

YOLO 구현 과제

팀명 : GELU JOA

고현서 노현지 손단하 신현식 이현





목차

- 01 프로젝트 개요
- 02 YOLO v2
- 03 YOLO v3
- 04 탐지기 성능 분석(하이퍼 파라미터 표 , 추론 그림)
- 05 분류기 성능 분석
- 06 한계점 및 향후 연구 과제

프로젝트 개요



✓ 프로젝트 기획 목표

- 아동 발달과 심리 분석의 중요성
- 기존 심리 평가 방식의 한계
- AI 기반 자동 분석의 필요성

✓ 데이터 소개

소개

- 다양한 연령대(7~13)와 성별의 아동 7,000명으로부터 수집한 4개 HTP 분류(집, 나무, 여자사람, 남자사람) 그림 56,000건과 그림 내 주요 객체에 대한 바운딩박스 라벨링을 시행한 라벨링 데이터 56,000건으로 구성
- 심리 해석이 포함된 의료 데이터가 아닌 객체 인식을 위한 그림 데이터로서 법적 문제가 없는 데이터를 확보

YOLOv2 객체 탐지 모델_파이프라인

데이터 준비 및 전처리

데이터 준비 → JSON 라벨 변환 class name 추출 → 클래스 인덱싱 매핑 생성 → Anchor 추출 (K-means clustering) → Target 생성 → YOLOv2 훈련 데이터셋 생성



YOLOv2 모델 훈련

Darknet-19 백본 기반 YOLOv 모델 정의 → YOLOv2 손실 함수 정의 (좌표 오차, objectness BCE, class CE 결합) → YOLOv2 로 탐지, ResNet-18로 분류 후 베스트 모델 저장

YOLOv2 객체 탐지 모델_하이퍼파라미터

구분	값	변경 이유
lr (learning_rate)	1e-4.	논문에선 1e-2부터 시작하지만 Adam과 같은 스케줄러와 함께 쓰기때문에 1e-4로 결정, 더 작아지면 수렴 속도가 매우 느려짐
Epoch	50	Anchor 재학습, Adam optimizer 사용으로 빠른 수렴 가능 → 적은 epoch로도 성능 확보
lambda_coord	5.0 → 6.0	모델이 (x, y, w, h) 예측 오차를 더 크게 페널티 받음 -> 객체 위치와 크기를 더 잘 맞추려는 방향으로 학습됨
Optimizer	Adam	custom dataset (작고 불균형할 가능성 ↑) → Adam이 빠른 수렴 + 적은 데이터에서 효과적
conf_th	0.15 ~ 0.25	작은 객체/희귀 객체 감지 강화를 위해 낮춤
iou_th	0.45	기존 0.5보다 살짝 낮춤 → 더 많은 박스 유지, 재현율 향상

최종 가중치 : train=1.6164 val=10.9846

YOLOv3 객체 탐지 모델_파이프라인

데이터 준비 및 전처리

데이터 준비 → JSON 파일 파싱 → **K-MEANS** 클러스터링으로 앵커박스 생성
→ YOLOv3 훈련 데이터셋 생성 (데이터 증강, 레터박스, 타겟변환)



YOLOv3 모델 훈련

Darknet53_Slim 백본 기반 YOLOv3 모델 정의 → YOLOv3 손실 함수 정의
→ **K-Fold** 교차 검증으로 모델 훈련 및 최적 가중치 저장

