RecyNET COMPANY

AI 기반 폐기물 분리배출 효율화
프로젝트 제안서

이지호, 최●●, 정●●, 한●●

2025.12.31

RecyNet

(Recycling + Neural Network)

재활용 / 폐기물 문제를 신경망으로 해결한다!

RecyNET COMPANY

Table of Contents

폐기물 분류 당위성

Why
- 행정 혼선과 처리 비효율
- 수도권 생활폐기물 직매립 금지(2026) 시행

What
- 생활폐기물의 정확한 분류
- 환경 부담과 행정력 낭비 최소화

How
- AI 기반 폐기물 자동 분류 모델
- 처리량 감소 → 소각시설 부담 완화

Who
- 지자체 및 행정기관
- 지역 폐기물 처리 업체

Where
- 수도권 : 공공 소각시설 부족
- 지방: 수도권 폐기물 처리 수요가 집중될 가능성

When
- 2026년부터 수도권 생활폐기물 직매립 금지
- 연간 약 51만 톤의 폐기물 처리 방식이 전환

문제 1 매립지 인프라 부족

문제 2 소각장 운영 비효율성

문제 3 지역별로 상이한 음식물 쓰레기 처리 정책

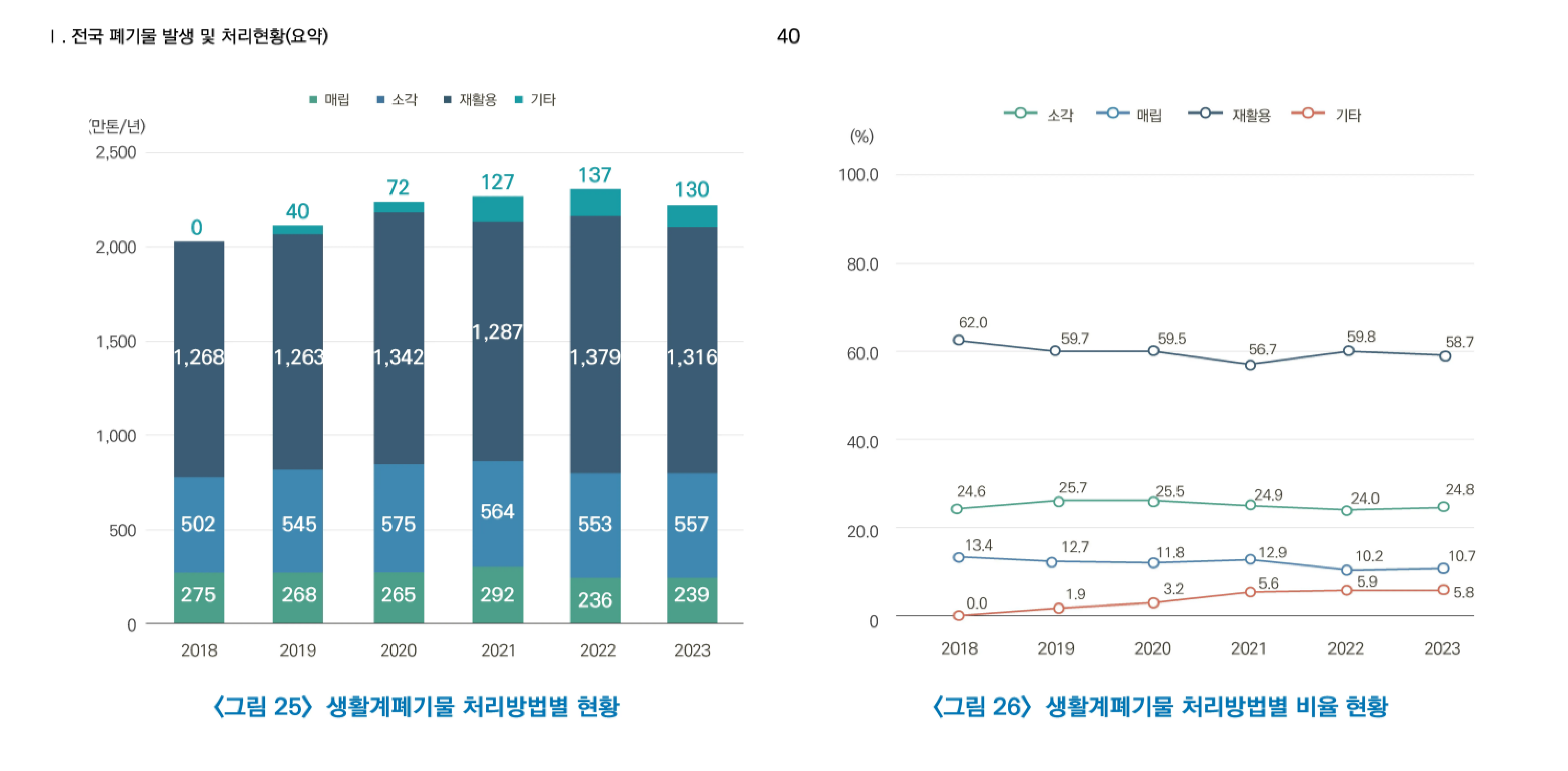
AI 기반으로 분리배출 정확도를 향상시켜 자원순환을 촉진한다.

지역별 폐기물 처리 기준 혼선을 해소하여 도시 폐기물 관리를 효율화한다.

공공문제 해결형 AI 분류 인프라 구축으로 산업 성장에 기여한다.

폐기물의 소각 및 매립 감소를 통해 기후변화 대응에 기여한다.

문제 1 매립지 인프라 부족



〈그림 25〉 생활계폐기물 처리방법별 현황

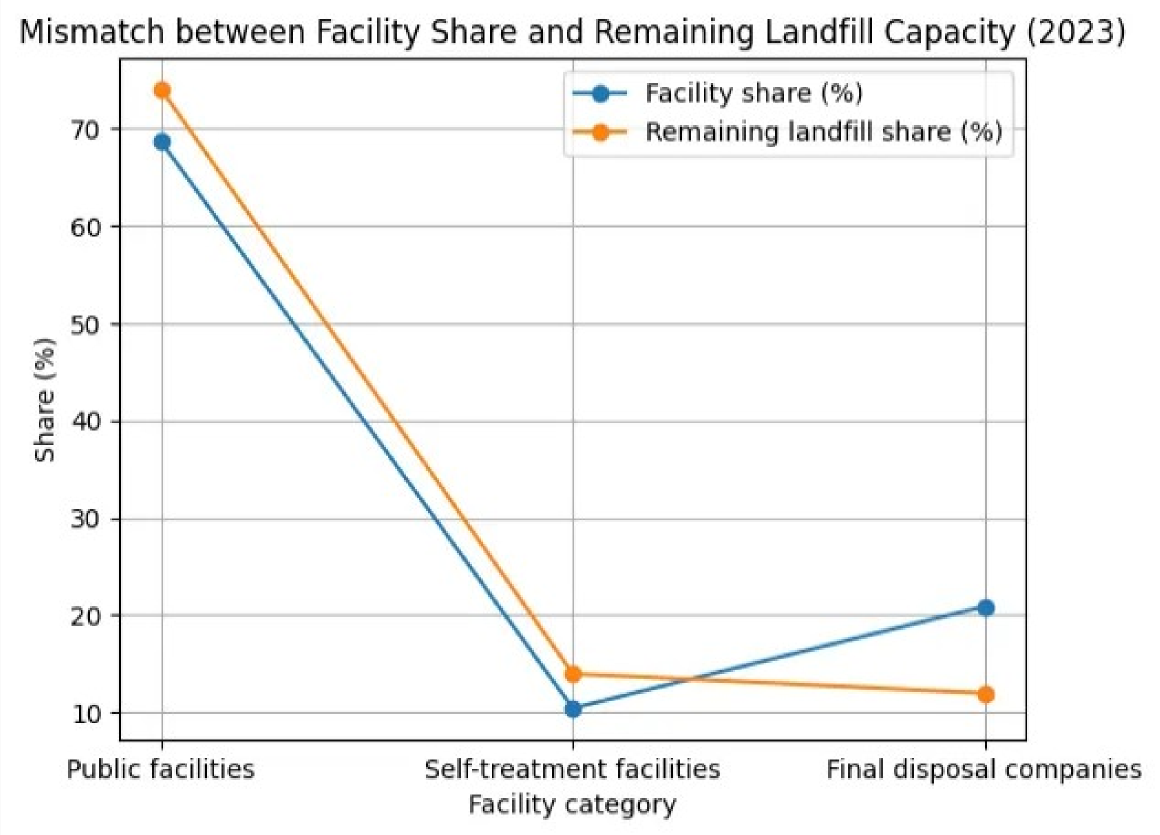
〈그림 26〉 생활계폐기물 처리방법별 비율 현황

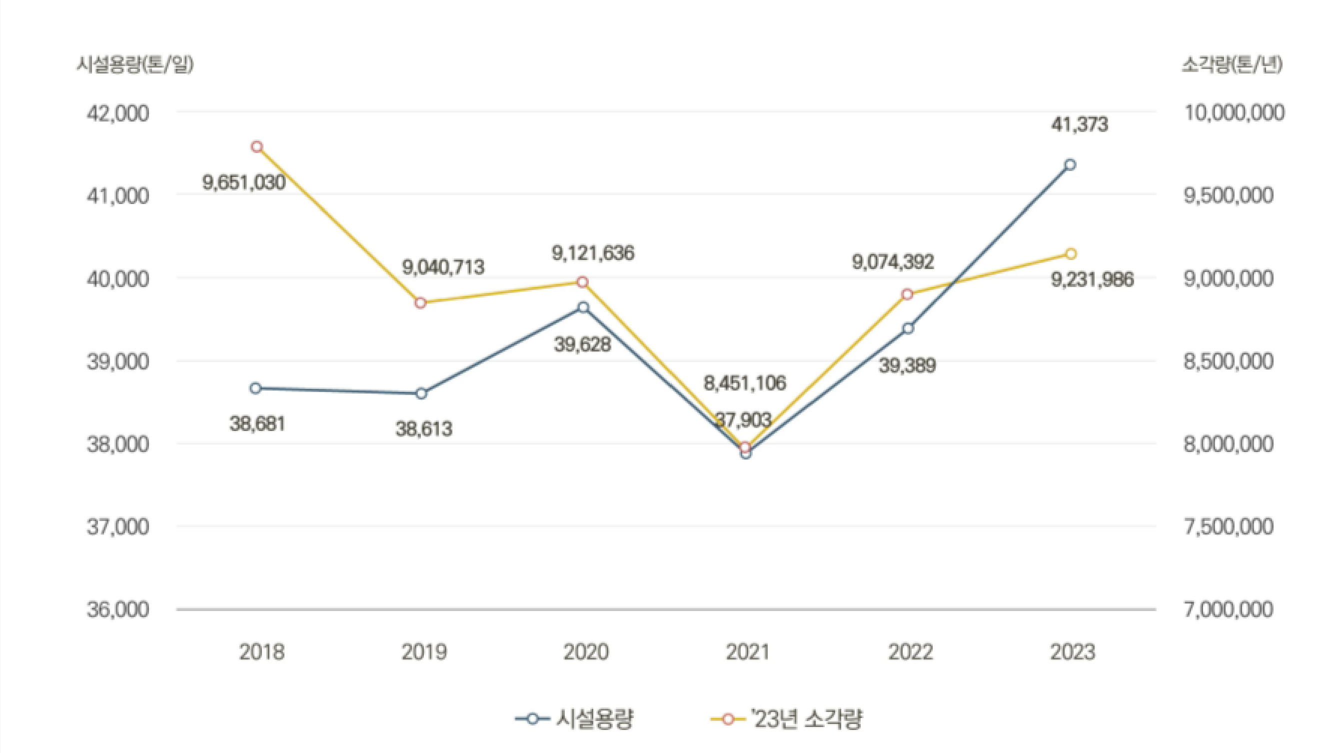
세부 내용

2026년 수도권(서울·인천·경기) 생활폐기물은
직접 매립이 법적으로 금지

연간 약 51만 톤(t) 수준의
생활폐기물 처리 방식이 바뀜

문제 2 소각장 운영 비효율성





〈그림 38〉 소각시설 현황 연도별 추이

세부 내용 전국 소각시설은 연간 약 1,510만 톤 처리가 가능하나, 실제 처리량은 923만 톤에 그침

약 38.9% 정도의 시설이 비효율적으로 운영되고 있으며, 이는 환경 및 정책 제약으로 인한 현상

문제 3 지역별로 상이한 음식물 쓰레기 처리 정책

구분	인천시 기준	서울시 기준	차이 요약
채소류 뿌리	명시 없음	쪽파·대파·미나리 뿌리	서울만 명확히 규정
곡류	명시 없음	왕겨	서울만 포함
생선뼈	명시 없음	생선뼈	서울만 포함
알껍질	명시 없음	달걀·오리알·메추리알·타조알	서울만 포함
독성 음식물	명시 없음	복어 내장	서울만 명시
기타 이물질	명시 없음	껌·비닐·플라스틱·금속·유리·의류 등 상세 나열	서울시가 생활 이물질을 광범위하게 포함

