98.24%** → 실제로 혼동행렬 상에서는 클래스별 정확도의 편차가 심함
- 불균형 데이터셋인 경우 accuracy는 정확한 평가지표가 될 수 없다!



■ 데이터셋 분할

- 원래 데이터는 클래스별로 0.1% ~ 85.2%의 극심한 **불균형 분포**를 보임
- train : val : test = 17,982 : 5,129 : 3,847 (총 30%)

■ 데이터 분포 (테스트셋)



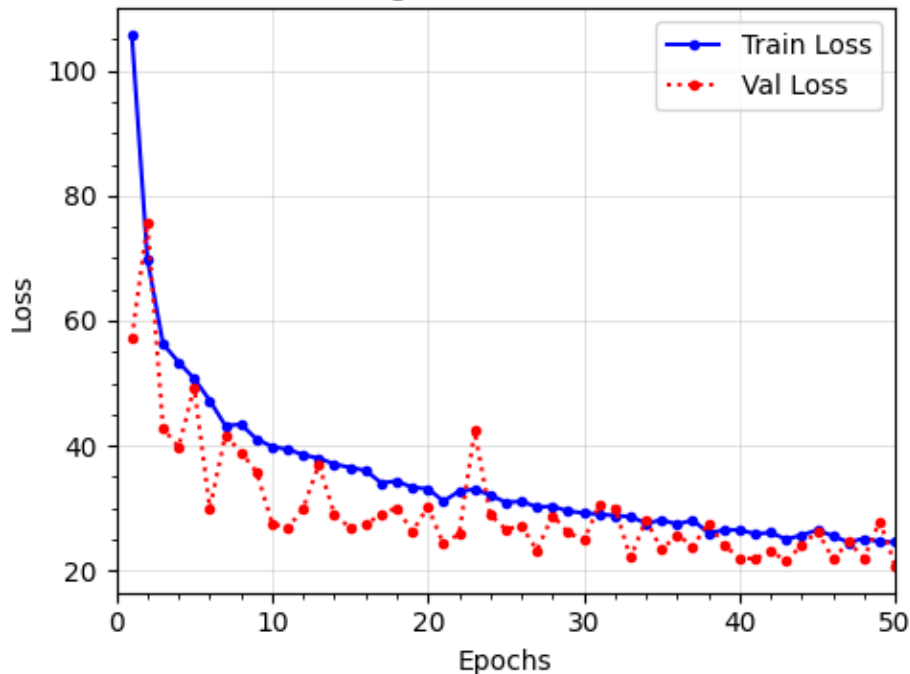
■ 학습 결과

- (augmentation) train : val : test = 0 : X : X
- 학습조건 : batch size 112, Adam (lr=0.001)
50 epoch, 2hr 소요
- (정확도) train: val: test = 98.24% : 98.64% : 98.48%

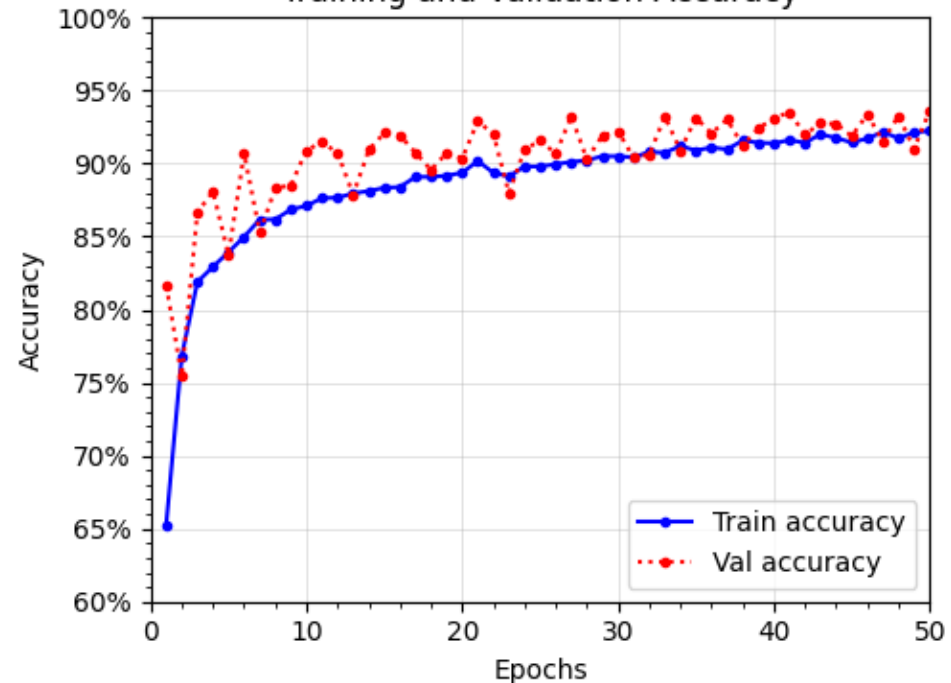
데이터 transform 정의

```
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)
```

