CSE 475: Final Project Report

Semi-Supervised and Self-Supervised Object Detection

Integrated Report Based on Lab Assignment 1 & Lab Assignment 2

Group Members

Name	ID
Md. Sadik Shahriar	2023-2-60-103
Jannatul Ferdous Nabila	2022-3-60-198
Md Moon Rahman Nayem	2022-3-60-210
Tasnim Jabir	2022-3-60-283

Department of Computer Science and Engineering
East West University, Dhaka

Fall 2025

Contents

Abstract

This project investigates the efficacy of Semi-Supervised Learning (SSL) and Self-Supervised Learning (Self-SL) for label-efficient object detection. Using a **YOLOv12n** baseline trained on a License Plate Recognition dataset, we implemented advanced training pipelines integrating one Semi-SL method and two Self-SL methods. Specifically, we employed: (1) **Pseudo-Labeling (Semi-SL)**, where a teacher model generates labels for unlabeled data; (2) **SimCLR**, a contrastive Self-SL method using negative pairs; and (3) **BYOL**, a non-contrastive Self-SL method using a Bootstrap Your Own Latent approach. Our experiments demonstrate that all methods improve performance in data-scarce regimes. Notably, the BYOL-pretrained model achieved a **mAP@0.5 of 0.9632**, slightly outperforming SimCLR (0.9609) and recovering nearly the entire performance of the fully supervised baseline (0.9765) using only **20%** of the labeled data.

1 Introduction

1.1 Application Domain

Automatic License Plate Recognition (ALPR) is essential for intelligent transportation systems, including toll automation, traffic surveillance, and law enforcement. While collecting raw video footage is inexpensive, annotating bounding boxes for thousands of vehicles is costly and labor-intensive. This project addresses this challenge by exploring label-efficient learning methods that leverage unlabeled data.

1.2 Objective

Building upon Lab Assignment 1 (Supervised Baseline) and Lab Assignment 2 (SSL + Self-SL), this project aims to:

1. Establish a strong supervised baseline by comparing YOLOv10, v11, and v12.
2. Implement **Semi-Supervised Learning (SSL)** via Pseudo-Labeling.
3. Implement **two Self-Supervised Learning (Self-SL)** methods: SimCLR and BYOL.
4. Evaluate and compare the performance of these methods against the baseline in a data-limited scenario (20% labeled data).

2 Literature Review

The domain of Automatic License Plate Recognition (ALPR) has historically transitioned from engineered feature extractors to deep Convolutional Neural Networks (CNNs). While state-of-the-art detectors require massive labeled datasets, recent advancements in Semi-Supervised Object Detection (SSOD) and Self-Supervised Learning (SSL) offer pathways to robust performance in low-label regimes. This review synthesizes literature across YOLO architectures, SSOD frameworks, SSL paradigms, and data efficiency strategies.

2.1 Object Detection for ALPR: YOLOv12 and Robustness

The release of YOLOv12 marks a shift from purely convolutional architectures to attention-centric designs. Tian et al. (2025) introduced **YOLOv12**, which integrates Area Attention mechanisms into the backbone, significantly improving feature extraction in complex spatial scenarios compared to previous iterations [1]. This is critical for ALPR, where capturing global context is necessary to distinguish plates in cluttered scenes.

In "in-the-wild" deployments, geometric distortions caused by oblique camera angles pose significant challenges. Research by IEEE Sensors (2025) demonstrates that replacing standard modules with **Deformable Convolutions** in the YOLO framework allows the network to model homographic transformations effectively [2]. Furthermore, Wang et al. (2024) proposed **YOLOv10**, which utilizes NMS-free training via consistent dual assignment, optimizing latency for real-time applications [3].