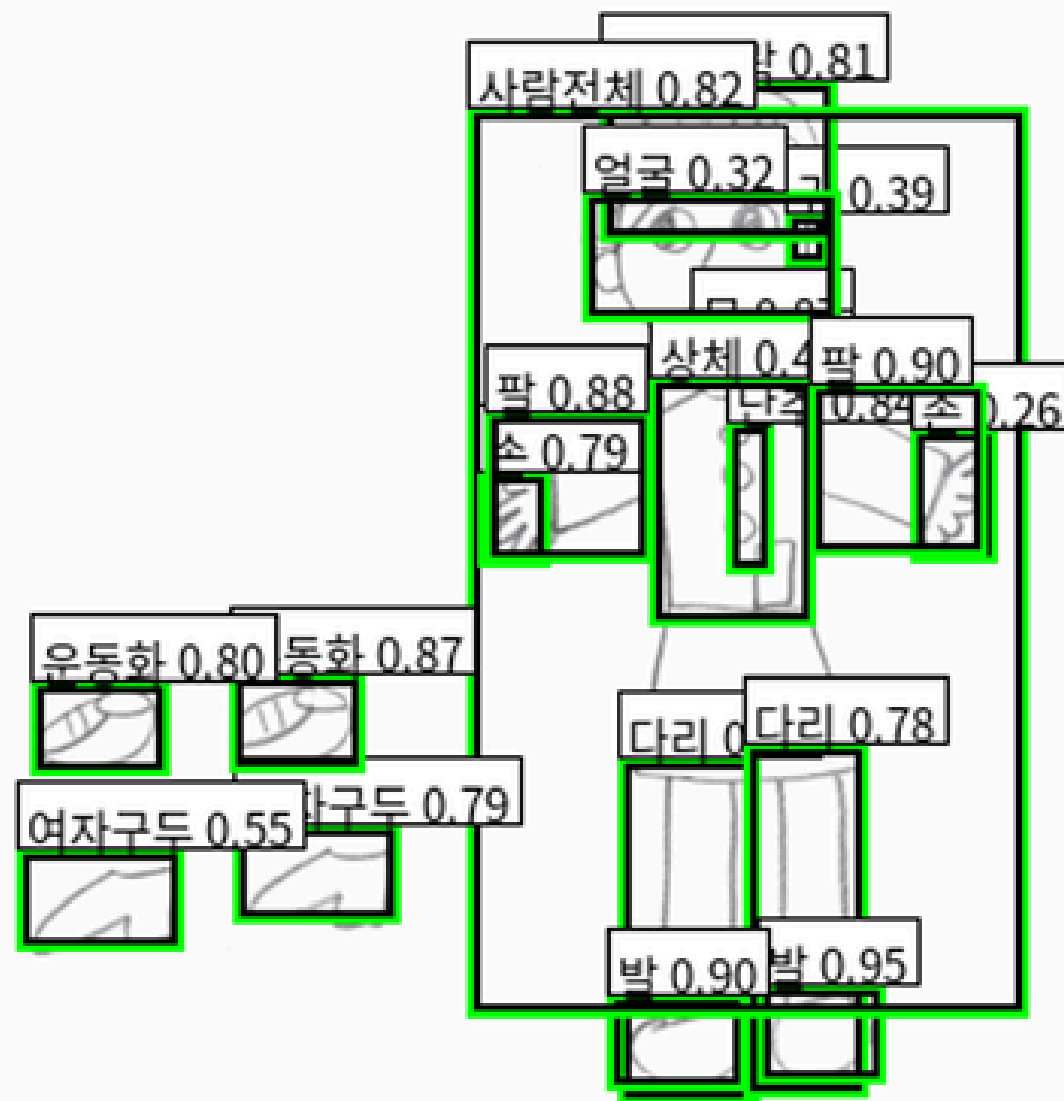
YOLOv3 객체 탐지 모델_하이퍼파라미터

구분	범위	변경 이유
lr (learning_rate)	1e-6 ~ 1e-3	val 손실이 정체되어 더 미세한 최적점을 탐색하기 위해 낮춤.
weight_decay	1e-4 ~ 1e-2	오버피팅 발생 시 정규화 강화를 위해 1e-2로 시도했으나, train 손실 정체로 인해 다시 복원.
lamdba_noobj	0.05 ~ 0.3	탐지 모델이 배경에 과도하게 집중하는 것을 막고, 객체 탐지에 더 집중하도록 가중치를 낮춤.
patience	5 ~ 20	손실이 5번 연속 개선되지 않으면 학습을 멈춤.
conf_th	0.35 ~ 0.6	탐지된 박스가 너무 많고, 낮은 신뢰도의 오탐지가 많아져 임계값을 높여 필터링을 강화함.
iou_th	0.45 ~ 0.6	하나의 객체에 여러 개의 겹치는 박스가 탐지되는 현상 해결을 위해 NMS 를 더 엄격하게 적용함.