Training and Validation Loss



Training and Validation Accuracy

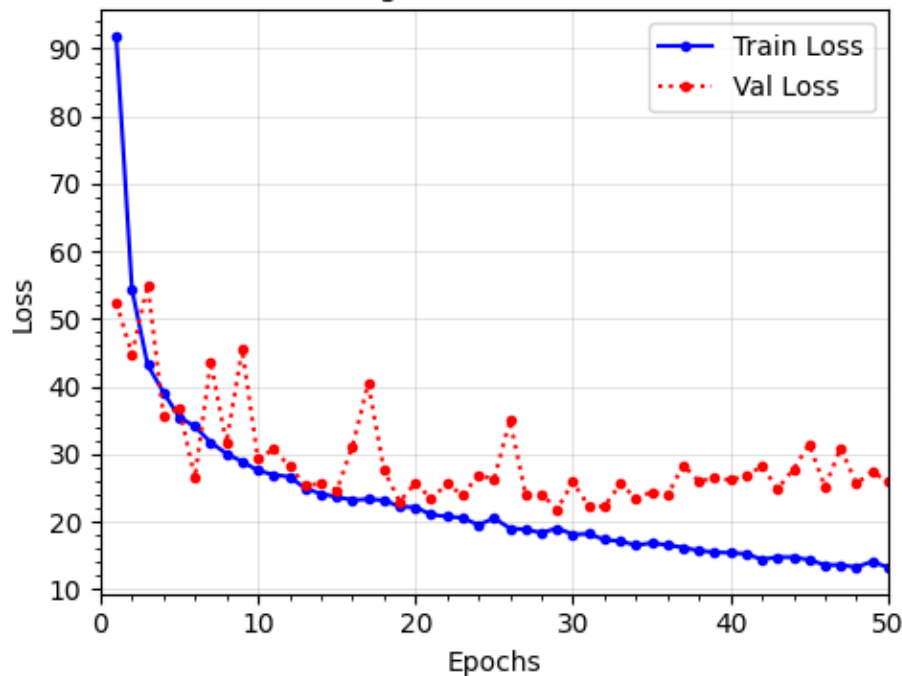


■ 데이터셋 분할

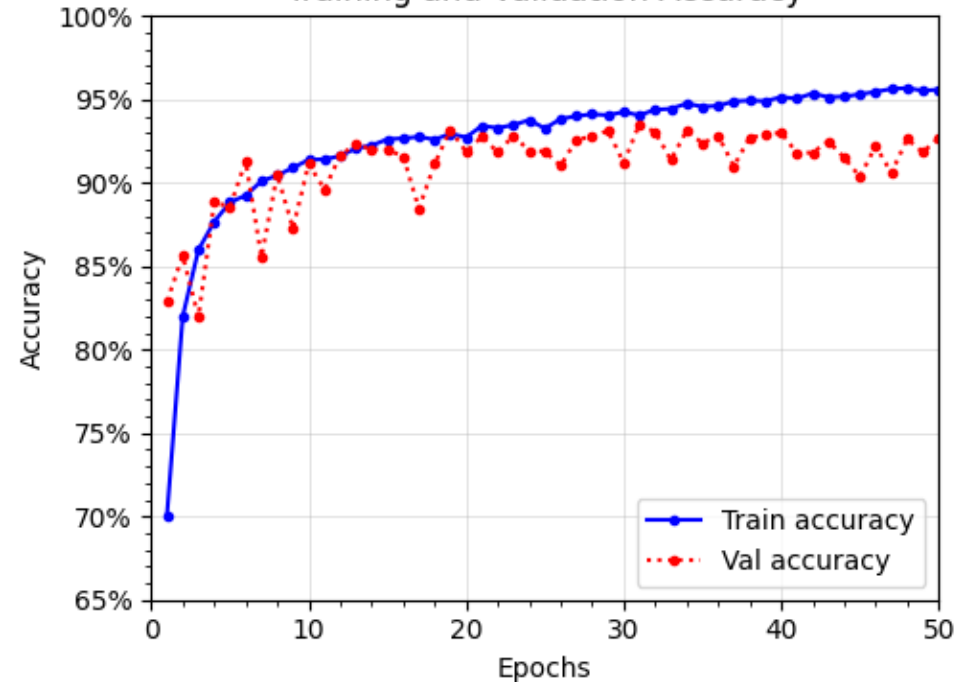
- 클래스별로 under/over-sampling을 통해 동일한 개수로 만들
- train : val : test = 18,000 : 5,535 : 4,158 (총 30%)