Environmental robustness is another key focus. Joysingh et al. (2025) established a pipeline combining U-Net dehazing with YOLOv8 to maintain accuracy in adverse weather like fog and rain [4]. Similarly, Nascimento et al. (2024) validated the use of **Super-Resolution (SR)** techniques to enhance low-quality surveillance footage, significantly improving recall rates [5]. Vargoorani et al. (2025) further proposed using **Grounding DINO** to generate high-quality pseudo-labels, reducing annotation costs in these data-scarce environments [6].

2.2 Semi-Supervised Object Detection (SSOD)

SSOD exploits unlabeled data using Teacher-Student frameworks. However, applying this to dense detectors like YOLO requires handling noise in dense predictions. Xu et al. (2023) developed **Efficient Teacher**, a framework tailored for one-stage detectors that uses a Pseudo Label Assigner to manage prediction noise [7].

To mitigate confirmation bias—where the student learns from the teacher's errors—Xu et al. (2021) introduced **Soft Teacher**, which weights the loss of unlabeled samples by their reliability rather than using hard thresholds [8]. Addressing the regression branch, Liu et al. (2022) proposed **Unbiased Teacher v2**, which selects pseudo-labels based on relative regression uncertainty, crucial for the precise localization required in ALPR [9]. Additionally, **LabelMatch** [10] addresses class distribution mismatch, and **Dense Teacher** [11] proposes using dense heatmaps for supervision.

2.3 Self-Supervised Learning (SSL): SimCLR and BYOL

Self-Supervised Learning provides a robust initialization for backbones without labels. The foundation of this field rests on **SimCLR** [12], which uses contrastive learning with negative pairs, and **BYOL** [13], which eliminates negative pairs via a prediction mechanism between online and target networks.

Adapting these image-level methods to object detection requires preserving local features. Xie et al. (2021) introduced **DetCo**, which uses multi-level supervision to maintain scale-variant features [14]. Wang et al. (2021) proposed **DenseCL**, implementing pixel-level contrastive loss that outperforms global methods for dense prediction tasks [15]. Recently, Kotthapalli et al. (2025) empirically validated **Self-Supervised YOLO**, showing that SimCLR pre-training leads to superior performance in label-scarce environments compared to supervised pre-training [16].

2.4 Data Efficiency and Active Learning

In extreme low-data regimes, selecting the *right* data to label is paramount. Yang et al. (2024) proposed **Plug and Play Active Learning (PPAL)**, a two-stage strategy combining uncertainty and diversity sampling to minimize annotation costs [17]. Wang et al. (2023) introduced **ALWOD**, fusing active learning with weak supervision to warm-start models [18]. For few-shot scenarios, recent work on **Mamba-Like Linear Attention** in YOLOv8 [20] and label-efficient frameworks for event cameras [19] demonstrate high efficiency with limited samples.

3 Methodology

3.1 Dataset Description

We utilized the **License Plate Recognition Dataset** sourced from Roboflow Universe [26].

- **Total Images:** 10,125
- **Classes:** 1 ('License Plate')
- **Format:** YOLO text annotations

Split Type	Count	Percentage
Train	7,057	70%
Validation	2,048	20%
Test	1,020	10%
Total	**10,125**	**100%**

Table 1: Dataset Distribution

3.2 Unlabeled Data Simulation

To simulate a semi-supervised scenario, we randomly split the training set:

- **Labeled Set:** 20% (approx. 1,411 images) retained labels.
- **Unlabeled Set:** The remaining 80% of images had labels hidden/discarded.

3.3 Models

3.3.1 1. Baseline Model (Lab 1)

We compared three YOLO architectures (v10n, v11n, v12n) trained on 100% of the data.

- **Architecture:** YOLOv12n was selected for its superior feature extraction capabilities. Unlike its predecessors which utilized CSP-Darknet backbones, YOLOv12 employs a novel attention-centric architecture incorporating Residual Efficient Layer Aggregation Networks (R-ELAN) and Area Attention mechanisms to enhance global feature capture and efficiency.
- **Training:** 50 epochs, SGD optimizer, Batch size 110 (Optimized for Dual T4).