발생할 수 있는 문제

이물질 혼입으로 인한 분류 기기 파손

행정력 낭비

미세 플라스틱 유입 등

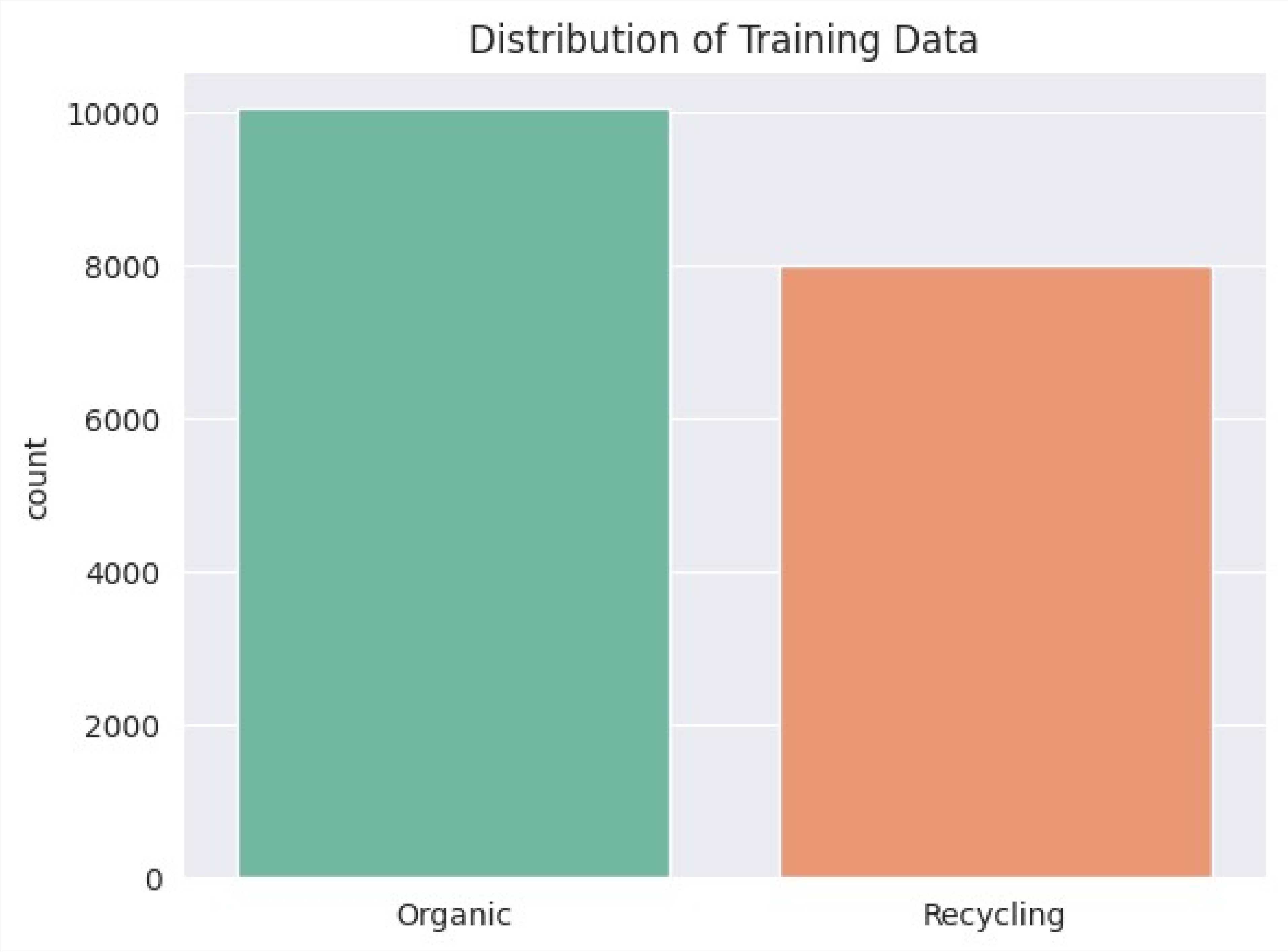
25,000 여 개의 폐기물 이미지

데이터 노이즈

유기물 데이터에 유기물이라 보기에 어려운
데이터가 다소 존재함

학습에 지장이 있을것으로 판단되었지만
전략적인 전처리 및 최적화로 데이터 노이즈 최소화

train-test 데이터 간 데이터 종류 분포가
상이함



학습 데이터 불균형

Organic 12,565개

Recycling 9,999개

Recycling에 가중치 부여
→ 1 : 1.25 비율로 class_weight

학습 데이터 조건 설정

Train Data 분리

- Train Data(80%) 18,054
- Val Data(20%) 4,516

Test Data 유지 2,516

고려사항(요구사항)	요구반영
CNN모델 선정	CNN(custom), VGG16, RegNet, YOLOv12
Epochs 조정	각자 하드웨어 인프라 성능을 고려하여 조정하되 optimizer 별 필요성에 따라 12~30회 내에서 조정
흑백/컬러 처리	질감 판별, 모델 성능, 학습 소요시간 등의 애로를 고려하여 전부 컬러로 분석하기로 결정함
Learning rate 조정	모델별로 CNN 표준이라 알려진 0.0001(1e-4) 외에도 모델 및 optimizer 별 최적 LR 반영하여 비교실험
ImageDataGenerator	학습시간이 상당히 소요되며 데이터가 충분하다고 판단하여 활용하지 않는것으로 결정함
Optimizer	각자 맡은 모델별로 adam, SGD, RMSprop, Momentum 중 최적 optimizer를 선택하거나 전부 수행하여 비교함
데이터 전처리	모델별로 최적화 된 전처리 과정을 수행함

모델별 데이터 전처리 CNN

Resize 224 * 224

Rescale 1 / 255

CNN

01 Learning Rate 조정 (0.0001 / 0.001)

02 Optimizer (RMSprop, Adam, Nadam, SGD, Momentum / Nesterov)

03 Epoch 20 고정

CNN 지표 분석

Learning rate 0.0001

	RMSprop	Adam	Nadam	SGD	Mometum / Nesterov
Accuracy (정확도)	0.859530	0.922005	0.894548	0.848786	0.871070
Precision (정밀도)	O 0.942568 R 0.785553	O 0.930050 R 0.911871	O 0.945141 R 0.842361	O 0.862314 R 0.831522	O 0.860201 R 0.887033
Recall (재현율)	O 0.796574 R 0.938849	O 0.930050 R 0.911871	O 0.860814 R 0.937050	O 0.867238 R 0.825540	O 0.917916 R 0.812050
F1-score	O 0.863443 R 0.855387	O 0.930050 R 0.911871	O 0.901009 R 0.887186	O 0.864769 R 0.828520	O 0.888122 R 0.847887
ROC-AUC	0.953507	0.972796	0.962861	0.910443	0.927466
Inference Time (추론 시간)	7.264136	4.771368	4.286859	4.456256	3.848110

CNN 지표 분석

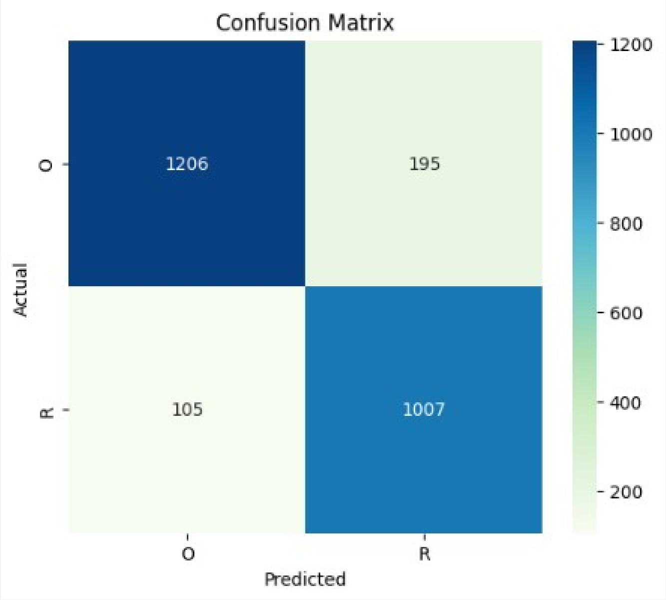
Learning rate 0.001

	RMSprop	Adam	Nadam	SGD	Momentum / Nesterov
Accuracy (정확도)	0.891365	0.905690	0.891364	0.868683	0.897732
Precision (정밀도)	O 0.929878 R 0.849292	O 0.885942 R 0.935323	O 0.929878 R 0.849292	O 0.838710 R 0.919528	O 0.892857 R 0.904447
Recall (재현율)	O 0.870807 R 0.917266	O 0.953605 R 0.845324	O 0.870807 R 0.917266	O 0.946467 R 0.770683	O 0.927909 R 0.859712
F1-score	O 0.899373 R 0.881971	O 0.918529 R 0.888049	O 0.899373 R 0.881971	O 0.889336 R 0.838552	O 0.910046 R 0.881512
ROC-AUC	0.953747	0.959686	0.953747	0.926187	0.951877
Inference Time (추론 시간)	4.516964	4.200357	4.972496	3.713852	4.546007