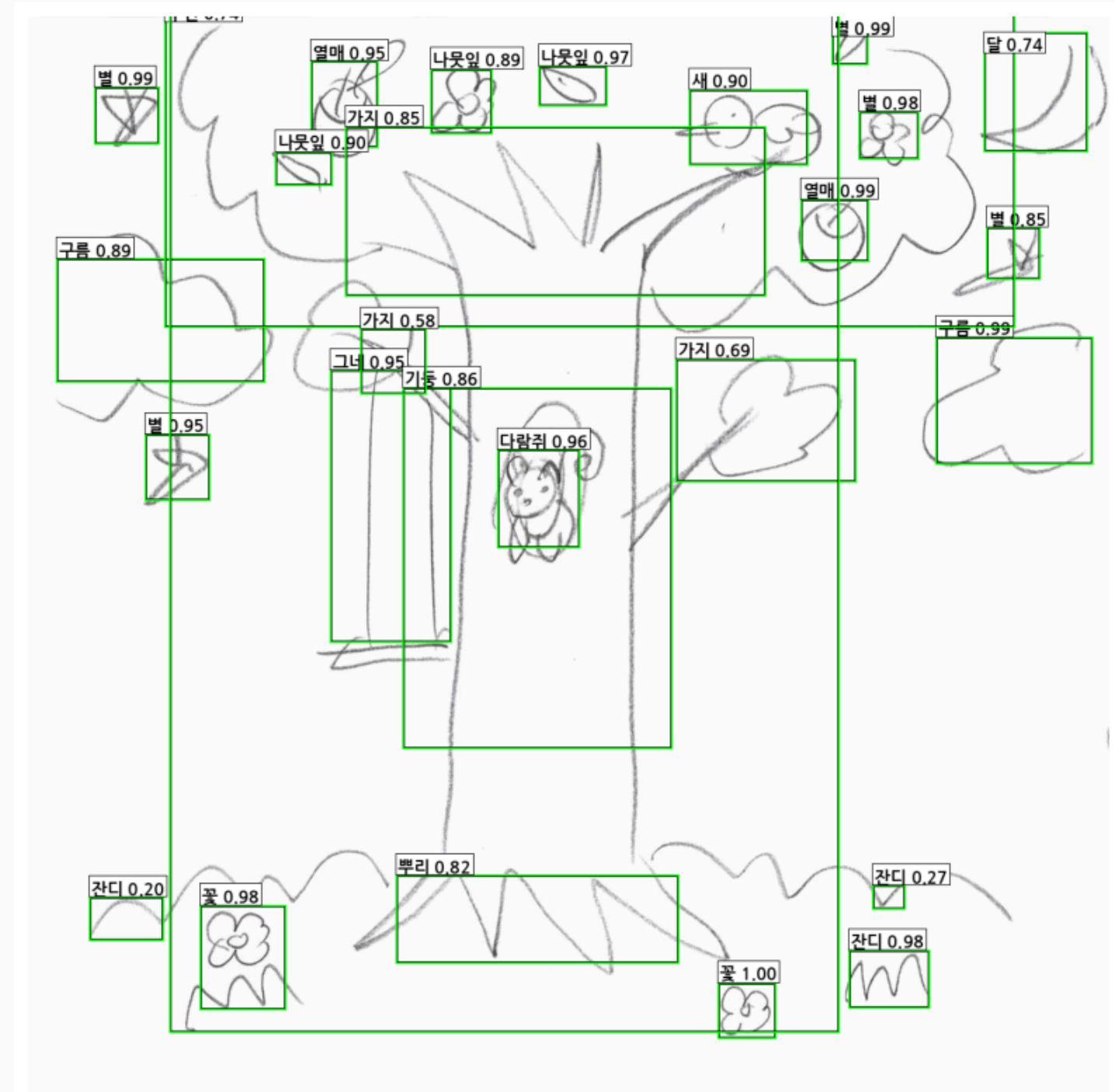
최종 가중치 : train=36.406 val=42.943

객체 탐지 모델 성능 분석

YOLOv2

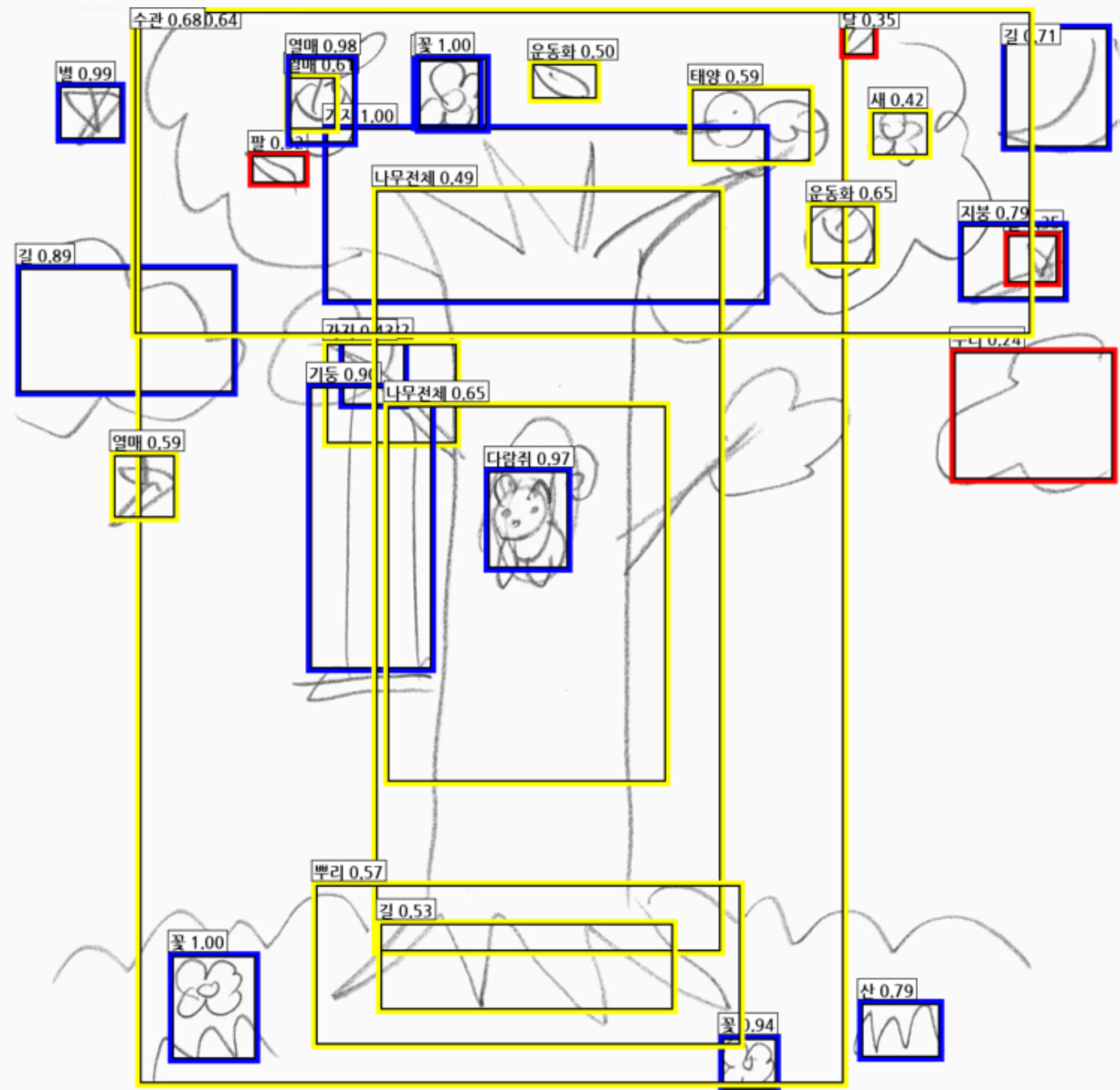


YOLOv3

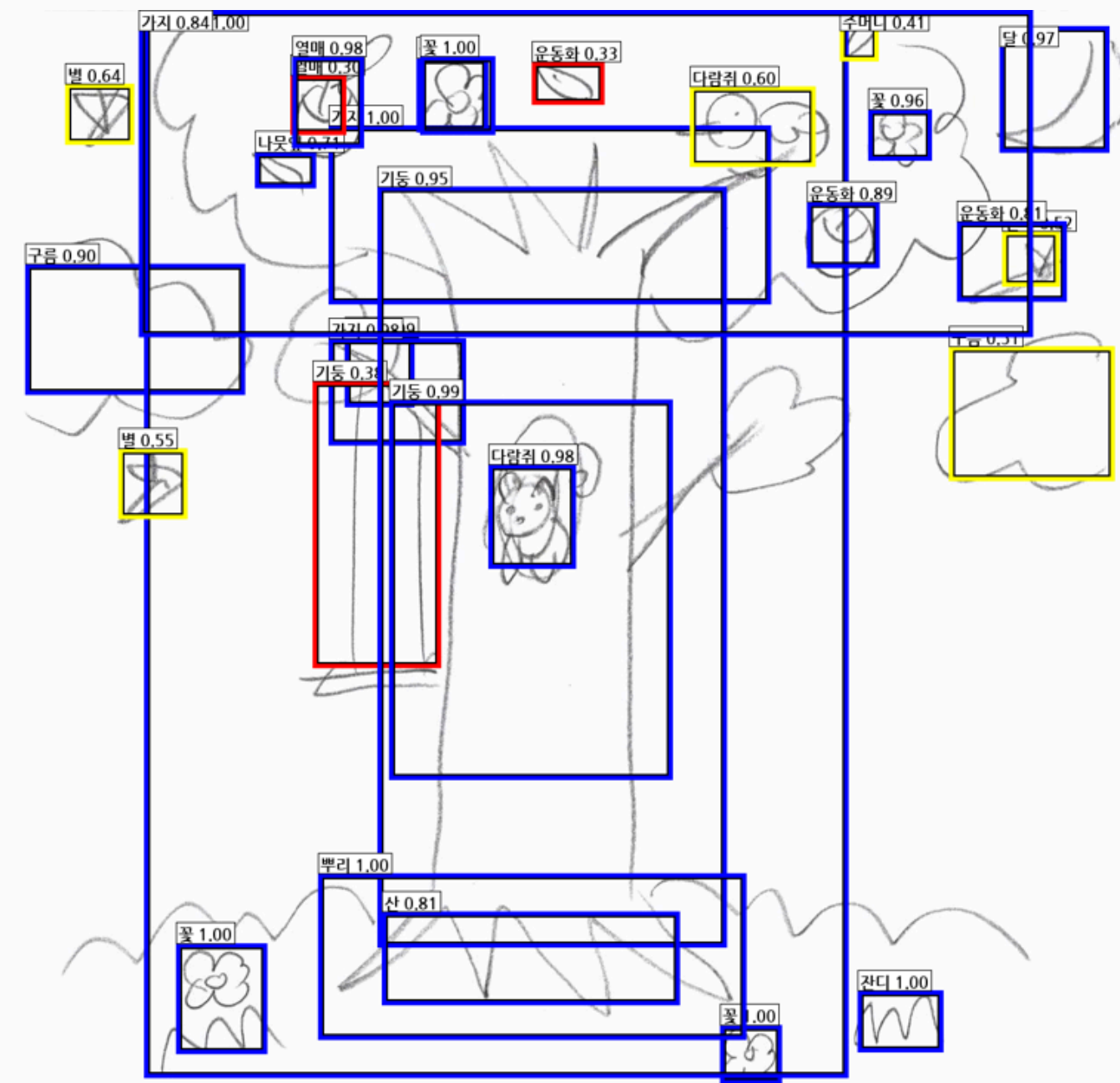


객체 분류 모델 성능 분석

ResNET-18



Efficientnet-b0



객체 분류 모델 성능 분석 그래프

ResNET-18

	f1-score	support
가치	1.00	3.00
구름	0.00	2.00
그네	0.00	1.00
기둥	0.00	2.00
꽃	0.86	3.00
나무전체	0.50	1.00
나뭇잎	0.00	4.00
다람쥐	1.00	1.00