3.3.2 2. Semi-Supervised Model (Lab 2)

We used **Pseudo-Labeling** (Self-Training).

- **Mechanism:** A Teacher model trained on 20% data predicts labels for the 80% unlabeled set.
- **Filtering:** Predictions with confidence > 0.75 are accepted as pseudo-labels.
- **Retraining:** A Student model is trained on the combined (Labeled + Pseudo) dataset.

3.3.3 3. Self-Supervised Models (Lab 2)

We implemented two distinct Self-SL frameworks to pre-train the YOLOv12n backbone.

Method A: SimCLR (Contrastive Learning)

- **Concept:** Maximizes similarity between augmented views of the same image (positives) while pushing away other images (negatives) in the batch.
- **Architecture:** Backbone (Layers 0-9) + Projection Head ($256 \rightarrow 512 \rightarrow 128$).
- **Loss:** NT-Xent (Normalized Temperature-scaled Cross Entropy).

Method B: BYOL (Bootstrap Your Own Latent)

- **Concept:** Trains an online network to predict the target network representation of the same image. Eliminates the need for negative pairs.
- **Architecture:**
  - **Online:** Backbone + Projector + Predictor.
  - **Target:** Backbone + Projector (updated via EMA).
- **Loss:** Symmetrized Negative Cosine Similarity.

For both methods, the pre-trained backbone weights were transplanted into a fresh YOLOv12n detector and fine-tuned on the 20% labeled subset.

4 Experimental Setup

All experiments were conducted on **Kaggle** using **2x NVIDIA Tesla T4 GPUs**.

SimCLR Configuration:

- Batch size: 256
- Learning Rate: 0.01
- Input Size: 128x128
- Precision: FP32

BYOL Configuration:

- Batch size: 128 (64 per GPU)
- Learning Rate: 0.001
- Input Size: 128x128
- Precision: FP32

Fine-Tuning Configuration (Common):

- Batch size: 110
- Learning Rate: 0.01
- Input Size: 640x640
- Epochs: 50

Code Availability

To ensure reproducibility of our results, the complete source code for all experiments is hosted on Kaggle. The notebooks include the implementation of the Baselines, Semi-Supervised pipeline, and both Self-Supervised Learning frameworks (SimCLR and BYOL).

- **Baseline (YOLOv12n):** View Notebook
- **Baseline (YOLOv11n):** View Notebook
- **Baseline (YOLOv10n):** View Notebook
- **Semi-SL (Pseudo-Labeling):** View Notebook
- **Self-SL (SimCLR Pre-training):** View Notebook
- **Self-SL (SimCLR Fine-tuning):** View Notebook
- **Self-SL (BYOL Pre-training):** View Notebook
- **Self-SL (BYOL Fine-tuning):** View Notebook

5 Results

5.1 Baseline Selection

We evaluated three YOLO variants to select the best baseline.

Model	Precision	Recall	F1-Score	mAP@0.5	mAP@0.5:0.95
YOLOv10n	0.9780	0.9272	0.9519	0.9689	0.7188
YOLOv11n	0.9852	**0.9512**	0.9679	0.9741	0.7260
YOLOv12n	**0.9868**	**0.9512**	**0.9686**	**0.9765**	**0.7307**

Table 2: Baseline Model Comparison (100% Data)