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

Training and Validation Loss



Training and Validation Accuracy



■ 학습 결과 비교

- Balanced dataset으로 학습한 모델의 성능 비교
- 테스트 데이터셋 : 불균형 데이터셋 VS 균일 데이터셋

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)

Confusion Matrix

Actual label \ Predicted label	Center	Donut	Edge-Loc	Edge-Ring	Loc	Near-full	none	Random	Scratch
Center	94.2%	1.7%	0.9%	0.0%	0.4%	0.0%	2.6%	0.2%	0.0%
Donut	3.5%	89.8%	0.0%	0.0%	1.7%	0.0%	4.1%	0.0%	0.9%
Edge-Loc	0.2%	0.6%	79.2%	2.2%	5.0%	0.4%	11.7%	0.6%	0.0%
Edge-Ring	0.0%	0.0%	3.7%	95.0%	0.0%	0.0%	1.3%	0.0%	0.0%
Loc	5.2%	3.0%	9.1%	0.0%	55.6%	0.0%	25.1%	0.4%	1.5%
Near-full	0.0%	0.0%	0.0%	0.0%	0.0%	95.9%	0.0%	4.1%	0.0%
none	0.4%	0.0%	3.7%	0.0%	0.0%	0.0%	95.0%	0.0%	0.9%
Random	1.9%	2.4%	7.1%	0.9%	2.4%	6.5%	9.3%	68.6%	0.9%
Scratch	0.0%	0.6%	2.4%	0.0%	5.0%	0.0%	35.1%	0.0%	56.9%

Confusion Matrix

Actual label \ Predicted label	Center	Donut	Edge-Loc	Edge-Ring	Loc	Near-full	none	Random	Scratch
Center	98.1%	0.9%	0.2%	0.0%	0.2%	0.0%	0.6%	0.0%	0.0%
Donut	0.6%	93.7%	0.0%	1.9%	2.8%	0.0%	0.0%	0.0%	0.9%
Edge-Loc	0.0%	0.0%	90.7%	2.4%	4.1%	0.2%	1.5%	0.6%	0.4%
Edge-Ring	0.0%	0.0%	1.9%	98.1%	0.0%	0.0%	0.0%	0.0%	0.0%
Loc	2.8%	1.9%	4.5%	0.0%	84.4%	0.0%	1.7%	1.3%	3.2%
Near-full	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%
none	0.0%	0.0%	5.4%	0.2%	1.7%	0.0%	92.0%	0.0%	0.6%
Random	0.0%	0.4%	2.2%	0.0%	1.9%	0.0%	0.0%	94.8%	0.6%
Scratch	0.6%	0.0%	2.8%	0.0%	4.1%	0.0%	2.2%	0.0%	90.3%