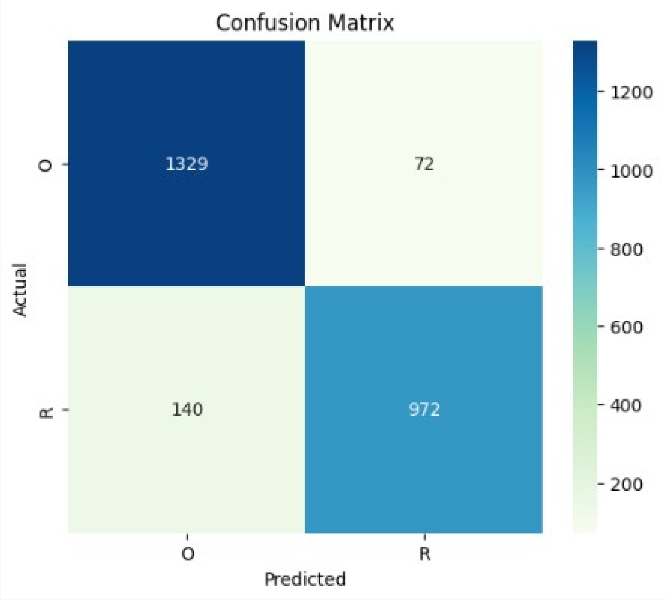
CNN 혼동 행렬

Learning rate 0.0001

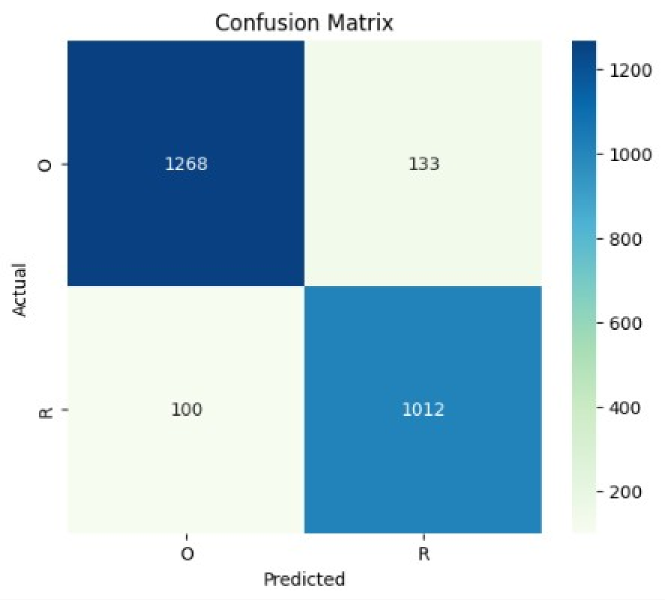
RMSprop



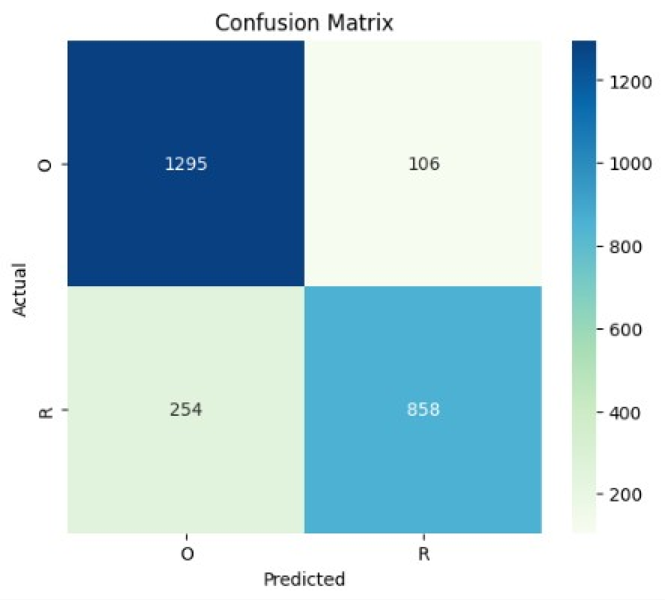
Adam



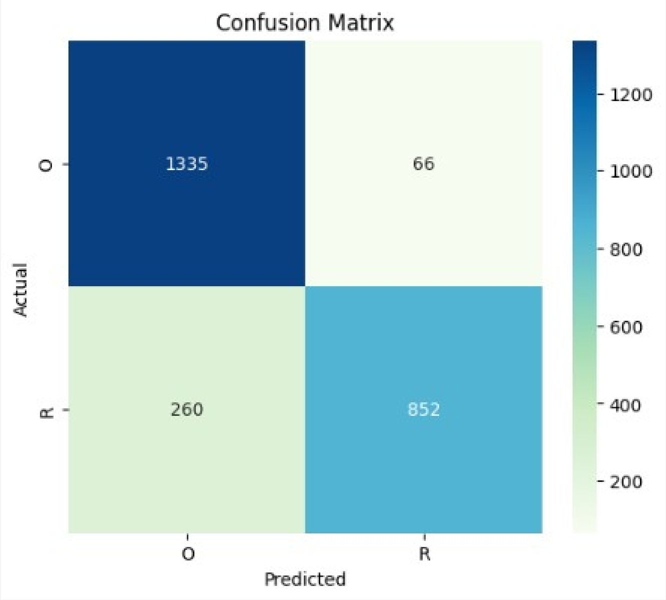
Nadam



SGD



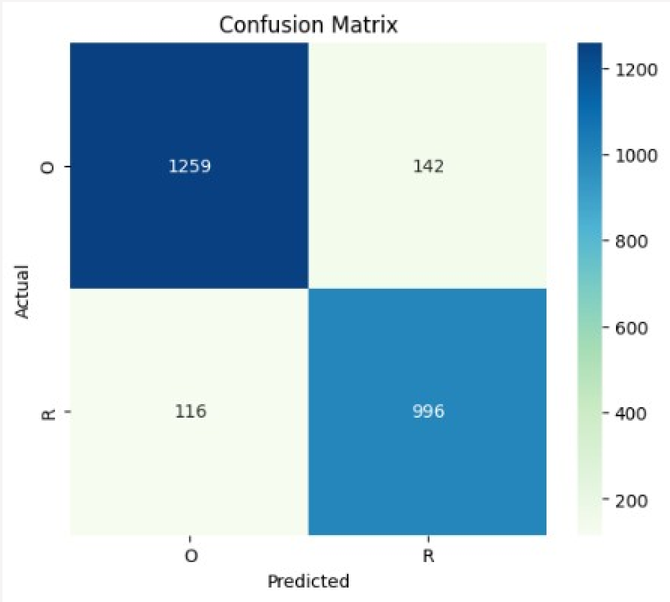
Momentum



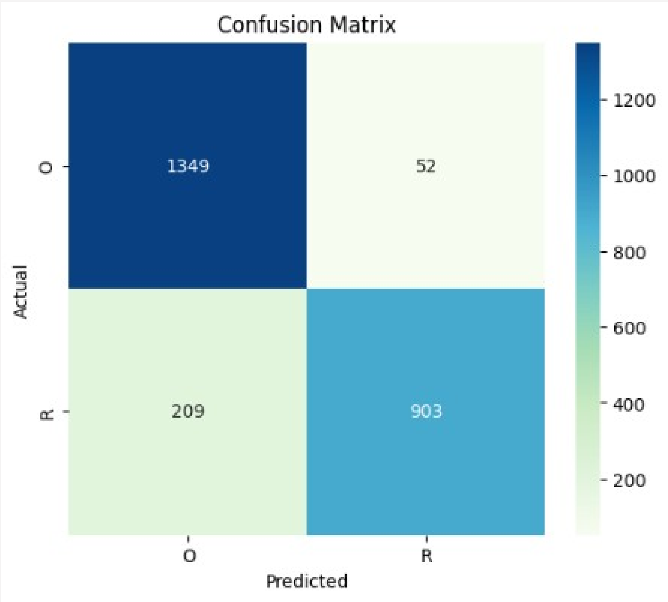
CNN 혼동 행렬

Learning rate 0.001

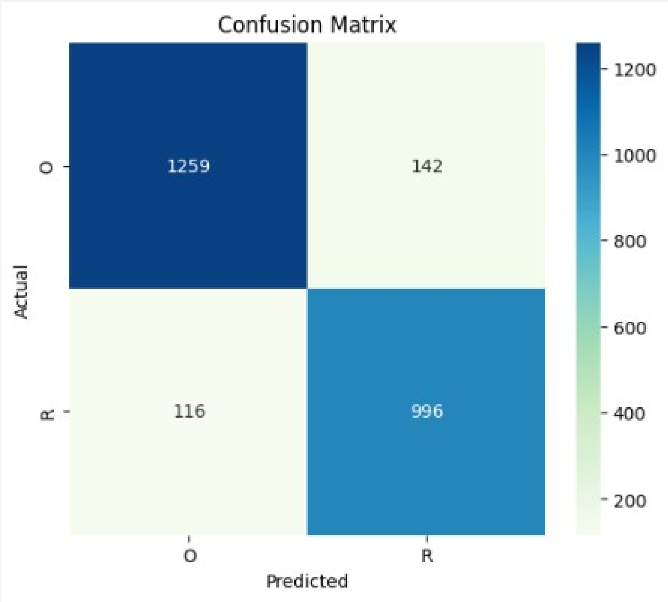
RMSprop



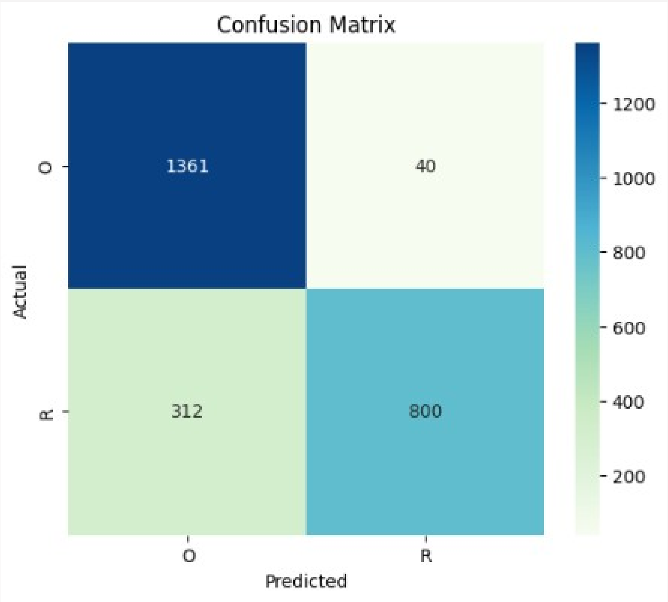
Adam



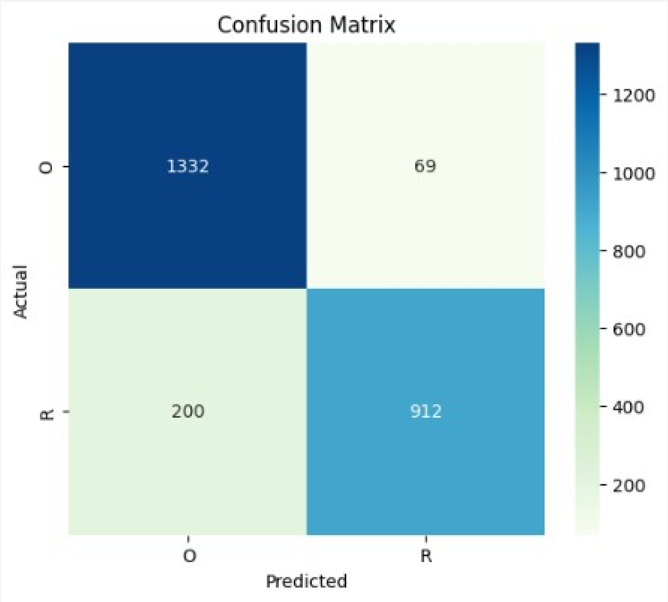
Nadam



SGD



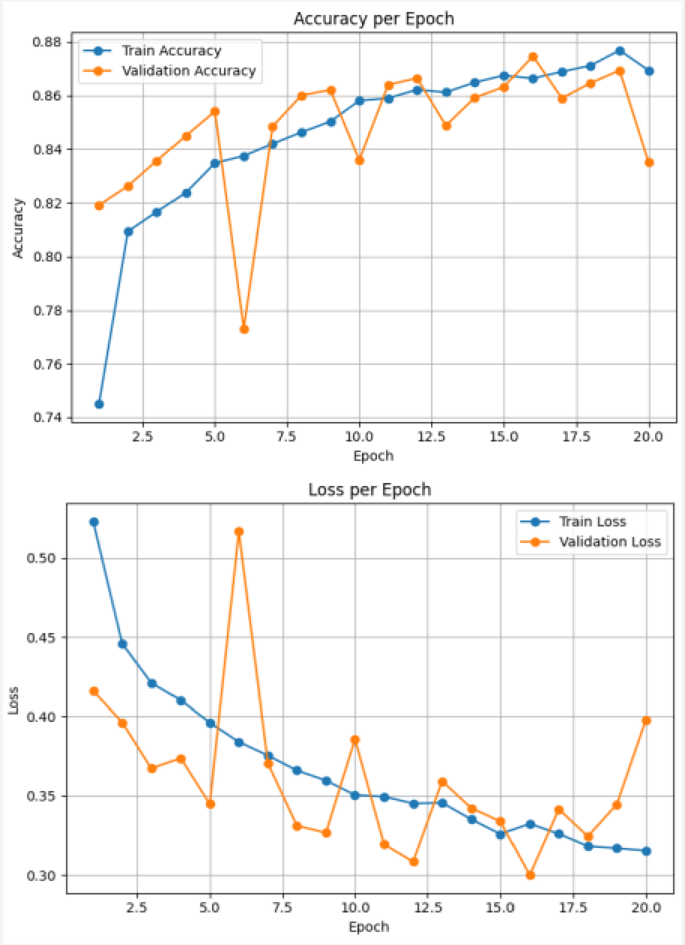
Momentum



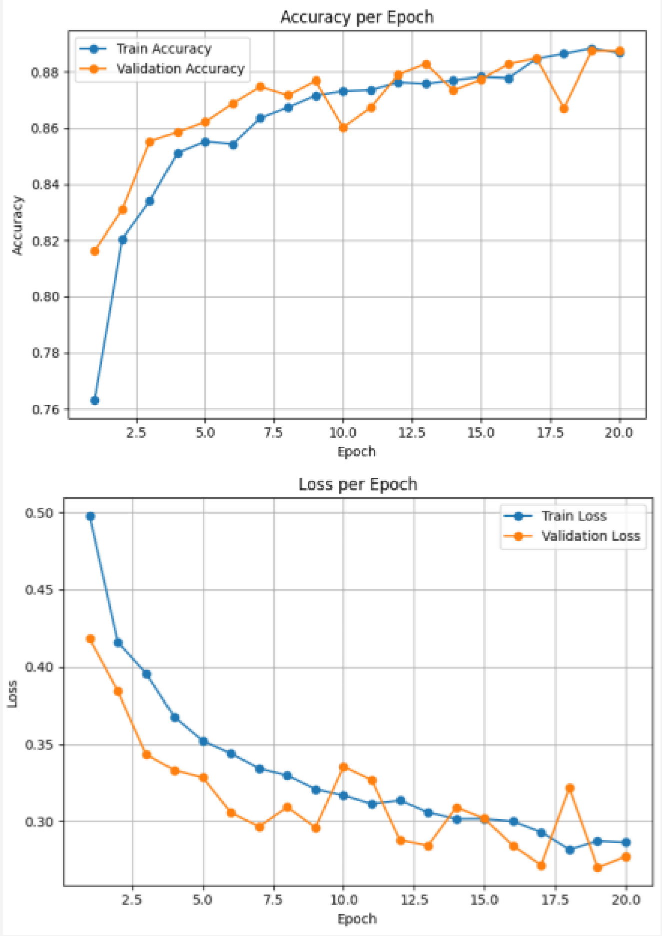
CNN Accuracy & Loss 그래프

Learning rate 0.0001

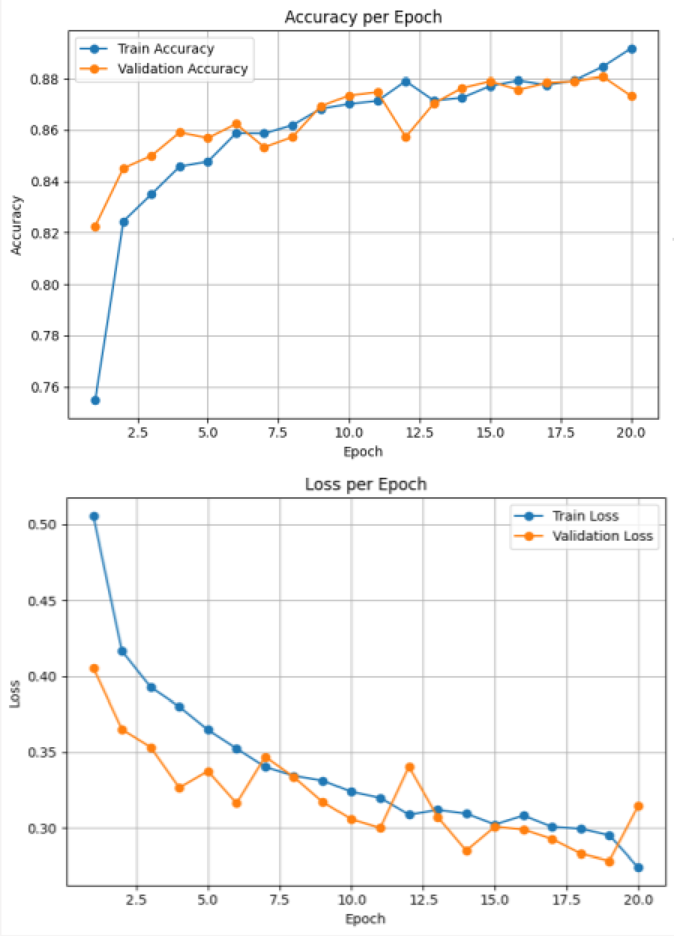
RMSprop