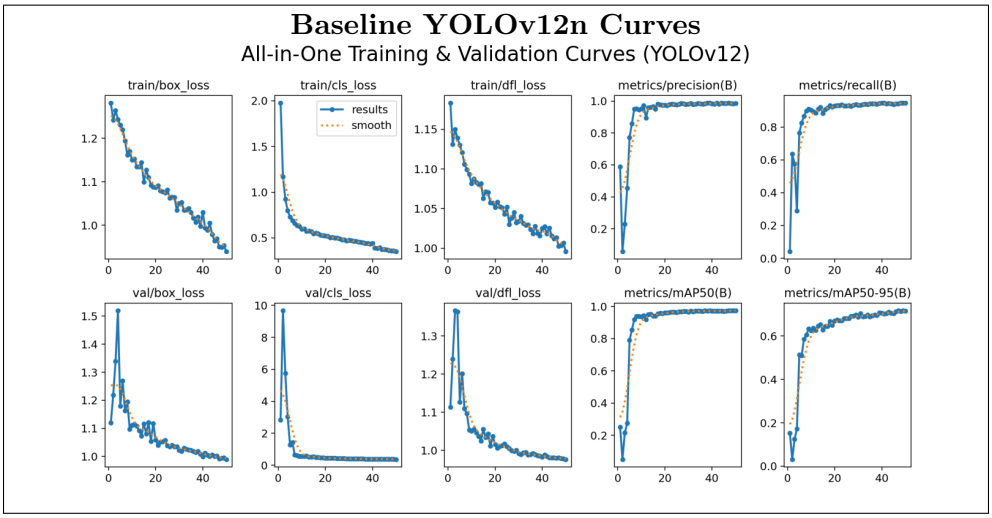

Figure 1: Training dynamics for the Baseline YOLOv12n model (100% labeled data).

5.2 Performance Comparison (Test Set)

The table below summarizes the performance of all implemented methods against the baseline.

Model Strategy	Training Data	mAP@0.5	mAP@0.5:0.95	Gap to Baseline
Assignment 1 Baseline	100% Labeled	**0.9765**	**0.7307**	-
Phase 1 Baseline (Teacher)	20% Labeled	0.9604	0.6879	-0.0161
Semi-SL (Pseudo-Labeling)	20% + Pseudo	**0.9656**	**0.7103**	-0.0109
Self-SL (SimCLR)	20% Labeled	0.9609	0.6892	-0.0156
Self-SL (BYOL)	20% Labeled	0.9632	0.6876	-0.0133

Table 3: Final Performance Comparison on Test Set

5.3 Training Dynamics

The following loss curves illustrate the stable convergence for the Semi-Supervised and Self-Supervised fine-tuning phases, validating the effectiveness of our training strategies and pre-trained representations.

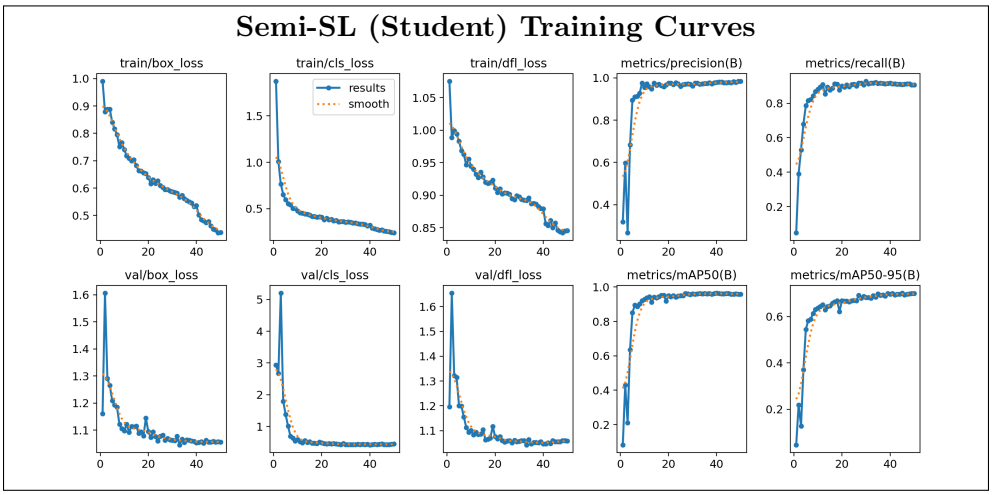

Figure 2: Training dynamics of the Student model during Semi-Supervised Learning (Pseudo-Labeling), showing convergence on the combined dataset.