■ EfficientNet_B0



EfficientNetB0

```
(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')

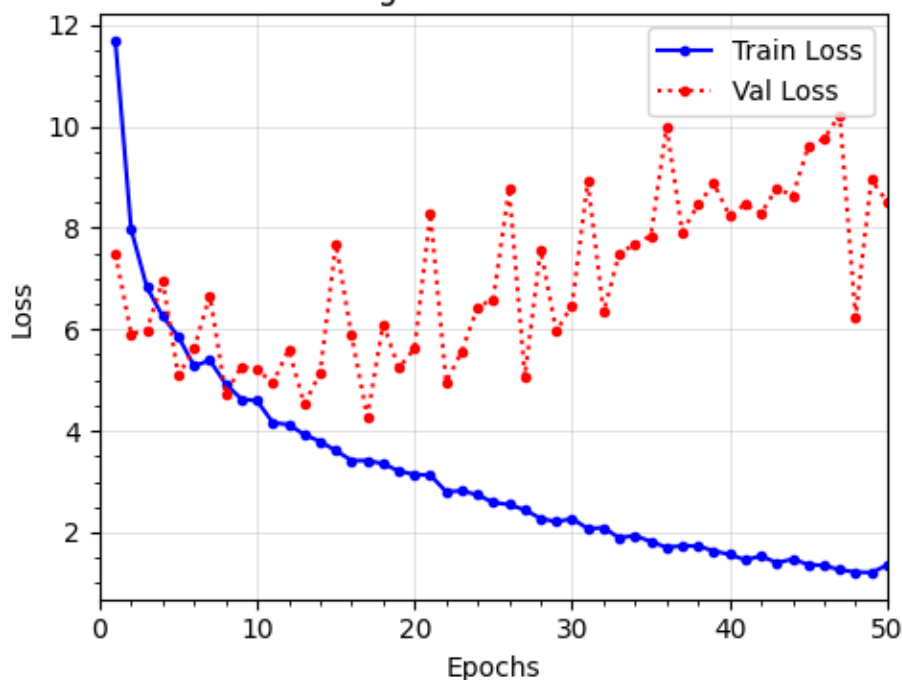
# ConvNet as fixed feature extractor vs. Finetuning
for param in model.parameters():
    param.requires_grad = False      # 파라미터 고정

# 새로 생성된 모듈의 매개변수는 기본값이 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)
```

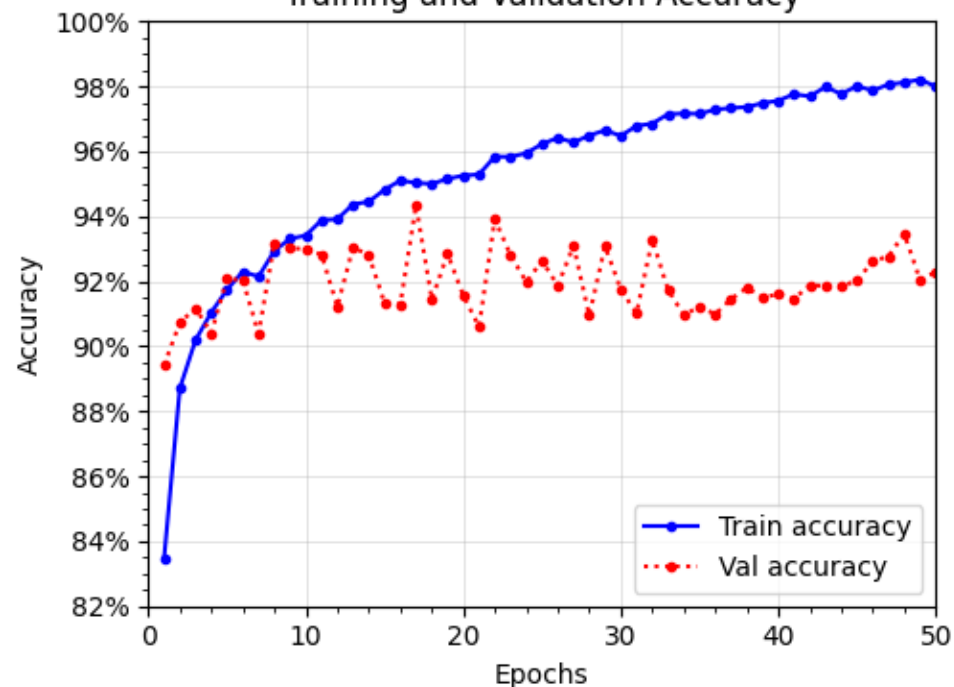
■ EfficientNet_B0

- train : val : test = 18,000 : 5,535 : 4,158 (총 30%)
- 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%