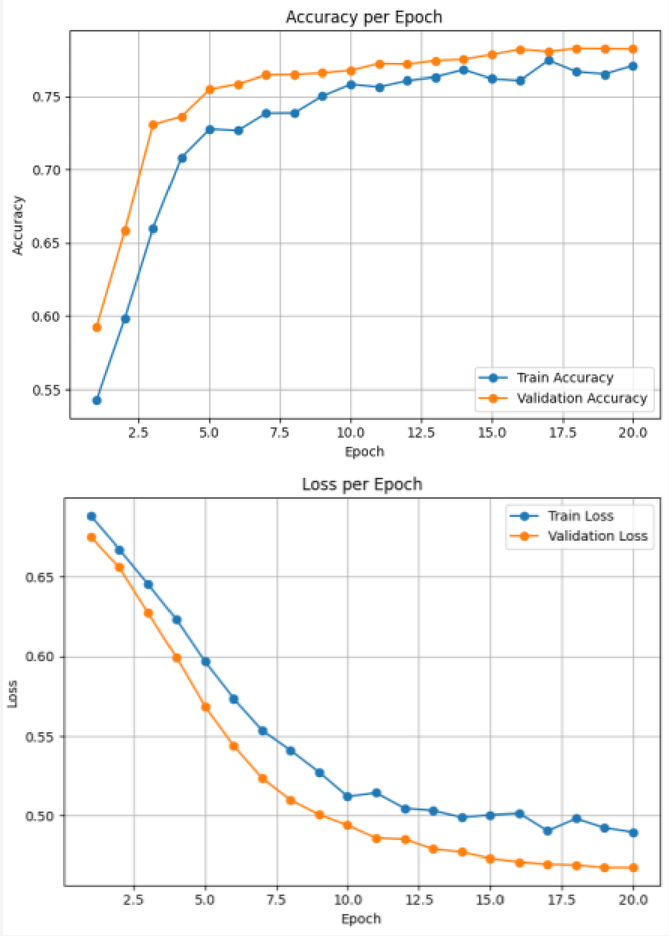
Adam



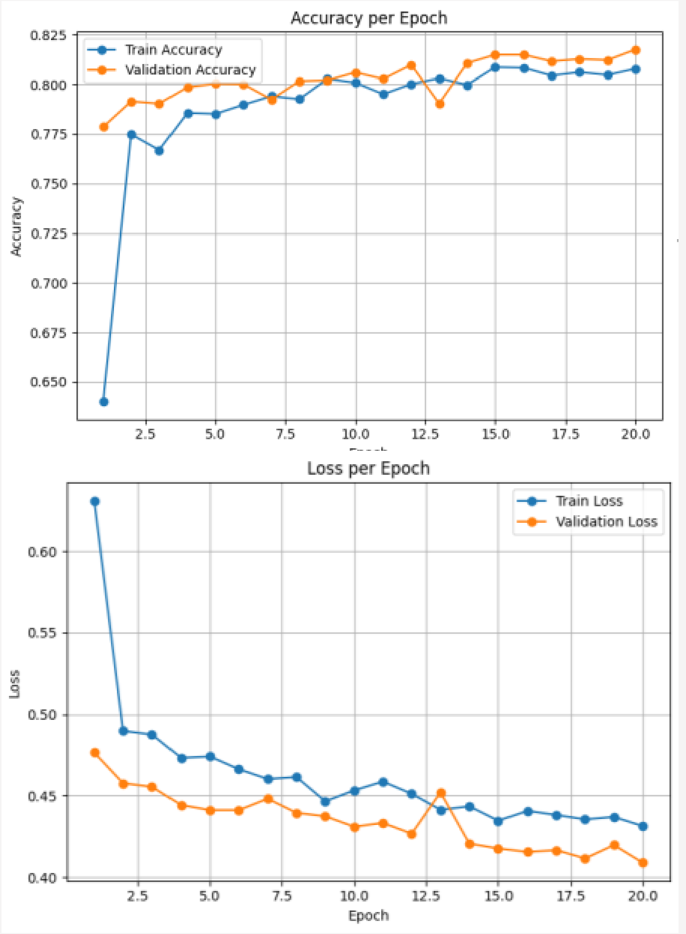
Nadam



SGD



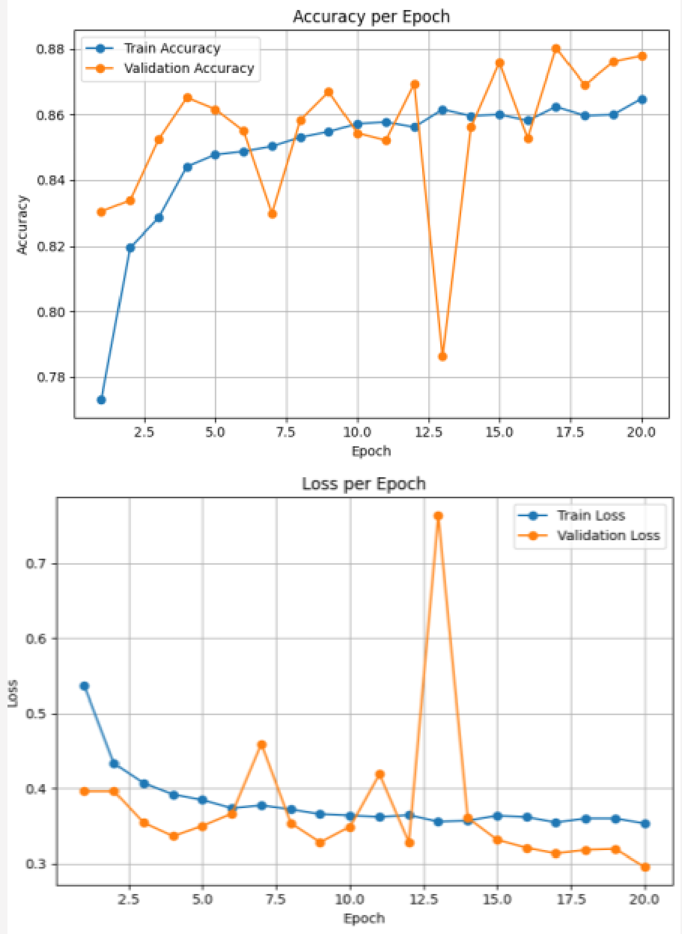
Momentum



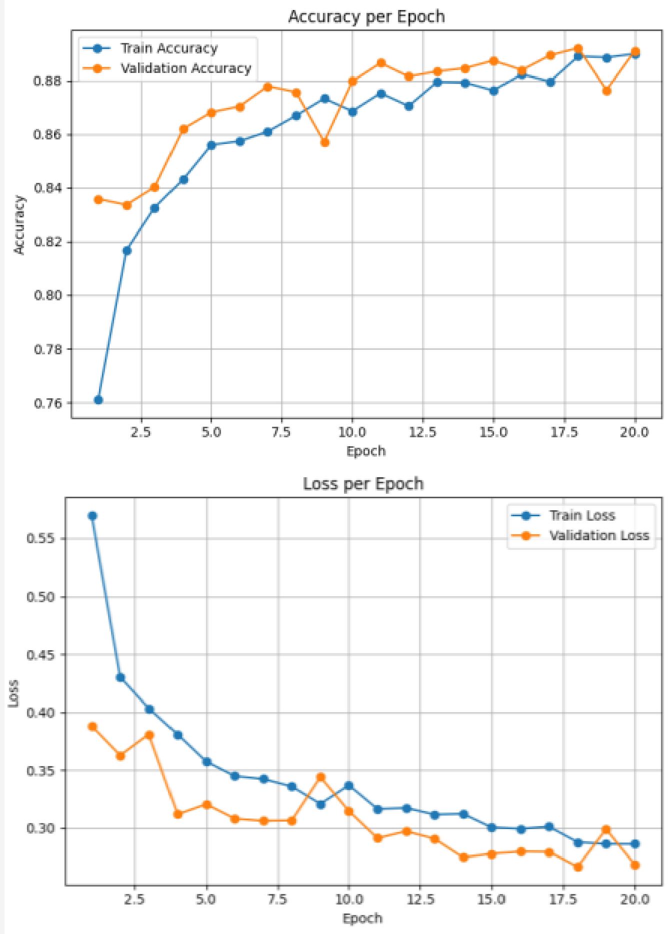
CNN Accuracy & Loss 그래프

Learning rate 0.001

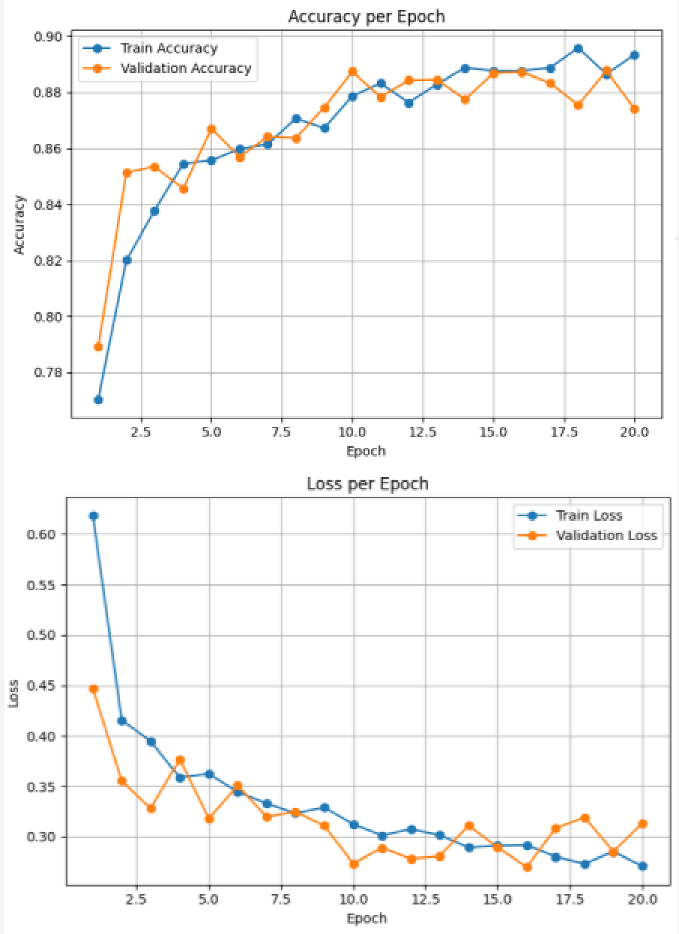
RMSprop



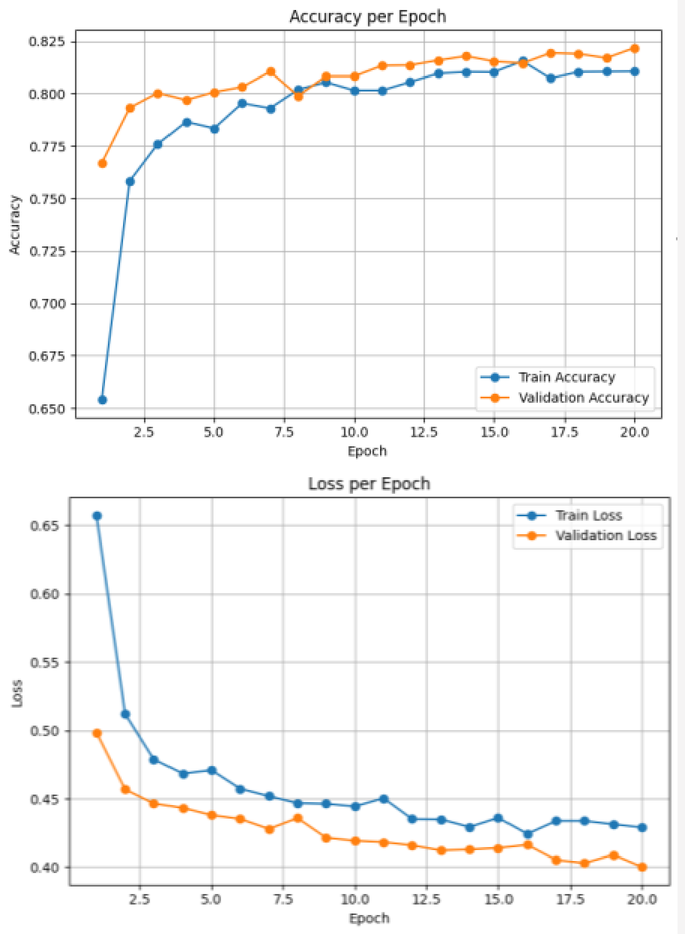
Adam



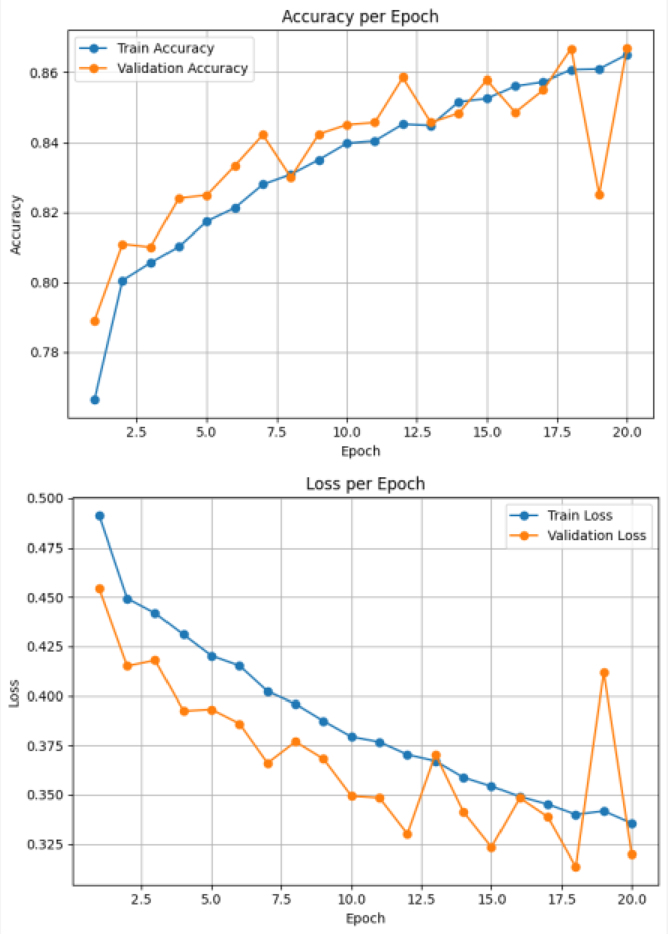
Nadam



SGD



Momentum

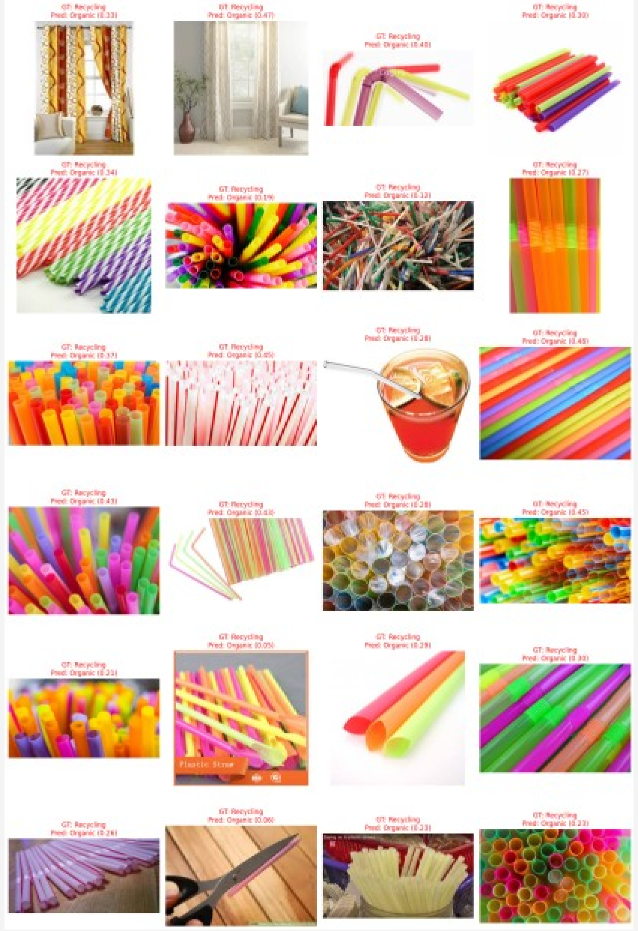


CNN 틀린 추론 사진(120장)

Learning rate 0.0001

Adam / 잘못 분류된 이미지 수: 212




모델별 데이터 전처리 RegNet

Resize	224 * 224
Rescale	1 / 255
Normalize	mean = 0.485, 0.456, 0.406 std = 0.229, 0.224, 0.225

RegNet

01 Learning Rate 조정 (Adam) 0.0001, 0.00001
(SGD) 0.01, 0.001