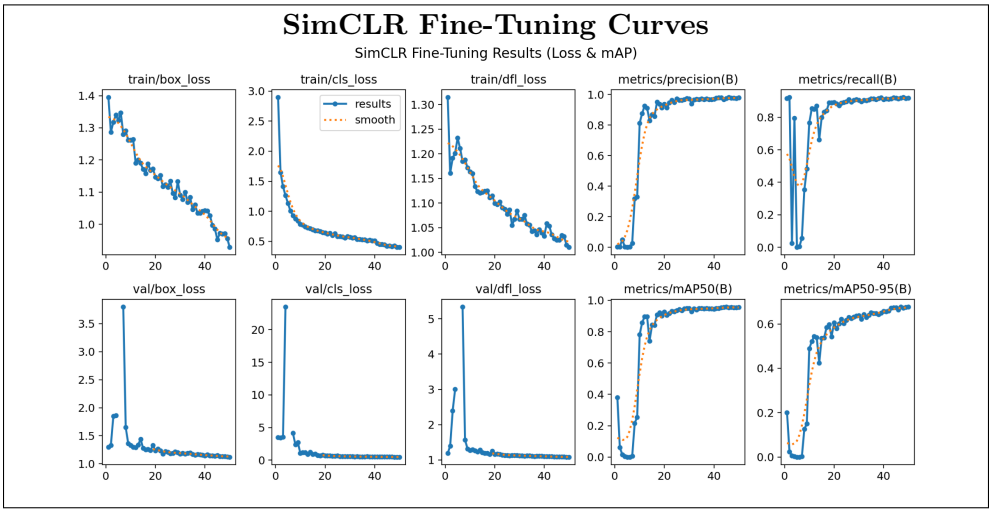

Figure 3: Loss evolution during fine-tuning of the SimCLR pre-trained backbone.

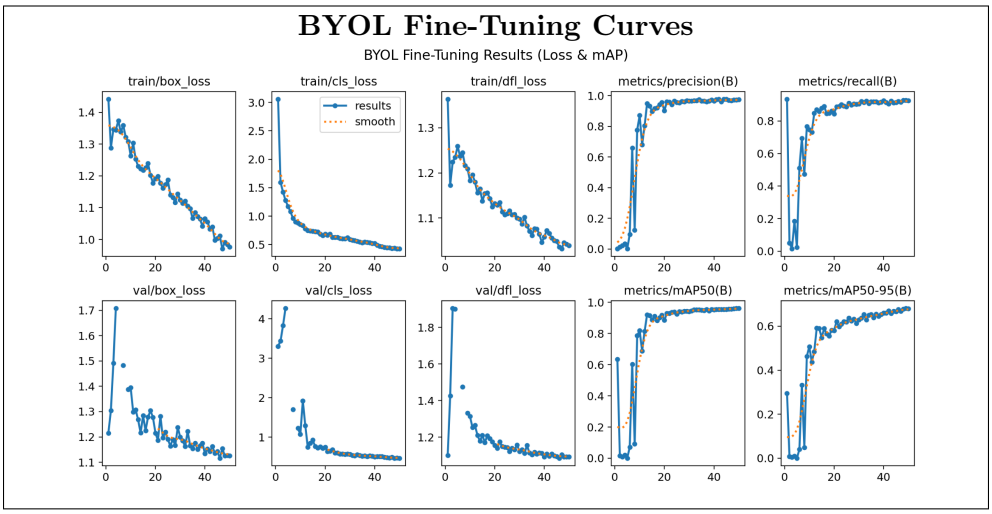

Figure 4: Loss evolution during fine-tuning of the BYOL pre-trained backbone.