Training and Validation Loss



Training and Validation Accuracy

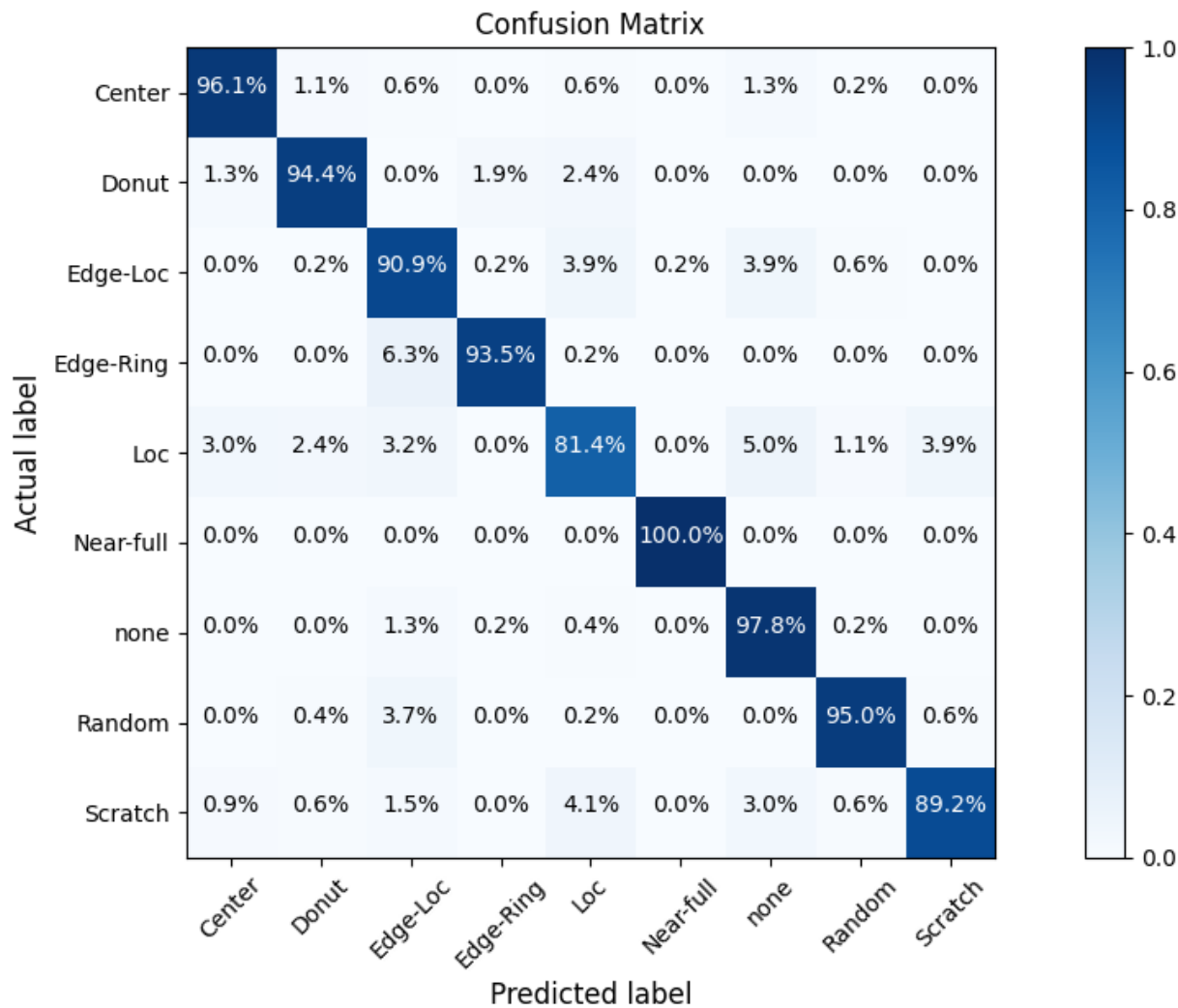


■ 학습결과 비교

- 테스트 데이터셋 : 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



■ 발표 주제 선정

- 논문, 인터넷, 책자 등 활용 → 출처를 **필히** 명시할 것
- 아래 주제나 자율주제로 10분 발표



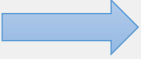

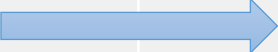

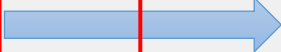

김정식(8)