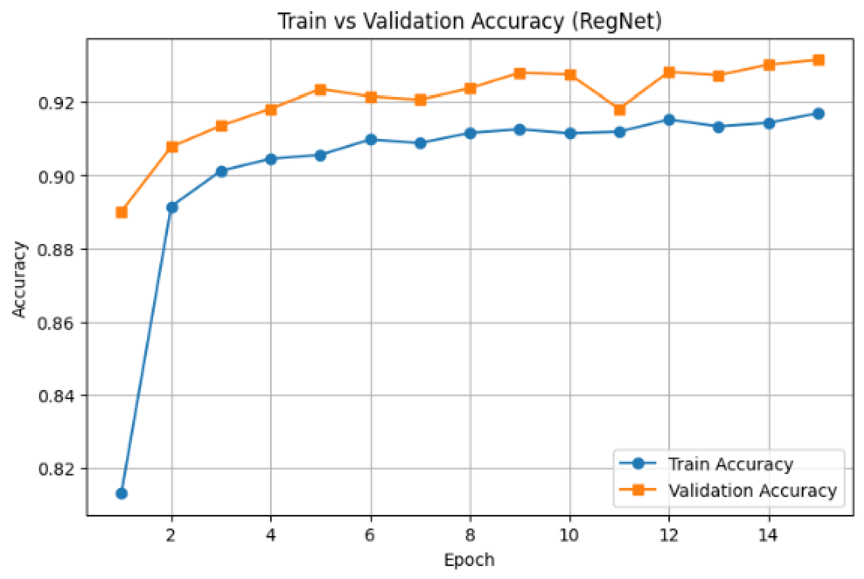
02 Optimizer (Adam, SGD + Momentum)

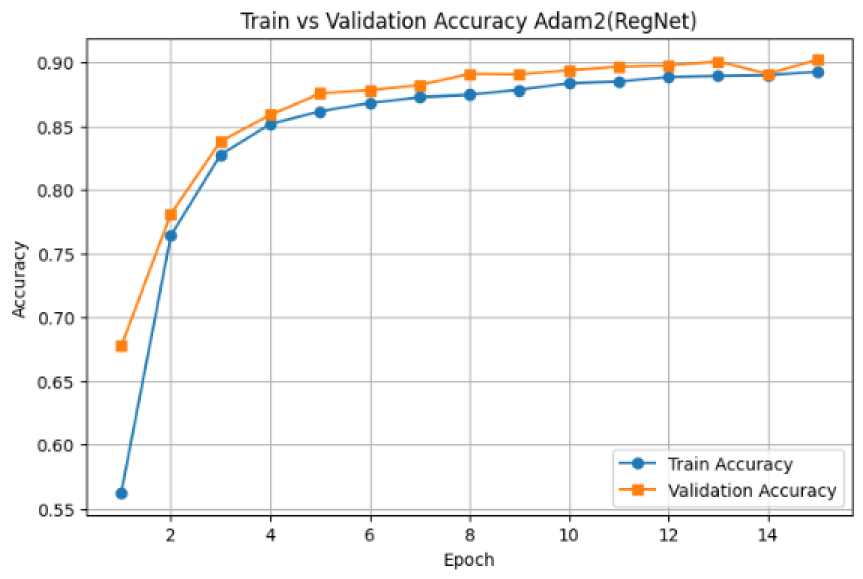
03 Epoch (Adam) 15 (SGD) 15, 30회 비교

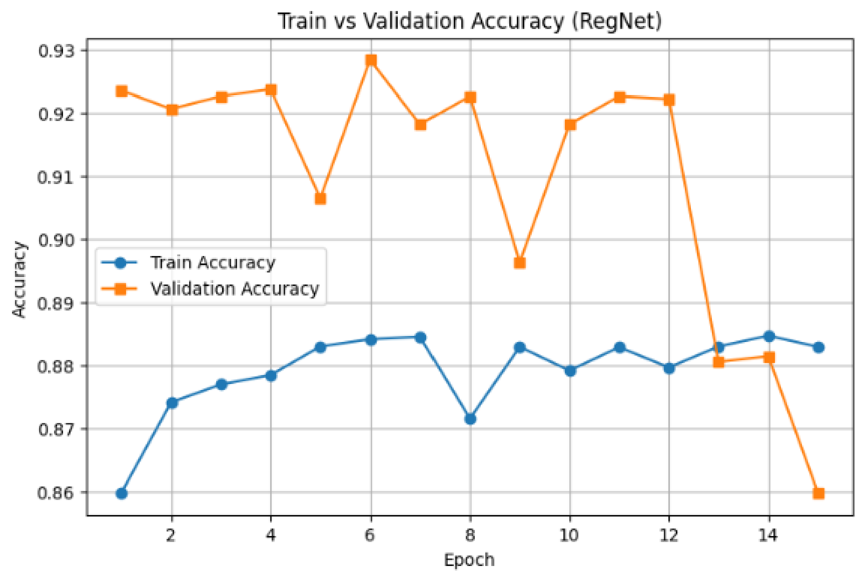
RegNet 지표 분석

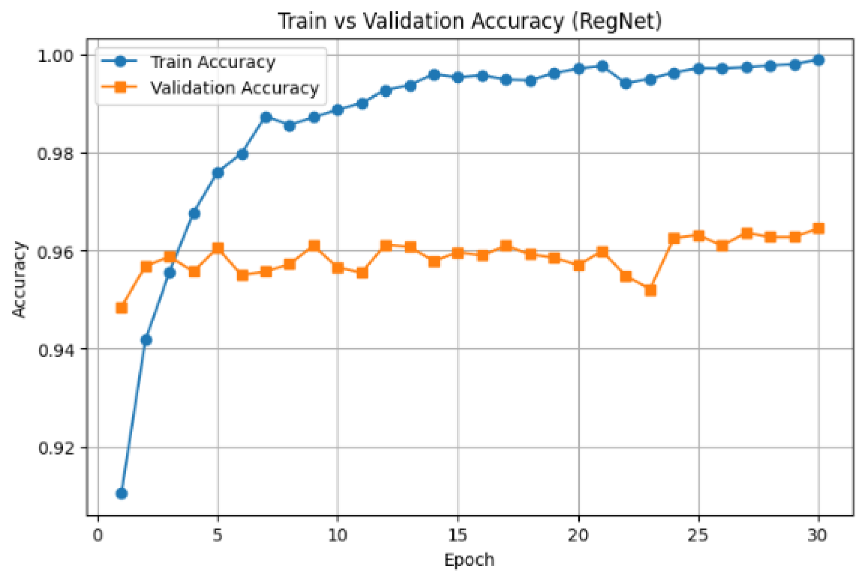
	Adam	Adam(Bat. 32)	SGD + Momentum (15E)	SGD + Momentum (30E)
Learning Rate	0.0001	0.00001	0.01	0.001
Accuracy (정확도)	0.9272	0.9331	0.9168	0.9268
Precision (정밀도)	O 0.8970 R 0.9745	O 0.9214 R 0.9495	O 0.9783 R 0.8564	O 0.8868 R 0.9936
Recall (재현율)	O 0.9822 R 0.8579	O 0.9622 R 0.8966	O 0.8701 R 0.9757	O 0.9957 R 0.8399
F1-score	O 0.9376 R 0.9125	O 0.9413 R 0.9223	O 0.9210 R 0.9121	O 0.9381 R 0.9103
Inference Time (시간) Apple MPS 기준	3.9707	3.8906	4.0867	4.1294