5.4 Qualitative Results

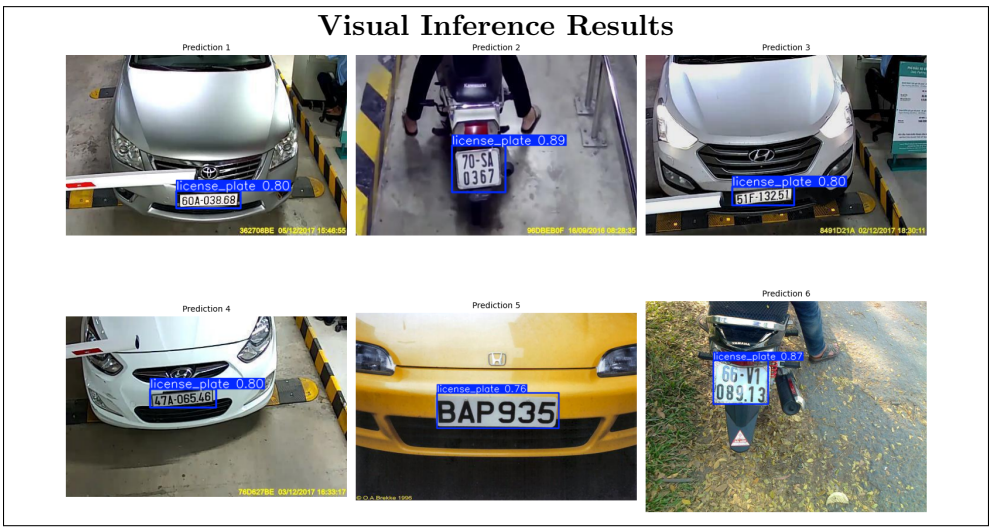

Figure 5: Visual Inference on Test Set (BYOL Fine-Tuned Model)

6 Discussion

Teacher vs. Student Performance: The Semi-Supervised experiment showed a clear improvement in the Student model over the Teacher. The Student achieved **0.9656 mAP@0.5** compared to the Teacher's **0.9604**, a gain of **+0.0052**. More significantly, the Student's **mAP@0.5:0.95** increased to **0.7103** (vs. 0.6879), proving that the additional unlabeled data helped the model localize objects more precisely.

SimCLR vs. BYOL: Both Self-Supervised methods successfully learned transferable features from unlabeled data. BYOL slightly outperformed SimCLR in terms of **mAP@0.5** (0.9632 vs. 0.9609). This performance edge may be attributed to BYOL's ability to learn from the target network's slowly evolving representations without relying on negative pairs, which can be computationally challenging to optimize with smaller batch sizes. While SimCLR required a larger batch size (256) to maintain stability, BYOL achieved superior convergence with a batch size of 128.

Computational Cost and Efficiency: While Self-Supervised Learning reduces the need for manual labels, it introduces a significant computational overhead. Pre-training the backbone (whether via SimCLR or BYOL) required an additional 50 epochs of training on the unlabeled data before the detection fine-tuning could even begin. In our experiments, BYOL was slightly more resource-efficient than SimCLR; it achieved convergence with a smaller batch size (128 vs. 256), reducing the VRAM requirement on the Tesla T4 GPUs. This makes BYOL a more practical choice for resource-constrained environments despite the higher complexity of its architecture (online/target networks).

Sensitivity Analysis: The performance of the Semi-Supervised Pseudo-Labeling pipeline is highly sensitive to the confidence threshold. We selected a threshold of **0.75** based on preliminary experiments. Lower thresholds (< 0.6) introduced significant label noise, causing the student model to drift and "hallucinate" plates in background clutter. Conversely, extremely high thresholds (> 0.9) resulted in too few pseudo-labels, starving the student of the additional data needed to improve generalization. The 0.75 threshold provided the optimal balance between label quantity and quality for this specific dataset.

