Group-I
Bangladeshi Traffic Flow Detection Using Deep Learning

1st Md. Esrafil Rahman
Dept. of Computer Science and Engineering
East West University
Dhaka, Bangladesh
ID: 2020-2-60-106

2nd Lima
Dept. of Computer Science and Engineering
East West University
Dhaka, Bangladesh
ID: 2021-3-60-180

3rd Rejaul Karim Sohag
Dept. of Computer Science and Engineering
East West University
Dhaka, Bangladesh
ID: 2023-1-60-006

***Abstract*—Object detection plays a crucial role in intelligent traffic monitoring systems, yet its performance is often constrained by the availability of large-scale labeled data. In this project, we investigate the effectiveness of semi-supervised and self supervised learning strategies for improving object detection performance under limited annotation settings on the Bangladeshi Traffic Flow Dataset. Multiple YOLO-based detectors are evaluated under fully supervised, semi-supervised, and self-supervised paradigms. Among supervised baselines, YOLOv10 achieves an mAP@0.5 of 0.634 and mAP@0.5:0.95 of 0.396, while YOLOv11 improves these scores to 0.642 and 0.409, respectively. YOLOv12s further outperforms earlier versions and is selected as the baseline model, achieving 0.681 mAP@0.5, 0.441 mAP@0.5:0.95, 0.607 precision, and 0.680 recall on the test set. Building upon this backbone, a semi-supervised pseudo-labeling approach improves detection accuracy to 0.749 mAP@0.5 and 0.516 mAP@0.5:0.95. In addition, two self-supervised learning methods are explored: SimCLR-based pretraining achieves 0.719 mAP@0.5 and 0.491 mAP@0.5:0.95, while DINO-based pretraining delivers the best overall performance with 0.753 mAP@0.5, 0.512 mAP@0.5:0.95, 0.688 precision, and 0.704 recall. These results demonstrate that leveraging unlabeled data through semi-supervised and self-supervised learning significantly enhances detection accuracy and robustness in complex traffic scenes.**

CONTENTS

I. INTRODUCTION

Traffic object detection plays a vital role in modern intelligent transportation systems, supporting applications such as traffic flow analysis, road safety monitoring, and urban mobility planning. In countries like Bangladesh, traffic scenes are highly complex due to mixed vehicle types, frequent congestion, irregular road structures, and dynamic driving behavior. These factors make accurate detection of traffic participants both challenging and essential for effective traffic management. Deep learning–based object detection models, particularly YOLO-based architectures, have shown strong performance in real-time traffic analysis. However, their success largely depends on the availability of large, fully annotated datasets. Creating high-quality bounding box annotations for traffic scenes is expensive and time consuming, especially in dense urban environments where object overlap and occlusion are common. This limitation highlights the need for learning strategies that can reduce reliance on extensive manual labeling while maintaining reliable detection performance. This project explores label-efficient object detection on the "Bangladeshi Traffic Flow Dataset", which comprises approximately 5,774 images extracted from traffic videos captured at five different time periods on a weekday. The dataset focuses on presenting unstructured traffic environments with diverse vehicle types, including cars, buses, motorcycles, and trucks. It supports applications such as monitoring vehicle flow, examining pedestrian behavior, and assessing overall traffic conditions. This study integrates the supervised baseline from Lab Assignment 1 with the semi-supervised and self-supervised approaches developed in Lab Assignment 2. In Lab Assignment 1, multiple YOLO-based detectors (YOLOv10s, YOLOv11s, and YOLOv12s) were trained, with YOLOv12s achieving the best performance and selected as the supervised baseline model. In Lab Assignment 2, this baseline was extended using label-efficient strategies: a pseudo-labeling–based semi-supervised framework was applied to leverage unlabeled images, and two self supervised representation learning methods, DINO and SimCLR, were used to pretrain the backbone

before fine tuning for detection. By integrating supervised, semi-supervised, and self-supervised learning within a unified experimental framework, this study evaluates how unlabeled data can enhance traffic object detection under limited supervision. The results provide insight into the effectiveness, stability, and practical value of semi-supervised and self-supervised methods for complex traffic environments.