프로젝트	주제	세부내용	발표자(2인)
#1	Data augmentation	geometric, photometric, ...	이치호, 최현동
	CNN 분류와 회귀	분류 모델과 회귀 모델의 구조	김정식(16), 오동엽
	최적화 알고리즘	GD, Momentum, Adam, ... / batch size – SGD	김봉균,
	과적합 방지 규제화	BN, Dropout, ...	이찬희, 이민수
	하이퍼파라미터 조정	lr, batch size, epochs 등 (자동화 : Optuna)	오혜민, 오세광
#2	YOLO hyps	Setting hyperparameters	이준혁, 사수진
	YOLO train arguments	imgsz, epoch, ...	김혜영, 최은주
	YOLO data augmentation	mosaic, mixup, ...	이선명, 김정호
	Data labeler (Tool)	Annotation Tools	이선경, 김영명
자율주제	객체 분할	U-net	이형욱,
	모델 경량화	Tensorrt	조성현,

조	조원	진행상황 점검	Milestones
1	이치호, 이선명, 김정호	환경설정 완료, 데이터셋 증량, PyTorch	Data 파악 및 증량
2	이형욱, 최은주, 오혜민	환경설정 완료, 데이터 증강, CNN 설계, 모델 설계, 학습 PyTorch	딥러닝 개발환경 구축
3	이선경, 김정식(16)	환경설정 완료, 데이터 증강(이), CNN 설계 중(김)	CNN 모델 설계
4	최현동, 이찬희, 사수진	환경설정 완료, 데이터 증강, CNN 설계중	모델 학습 및 추론
5	김영명, 오세광, 김정식(8)	환경설정 완료, 데이터 증강 진행중 (전원불참)	혼동행렬 및 시각화
6	오동엽, 이민수, 조성현	환경설정 완료, 데이터 증강, CNN 설계중	비교 실험 (모델 구조, hyperparameters, 전이학습 등)
7	이준혁, 김혜영, 김봉균	환경설정_Colab, 환경설정, 데이터 증량	발표자료 작성

주차	날짜	발표 주제	비고
1	3/04	[강의] 교과목 소개 및 딥러닝 / 조 편성, 주제발표 → [과제] 논문리뷰	비대면수업
2	3/11	[강의] Project #1: CNN 분류(데이터 공개), 노트북 대여 → [과제] 중간고사 풀이	대면수업(1)
3	3/18	[강의] 딥러닝 개발환경(CUDA) 구축, Dataset 설명, 조별 토의 및 멘토링	비대면수업
4	3/25	[강의] OpenCV와 Matplotlib 시각화, 조별 토의 및 멘토링	비대면수업
5	4/01	[강의] CNN 논문작성법과 적용 사례, 조별 토의 및 멘토링	비대면수업
6	4/08	프로젝트 점검(사전발표), 조별 토의 및 멘토링	비대면수업
7	4/15	프로젝트 최종점검(사전발표), 조별 토의 및 멘토링	비대면수업
8	4/22	Project #1 발표평가	대면수업(2)
9	4/29	[강의] Project #2 : YOLO를 이용한 객체 검출 → [과제] 기말고사 풀이	대면수업(3)
10	5/06	[동영상 강의] YOLO 모델 개발 실습 → [과제] 논문리뷰	비대면수업
11	5/13	가디언 모임으로 수업 대체 (불참시 결석 처리)	비대면수업
12	5/20	[강의] YOLOv8과 객체분할, 조별 토의 및 멘토링	비대면수업
13	5/27	주제발표	비대면수업
14	6/03	프로젝트 최종점검(사전발표), 조별 토의 및 멘토링	비대면수업
15	6/10	Project #2 발표평가	대면수업(4)

일정검토

항 목	1주	2주	3주	4주	5주	6주	7주	8주	비 고
	3/4~	3/11~	3/18~	3/25~	4/1~	4/8~	4/15~	4/22~	
	논문리뷰		개인발표 주제선정	중간고사 문제풀이				중간발표	
Data 파악 및 증량								 프로젝트#1_발표 자료양식	
딥러닝 개발환경 구축									
CNN 모델 설계									
모델 학습 및 추론									
혼동행렬 및 시각화									
비교 실험 (모델 구조, hyperparameters, 전이 학습 등)									
발표자료 작성									



Q & A

감사합니다