7 Conclusion

This project successfully integrated Baseline, Semi-Supervised, and Self-Supervised methodologies for object detection. We demonstrated that:

1. **YOLOv12n** is the superior baseline architecture compared to v10n and v11n.
2. **Pseudo-Labeling** is highly effective when 20% of labels are available, recovering 99% of full performance.
3. **Self-Supervised Learning** is a viable strategy for data efficiency. Between the two methods, **BYOL** demonstrated superior stability and performance compared to **SimCLR**, achieving a mAP@0.5 of 0.9632.

Future work could explore hybrid approaches that combine BYOL pre-training with Pseudo-Labeling fine-tuning to maximize performance, or extend the evaluation to video-based domain adaptation where unlabeled temporal frames are abundant.

References

[1] Y. Tian, Q. Ye, and D. Doermann, "YOLOv12: Attention-Centric Real-Time Object Detectors," 2025.

[2] IEEE Sensors, "Real-Time Vehicle Classification and License Plate Recognition via Deformable Convolution-Based YOLOv8 Network," 2025.

[3] A. Wang et al., "YOLOv10: Real-Time End-to-End Object Detection," 2024.

[4] S. J. Joysingh et al., "U-Net and YOLOv8-based Pipeline for License Plate Recognition in Adverse Weather," 2025.

[5] V. Nascimento et al., "YOLOv8 and Faster R-CNN Performance Evaluation with Super-resolution in License Plate Recognition," 2024.

[6] Z. E. Vargoorani et al., "Efficient License Plate Recognition via Pseudo-Labeled Supervision with Grounding DINO and YOLOv8," 2025.

[7] B. Xu et al., "Efficient Teacher: Semi-Supervised Object Detection for YOLOv5," 2023.

[8] M. Xu et al., "End-to-End Semi-Supervised Object Detection With Soft Teacher," 2021.

[9] Y. C. Liu, C. Y. Ma, and Z. Kira, "Unbiased Teacher v2: Semi-supervised Object Detection for Anchor-free and Anchor-based Detectors," 2022.

[10] B. Chen et al., "Label Matching Semi-Supervised Object Detection," 2022.

[11] H. Zhou et al., "Dense Teacher: Dense Pseudo-Labels for Semi-supervised Object Detection," 2022.

[12] T. Chen et al., "A Simple Framework for Contrastive Learning of Visual Representations," 2020.

[13] J. B. Grill et al., "Bootstrap Your Own Latent (BYOL): A New Approach to Self-Supervised Learning," 2020.

[14] E. Xie et al., "DetCo: Unsupervised Contrastive Learning for Object Detection," 2021.

[15] X. Wang et al., "Dense Contrastive Learning for Self-Supervised Visual Pre-Training," 2021.

[16] M. Kotthapalli et al., "Self-Supervised YOLO: Leveraging Contrastive Learning for Label-Efficient Object Detection," 2025.

[17] C. Yang et al., "Plug and Play Active Learning for Object Detection," 2024.

[18] Y. Wang et al., "ALWOD: Active Learning for Weakly-Supervised Object Detection," 2023.

[19] Z. Wu et al., "LEOD: Label-Efficient Object Detection for Event Cameras," 2024.

[20] IEEE, "Few-shot Object Detection Algorithm Based on YOLOv8-Mamba-Like Linear Attention," 2024.

[21] Z. Ebrahimi Vargoorani et al., "Efficient License Plate Recognition via Pseudo-Labeled Supervision with Grounding DINO and YOLOv8," 2025.

[22] S. Jain, G. Mittal, and S. Kumar, "Automatic number plate recognition using CNN based self synthesized feature learning," 2017.

[23] H. A. Alani et al., "YOLO-Based Detection and OCR for Automatic Number Plate Recognition in Diverse Conditions," 2024.

[24] G. C. M. de Oliveira et al., "License Plate Super-Resolution and Recognition Under Real-World Conditions," 2025.

[25] Y. Z. Li et al., "Optimization of YOLOv7 Architecture for Object Detection in Low-Light Conditions," 2024.

[26] Roboflow Universe Projects, "License Plate Recognition Dataset," 2025.