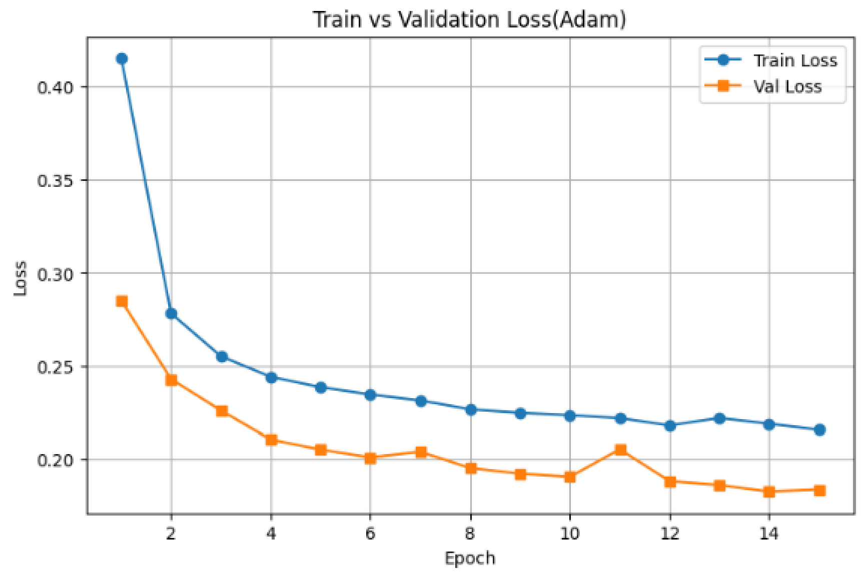
RegNet Accuracy & Loss 그래프

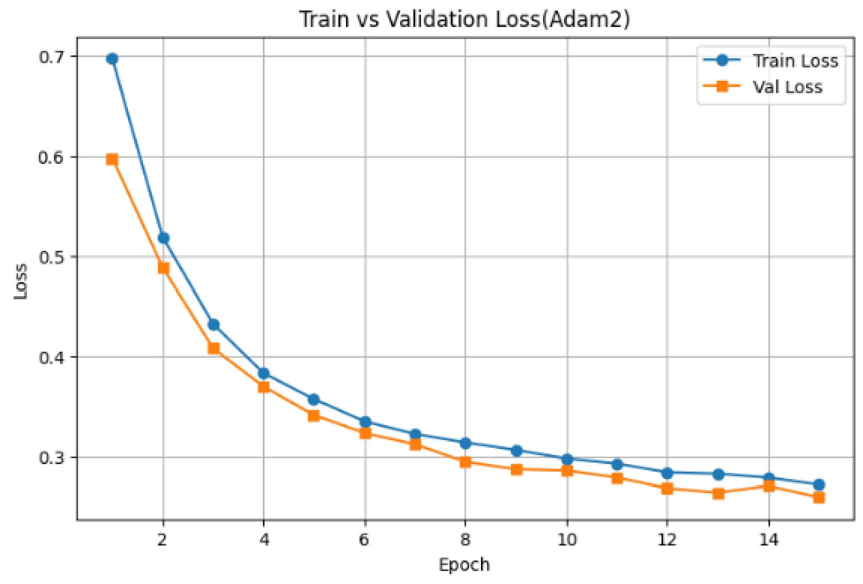


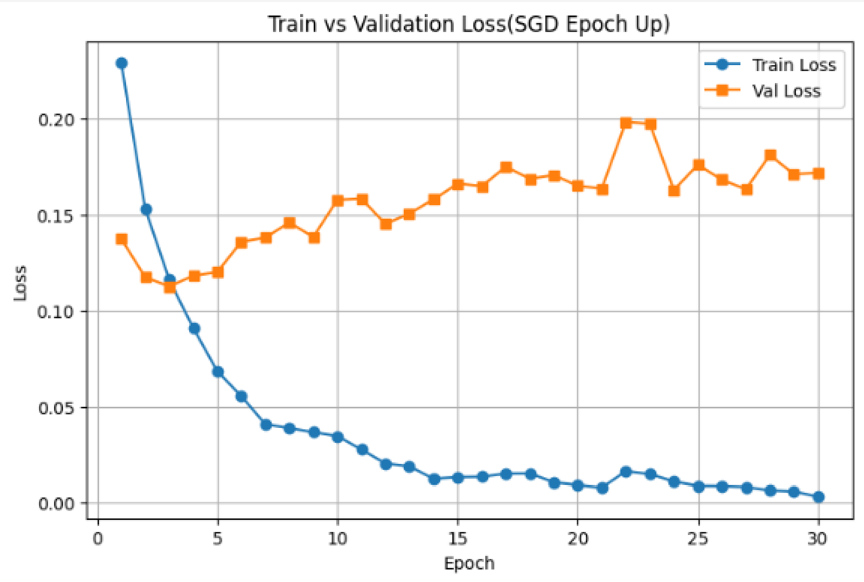




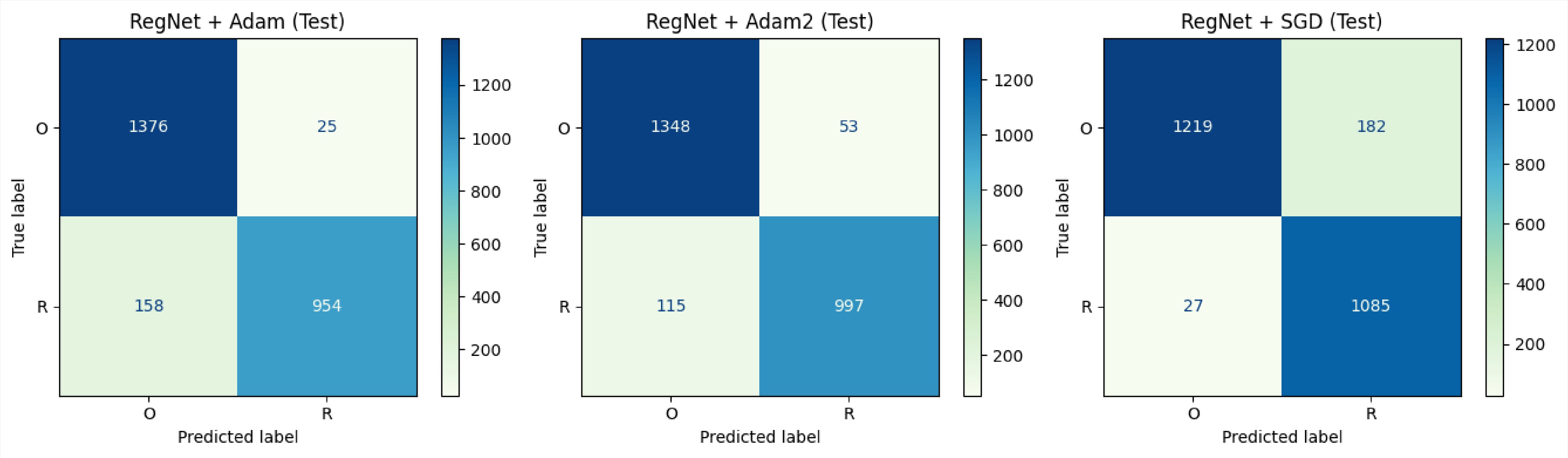


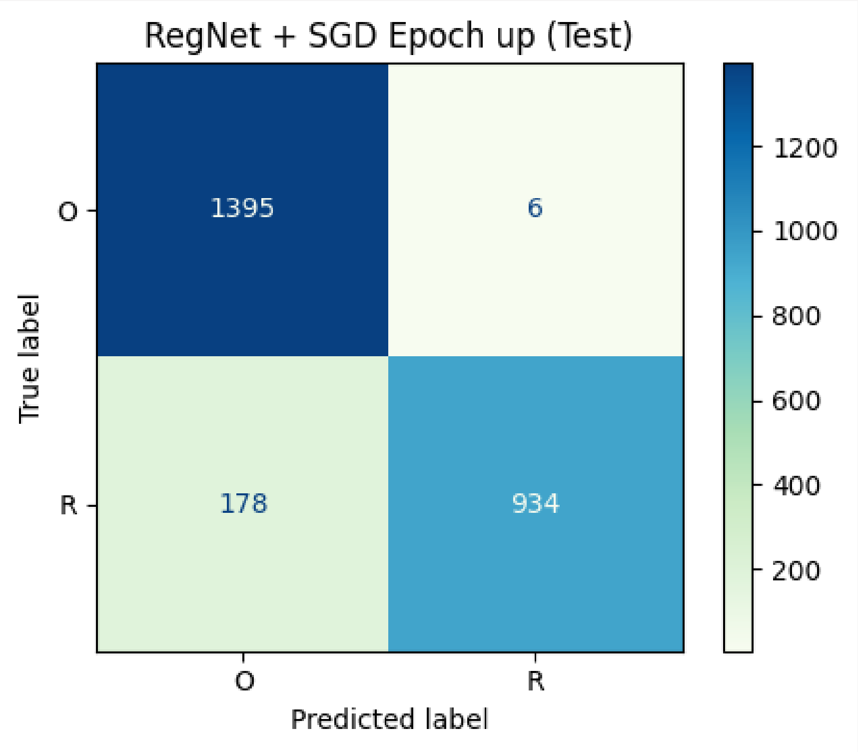




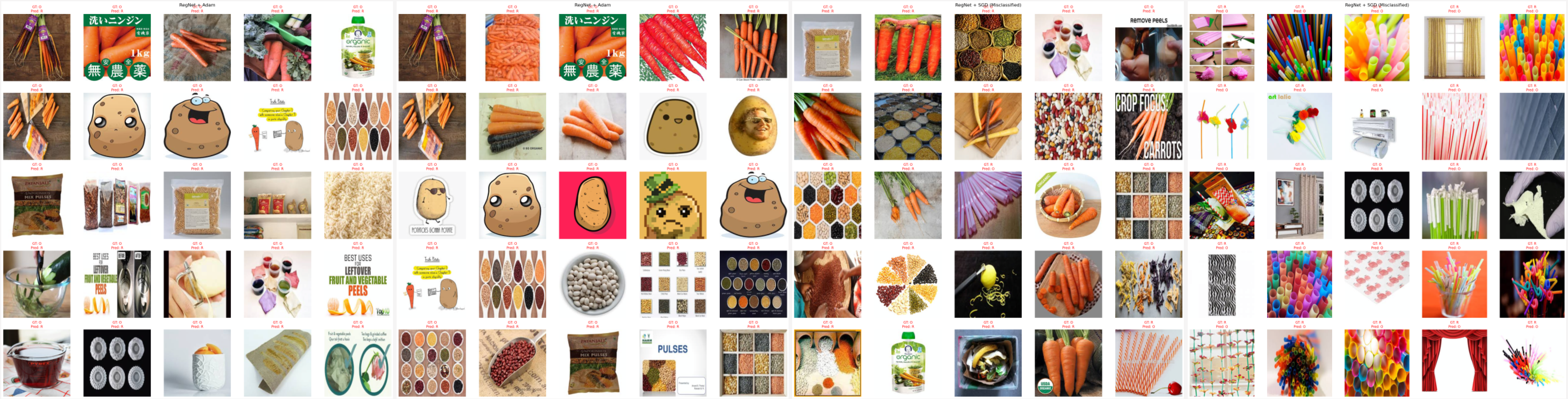


RegNet 혼동 행렬


RegNet + Adam (Test)
True label
Predicted label
O
R
1376
25
158
954
RegNet + Adam2 (Test)
1348
53
115
997
RegNet + SGD (Test)
1219
182
27
1085


RegNet + SGD Epoch up (Test)
True label
Predicted label
O
R
1395
6
178
934

RegNet 틀린 추론 사진(20개 sampling)



모델별 데이터 전처리 YOLOv12

Resize	224 * 224
Rescale	1 / 255 (자동)
Normalize	mean = 필요 없음(기본 설정) std = 필요 없음(기본 설정)

