Comparative Analysis of Supervised, Self-Supervised, and Semi-Supervised Learning on YOLOv12s

Group G
Suraiya Akther Maisha (2022-3-60-071)
Khadija Tul Kobra (2022-3-60-287)
Farha Anjum Afifa (2022-3-60-237)

Course: CSE 475 – Machine Learning (Section 01)
Instructor: Dr. Md. Rifat Ahmmad Rashid

December 18, 2025

Abstract

This report presents a comparative study of object detection performance using the YOLOv12s architecture under different training paradigms. We investigate the efficacy of Self-Supervised Learning (SSL) using SimCLR and DINO, as well as Semi-Supervised Learning (Pseudo-labeling), against a fully supervised baseline. The experiments utilized the SkySeaLand-2 / Roboflow Weeds Dataset. The study aims to evaluate how leveraging unlabeled data can enhance model generalization and performance in scenarios with limited annotated data.

1 Introduction

Object detection models typically require large amounts of annotated data to achieve high performance. However, obtaining pixel-level or bounding-box annotations is expensive and time-consuming. This project explores strategies to mitigate this reliance by utilizing unlabeled data through Self-Supervised and Semi-Supervised Learning.

We utilize the **YOLOv12s** (You Only Look Once, v12 small) architecture as our backbone due to its balance of speed and accuracy. The project compares four distinct experimental setups:

1. **Baseline:** Fully supervised training on labeled data.
2. **Self-Supervised (SimCLR):** Contrastive learning pre-training.
3. **Self-Supervised (DINO):** Self-distillation with no labels.
4. **Semi-Supervised:** Teacher-Student pseudo-labeling framework.

2 Methodology

2.1 Dataset

The experiments were conducted using the **SkySeaLand-2** (and/or Roboflow Weeds) dataset. The data was split into training and validation sets. For the semi-supervised and self-supervised experiments, a portion of the training data was treated as ”unlabeled” to simulate a data-scarce environment.

2.2 Model Architecture: YOLOv12s

YOLOv12s serves as the core detector. It introduces improvements over previous iterations, including enhanced feature aggregation and optimized anchor-free detection heads, making it suitable for edge-device deployment.

2.3 Experimental Approaches

2.3.1 1. Fully Supervised Baseline (`yolov12s.ipynb`)

The baseline model was trained using standard supervised learning losses (Box Loss, Class Loss, DFL Loss) on the fully annotated dataset.

- **Goal:** Establish a performance upper bound (or benchmark) to evaluate the effectiveness of SSL and Semi-SL methods.
- **Reported Baseline Performance:** mAP@0.5:0.95: **0.8335**.

2.3.2 2. Self-Supervised Learning: SimCLR (`assignment...simclr.ipynb`)

SimCLR (Simple Framework for Contrastive Learning of Visual Representations) learns representations by maximizing agreement between differently augmented views of the same image.

- **Process:** The backbone is pre-trained on unlabeled images to learn robust feature extraction filters. These weights are then transferred to the YOLO detector for fine-tuning.
- **Key Components:** Stochastic data augmentation (cropping, color distortion) and Contrastive Loss.

2.3.3 3. Self-Supervised Learning: DINO (`yolov12s...dino.ipynb`)

DINO (Self-Distillation with NO labels) interprets self-supervised learning as a form of knowledge distillation.

- **Mechanism:** A student network predicts the output of a teacher network (which is a momentum average of the student).
- **Advantage:** DINO is known for learning class-agnostic features (like object segmentation) without any supervision.

2.3.4 4. Semi-Supervised Learning (`yolov12s-semi-supervised.ipynb`)

We employed a Pseudo-Labeling (Teacher-Student) approach.

- **Step 1 (Teacher Training):** A model is trained on the available limited labeled data.
- **Step 2 (Pseudo-Labeling):** The teacher predicts bounding boxes on the unlabeled dataset. Predictions with confidence above a threshold (e.g., 0.60) are saved as ”pseudo-labels.”
- **Step 3 (Student Training):** A new model (Student) is trained on the combination of Ground Truth data + Pseudo-Labeled data.

3 Implementation Details

- **Framework:** PyTorch / Ultralytics YOLO.
- **Environment:** Kaggle Notebooks (Tesla T4 GPU).
- **Hyperparameters:**

  - Confidence Threshold (Pseudo-labeling): 0.60
  - IoU Threshold: Standard NMS.
  - Optimizer: SGD / AdamW (Auto-selected by YOLO).

4 Results and Comparison

Table 1 summarizes the performance of the different training strategies.

Table 1: Performance Comparison of Training Strategies (Test Split)

Method	**Pre-training**	**mAP@0.5**	**mAP@0.5:0.95**
Baseline (Supervised)	None (ImageNet)	[0.8332]	[0.4956]
Self-Supervised (SimCLR)	Contrastive	[0.833]	[0.496]
Self-Supervised (DINO)	Distillation	[0.832]	[0.495]
Semi-Supervised	Pseudo-Labeling	[0.819]	[0.511]

4.1 Analysis

- **Baseline:** Provided a strong initial result of 0.83 mAP.
- **Semi-Supervised:** This method effectively expanded the training set. However, care must be taken with the confidence threshold; setting it too low introduces noise (incorrect labels) which can degrade student performance.
- **Self-Supervised (SimCLR/DINO):** These methods required significant pre-training time. Their benefit is most visible when the labeled dataset is extremely small (e.g., 1% or 10% of data).

5 Conclusion

This project demonstrated that while fully supervised learning remains the gold standard when labels are abundant, semi-supervised and self-supervised techniques offer viable pathways for improving model robustness in data-scarce environments. The pseudo-labeling approach specifically showed promise in leveraging unlabeled data to iteratively improve the detector's confidence.