달	0.00	1.00
별	0.29	6.00
뿌리	0.50	2.00
새	0.00	1.00
수관	1.00	1.00
열매	0.40	2.00
잔디	0.00	1.00
accuracy	0.39	0.39
macro avg	0.12	31.00
weighted avg	0.37	31.00

Efficientnet-b0

	f1-score	support
가치	0.86	3.00
구름	1.00	2.00
그네	0.00	1.00
기둥	0.80	2.00
꽃	0.75	3.00
나무전체	1.00	1.00
나뭇잎	0.40	4.00
다람쥐	0.67	1.00
달	1.00	1.00
별	0.50	6.00
뿌리	0.67	2.00
새	0.00	1.00
수관	0.00	1.00
열매	0.50	2.00
잔디	0.67	1.00
accuracy	0.61	0.61
macro avg	0.19	31.00
weighted avg	0.60	31.00

실전 테스트 시연

한계점 및 향후 연구 과제

탐지기

한계점

- 작은 객체 탐지 한계
- 유사 객체 혼동
- False Positive

향후 연구 과제

- 고해상도 입력과 멀티스케일 피처 강화 적용
- Fine-grained 분류 모듈 추가로 세밀한 차이 학습
- 처리 기법 개선 - Soft-NMA

분류기

한계점

- 크롭 품질 의존성
- 다수 클래스 분류 어려움
- Generalization 약함

향후 연구 과제

- 엔드 투 엔드 구조
- 데이터 증강
- 대규모 데이터 확보

질문과 답변



프로젝트에 대해 궁금하신 점이 있으신가요?
자유롭게 질문해 주세요.

감사합니다!