YOLOv12 Fine Tuning

01 Learning Rate 조정 (AdamW) 0.0001, 0.001
(SGD) 0.001, 0.01

02 Optimizer (AdamW, SGD)

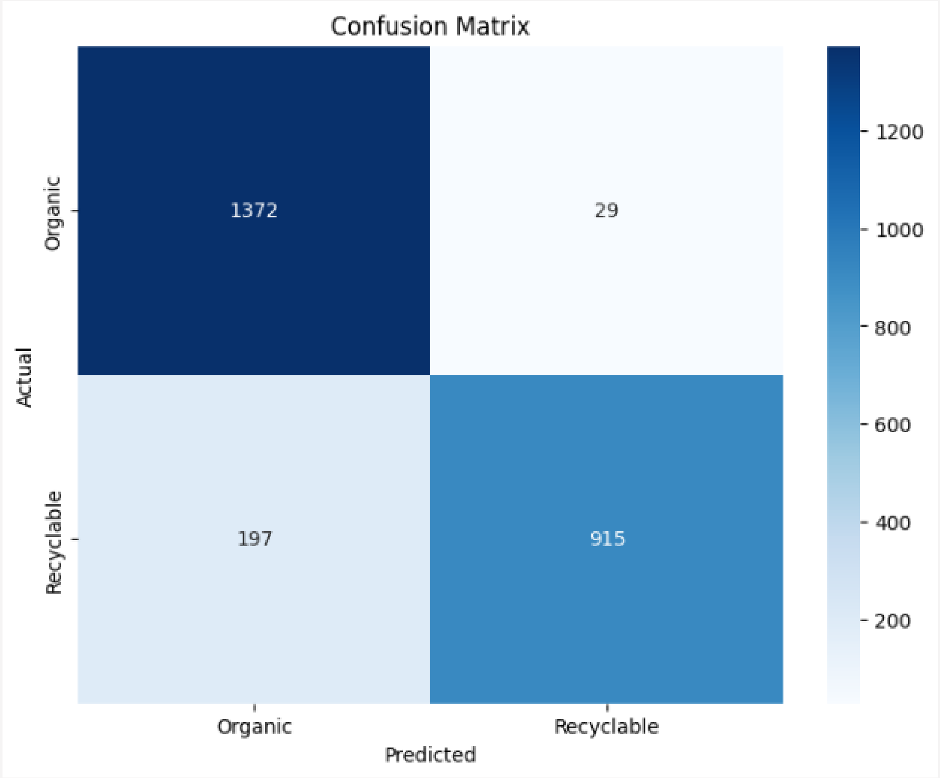
03 Epoch 30회 고정, 가장 가벼운 분류 모델 YOLOv12n_cls 사용

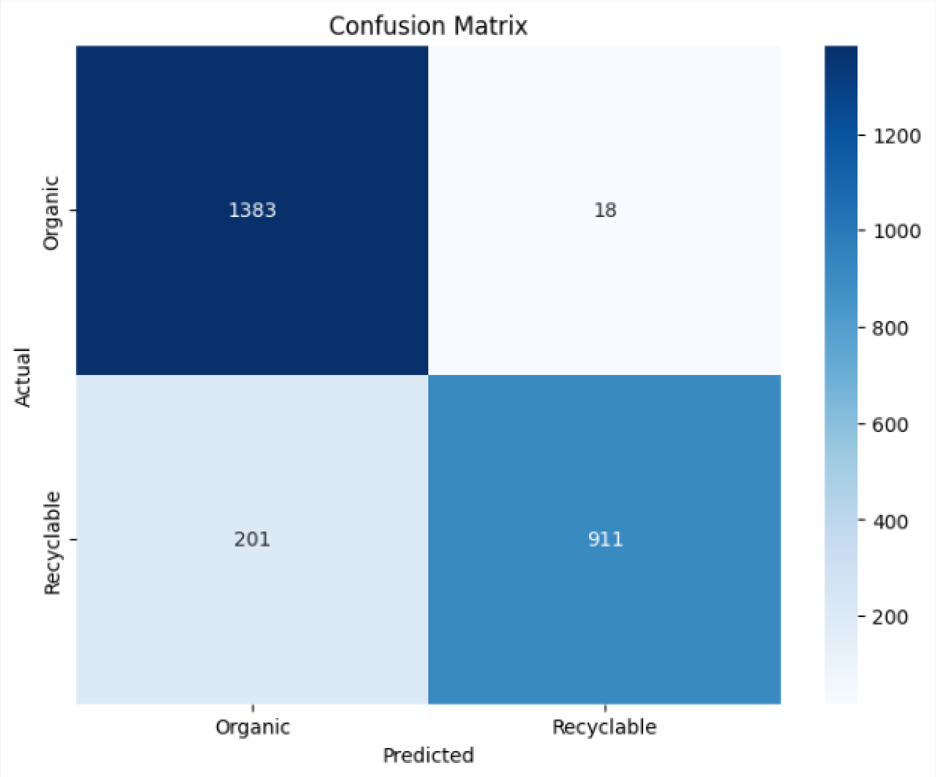
YOLOv12 지표 분석

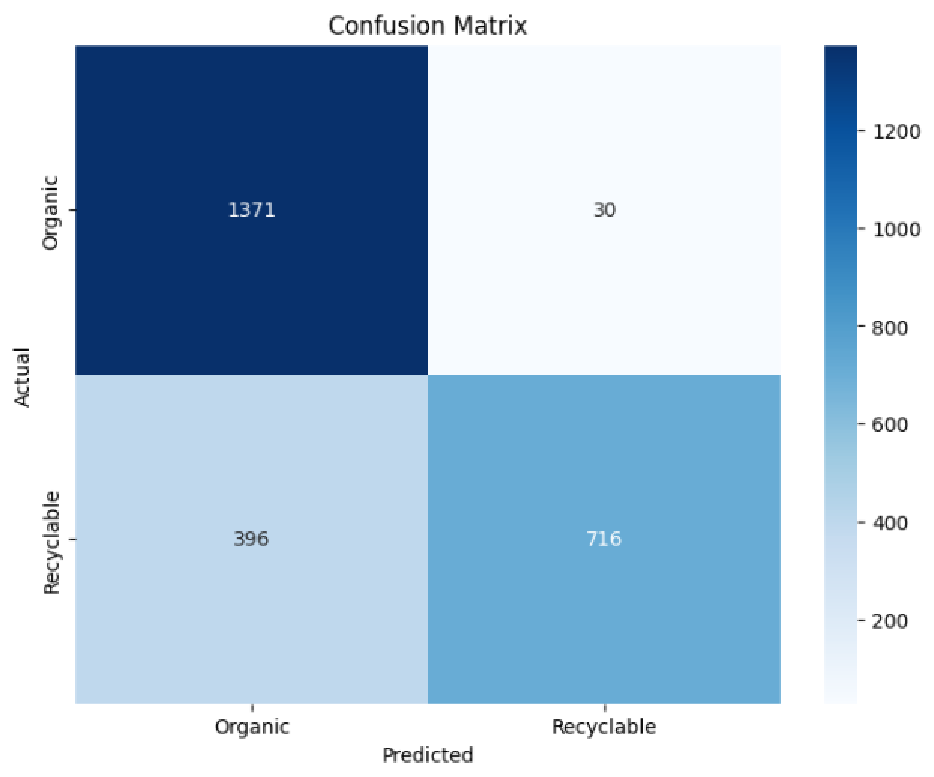
	AdamW	AdamW	SGD	SGD
Learning Rate	0.0001	0.001	~~0.001~~	0.01
Accuracy (정확도)	0.91	0.91	~~0.83~~	0.92
Precision (정밀도)	O 0.87 R 0.97	O 0.87 R 0.98	~~O 0.78~~ ~~R 0.96~~	O 0.89 R 0.98
Recall (재현율)	O 0.98 R 0.82	O 0.99 R 0.82	~~O 0.98~~ ~~R 0.64~~	O 0.98 R 0.85
F1-score	O 0.92 R 0.89	O 0.93 R 0.89	~~O 0.87~~ ~~R 0.77~~	O 0.93 R 0.91
Inference Time (시간) Tesla T4 기준	12.36 초 이미지 당 평균 4.92 ms	10.14 초 이미지 당 평균 4.03 ms	~~11.55 초~~ ~~이미지 당 평균 4.59 ms~~	13.92 초 이미지 당 평균 5.54 ms

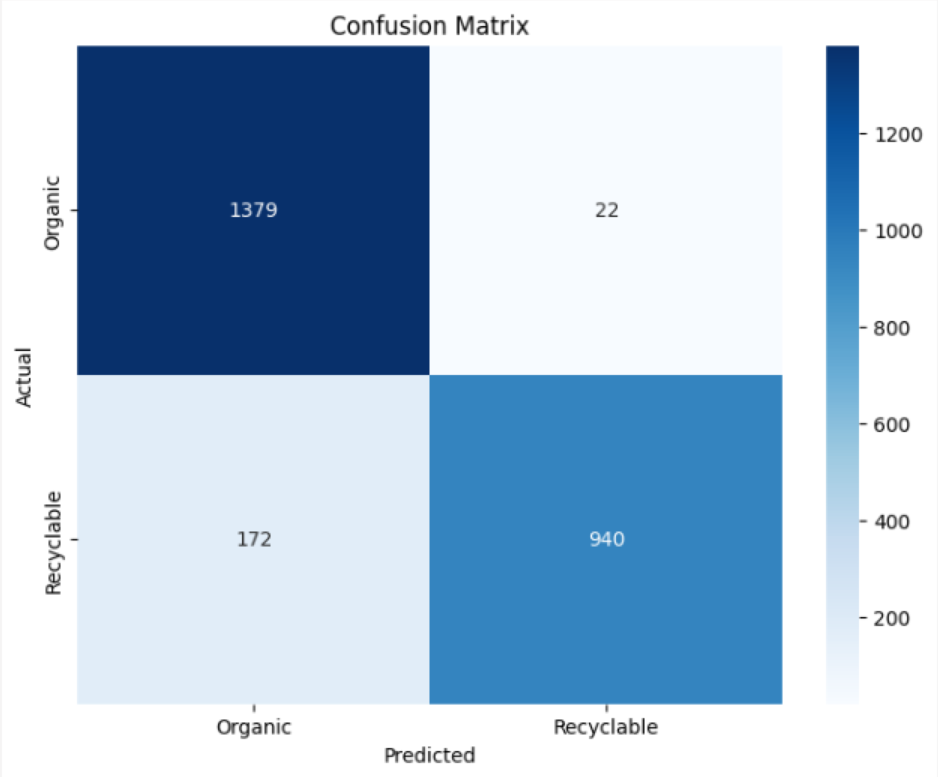
YOLOv12 틀린 추론 사진(20개 sampling)

YOLOv12 혼동 행렬 (Confusion Matrix)

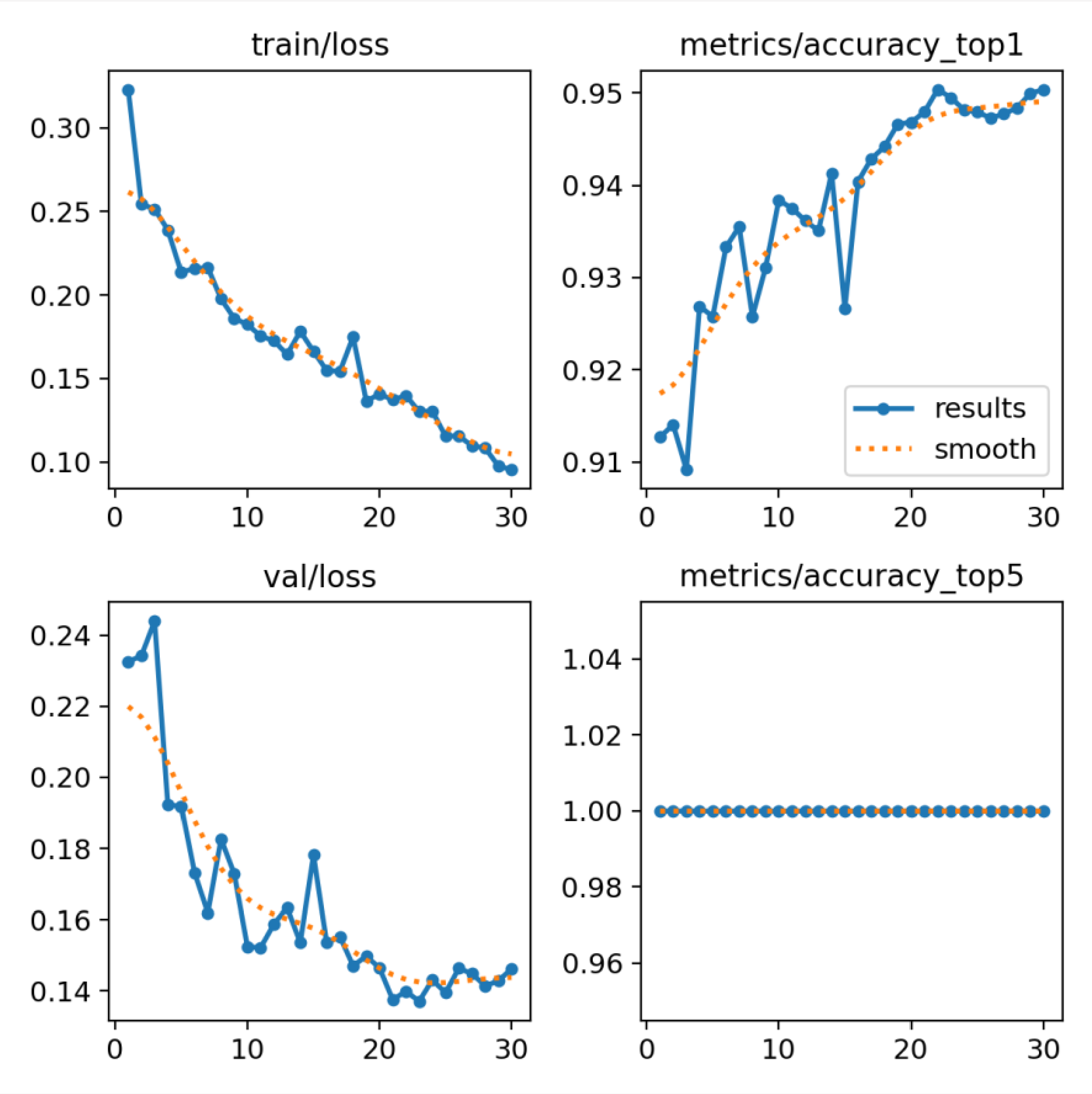








YOLOv12 모델 학습 시 자동으로 저장되는 지표



epoch	time	train/loss	metrics/accuracy_to	metrics/accuracy_to	val/loss
1	115.26	0.3229	0.9127	1	0.2326
2	216.5	0.2544	0.91403	1	0.23425
3	317.565	0.25148	0.90915	1	0.24406
4	418.969	0.23899	0.92688	1	0.19236
5	531.214	0.21389	0.92577	1	0.19174
6	633.876	0.21581	0.9333	1	0.17306
7	738.167	0.21644	0.93552	1	0.16189
8	842.344	0.19771	0.92577	1	0.18263
9	949.425	0.18597	0.93109	1	0.17284
10	1051.38	0.1824	0.9384	1	0.15234
11	1153.42	0.17533	0.93751	1	0.15216
12	1256.21	0.17264	0.93618	1	0.15874
13	1360.21	0.16439	0.93508	1	0.1634
14	1465.13	0.1783	0.94128	1	0.15354
15	1568.47	0.16623	0.92666	1	0.17809
16	1671.23	0.15485	0.94039	1	0.15351
17	1772.71	0.15424	0.94283	1	0.15527
18	1876.08	0.175	0.94416	1	0.14701
19	1984.16	0.13646	0.9466	1	0.14978
20	2086.26	0.14074	0.94682	1	0.14638

모델별 데이터 전처리 VGG16

Resize 224 * 224

preprocess_input VGG16 전용 전처리 함수 사용

VGG16

01 Learning Rate 0.0001

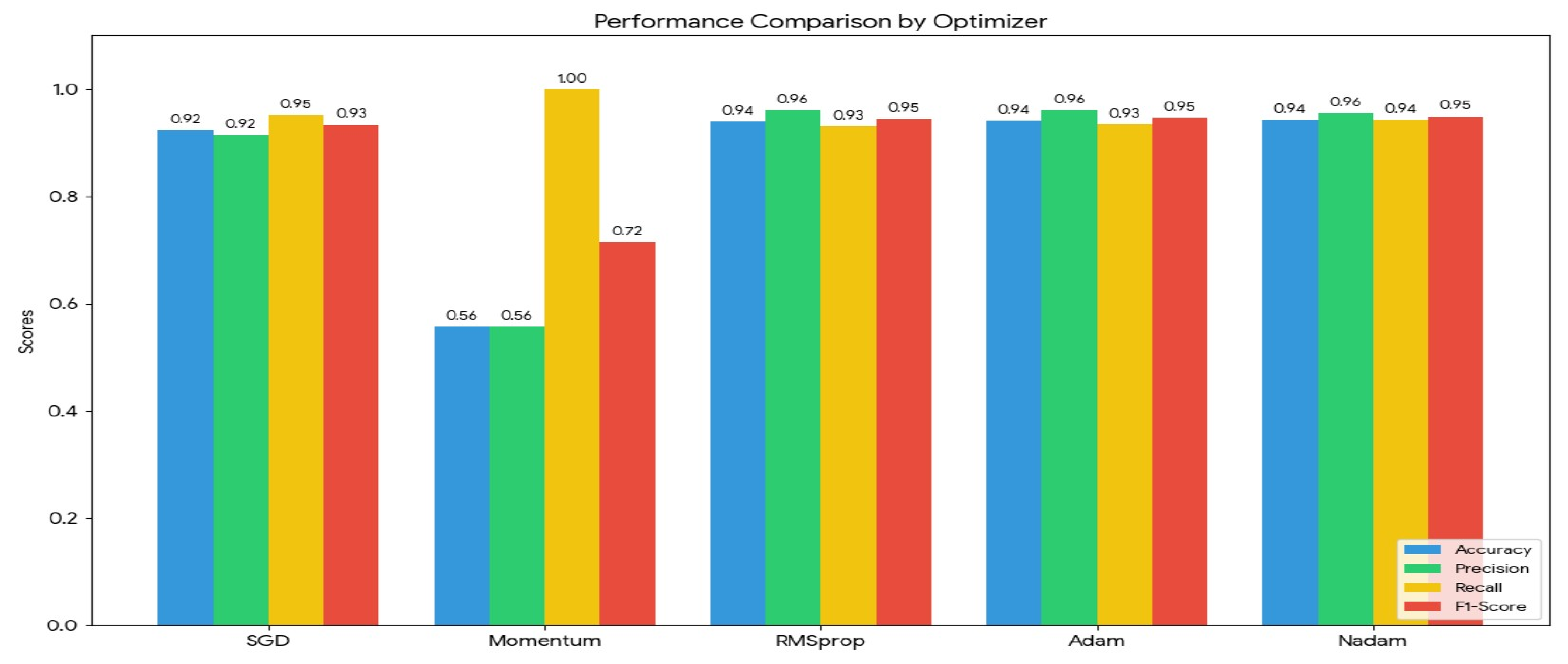
02 Optimizer (RMSprop, Adam, Nadam, SGD, Momentum

03 Epoch 20 고정

VGG16 지표 분석

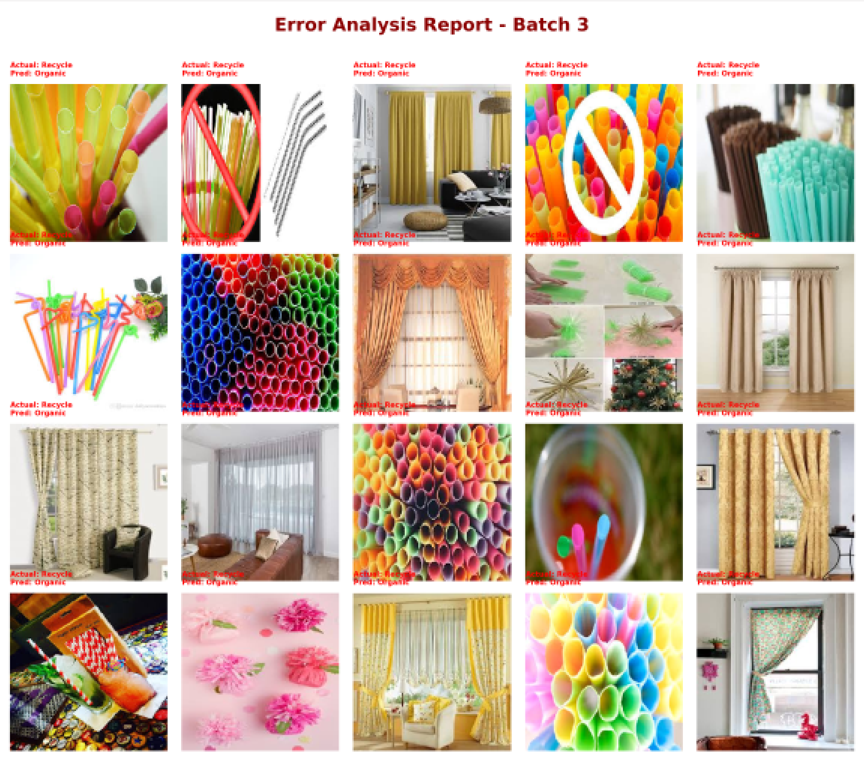
	RMSprop	Adam	Nadam	SGD	Momentum
Learning Rate	0.0001	0.0001	0.0001	0.01	0.01
Accuracy (정확도)	0.9400	0.9419	0.9437	0.9242	0.5568
Precision (정밀도)	0.9606	0.9611	0.9556	0.9151	0.5568
Recall (재현율)	0.9304	0.9335	0.9427	0.9522	1.0000
F1-score	0.9452	0.9471	0.9491	0.9333	0.7153

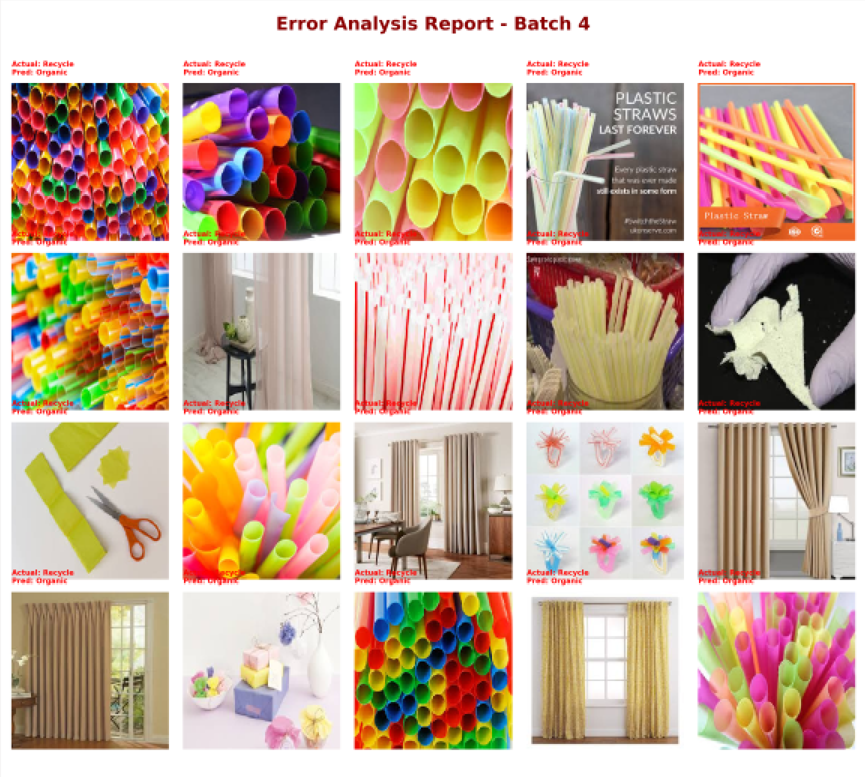
VGG16 모델 평가 (Model Evaluation Results)



VGG16

VGG16(SDG) 틀린 추론 사진





폐기물 판별 시연

행정 처리 비용 예상 절감 효과

예상 수치

500억원

서울시 500만 가구 기준으로 오분류율 2%를 잡았을때,
인력 500명을 투입한다는 가정하에
행정 처리비용 연간 500억 원을 절약

음식물 처리 시설 비용 예상 절감 효과

예상 수치

11억원

서울시 500만 가구 기준 음식물 배출량을 1가구당 0.3kg,
오분류율 2%로 잡았을 때,
일 30톤 처리 = 300만원, 연간 약 11억원 절약

CO2 예상 감축 효과

예상 수치

5 천톤

연간 오분류 음식물 감축량 약 5천 톤, 1톤당 1톤의 CO2를
감축한다고 가정, 약 5천 톤의 CO2 감축
(약 중형차 3천대 분량의 CO2 감축)

사용한 툴 & python 라이브러리

모델링, 분석

자료 취합

소통

도움

HW

T4 & cuda
(Colab)

MPS
(Apple silicon)

기본 데이터 처리	numpy, pandas, os, math
이미지 처리	torch, torchvision, opencv-python
시각화	matplotlib, seaborn
모델링	tensorflow, keras, ultralytics(YOLO)
기타	time, zipfile, google.colab.drive, urllib, shutil
prototype	(streamlit)

Timeline

12.29

16:00 ~ 17:00

데이터 탐색

각자 학습할 모델 선정

최종 성능평가시
고정할 조건 정하기

프로젝트 당위성 아이디어 도출

12.30

9:00 ~ 10:00

각자 돌려본 모델
성능평가 형식 논의
: 평가 항목, 혼동행렬 컬러 등

10:00 ~ 12:30

CNN 모델 조건별 실험 및 자료 정리(●●)

VGG16 모델 조건별 실험 및 시연모델 개발(●●)

RegNet 모델 조건별 실험 및
스토리텔링 근거자료 검색(지호)

YOLOv12 실험(●●)

14:00 ~ 17:00

실험 진행 및 결과 정리
자료 1차 취합

12.31

9:00 ~ 12:30

자료 디벨롭 및 시연 프로그램 QA,
발표 스크립트 정리

14:00 ~ 15:00

발표준비

RecyNET COMPANY

Happy New Year!!!