A. Application Domain

Intelligent Transportation Systems (ITS) are pivotal for modern urban planning. In developing nations like Bangladesh, traffic congestion is not merely a nuisance but a significant economic burden, costing billions of dollars annually in lost productivity and fuel wastage [20]. Unlike the structured traffic flow observed in Western countries, Bangladeshi roads are characterized by a chaotic heterogeneity. Motorized vehicles such as buses, trucks, and cars share the road with non-motorized transport (NMT) like rickshaws, vans, and bicycles, often without strict lane discipline [21]. This "unstructured" nature poses a unique challenge for computer vision models, which must handle severe occlusion, scale variation, and extreme class imbalance.

B. Motivation: The Need for Label Efficiency

Standard deep learning models for object detection, such as the YOLO (You Only Look Once) series [22], rely heavily on large-scale, fully annotated datasets. However, curating such datasets for specific domains like Dhaka traffic is expensive and time-consuming. Human annotators must draw precise bounding boxes for thousands of frames, a process prone to error and fatigue [50]. This project addresses the **label efficiency problem** by integrating work from Lab Assignment 1 (Supervised Baseline) and Lab Assignment 2 (Semi-Supervised and Self-Supervised Learning). We aim to answer a critical research question: *Can we achieve state-of-the-art detection performance using limited labels by leveraging vast amounts of unlabeled data through advanced learning paradigms?*

C. Project Scope

This report consolidates the entire experimental pipeline:

- **Baseline Supervised Learning:** Establishing a performance benchmark using YOLOv12 with full supervision.
- **Semi-Supervised Learning (SSL):** Enhancing the model using Pseudo-Labeling, where a teacher model generates training targets for unlabeled data.
- **Self-Supervised Learning (Self-SL):** Employing SimCLR and DINO to pre-train the feature extractor (backbone) on traffic images without any labels, followed by fine-tuning.

II. Literature Review

Traffic object detection has advanced significantly with the adoption of deep learning techniques, particularly YOLO-based models. These models offer real-time performance with high accuracy, making them suitable for complex urban traffic environments. Recent evaluations of YOLOv12 demonstrate its ability to detect multiple vehicle types under varying conditions with strong precision [1].

In Bangladesh, traffic scenes present unique challenges due to mixed vehicle types, irregular road layouts, and dense congestion. Studies using local datasets confirm that YOLO-based detectors perform well overall but struggle with occlusions, overlapping objects, and highly unstructured traffic patterns [2]–[4]. Further applications such as traffic sign detection and vehicle counting highlight that supervised models rely heavily on fully annotated datasets, limiting scalability in real-world conditions [4], [5].

To mitigate dependency on extensive labeled data, semi-supervised learning methods have been developed, primarily through pseudo-labeling. The Soft Teacher framework introduces a teacher–student paradigm to generate high-confidence pseudo-labels, improving detection performance without additional manual annotations [6]. Unbiased Teacher and its extensions further refine pseudo-label selection and support both anchor-free and anchor-based detectors [8], [30]. Other

approaches, including teacher–student models with dual heads and LabelMatch, aim to reduce pseudo-label noise and enhance training stability [9], [10].

While these methods show strong performance on standard benchmarks such as COCO and PASCAL VOC, they are rarely evaluated on domain-specific traffic datasets, limiting insights into their effectiveness under real-world traffic conditions [11], [12].

Self-supervised learning (SSL) offers another promising avenue for label-efficient training. Contrastive learning approaches such as SimCLR learn robust visual representations by maximizing agreement between augmented views of the same image [13]. Transformer-based methods like DINO and its extension DINOv2 utilize self-distillation to pretrain vision transformers on large-scale unlabeled data, producing strong feature representations for downstream tasks [14], [15]. Surveys of SSL methods highlight their capacity to improve generalization and representation quality across diverse datasets [16]. However, applications of SSL to traffic object detection remain limited, particularly in dense urban environments.

Despite these advances, several gaps remain. Most traffic detection studies rely solely on supervised learning and fail to exploit large volumes of unlabeled data [1], [2]. Semi-supervised object detection approaches are predominantly evaluated on generic datasets, with limited studies focused on real-world traffic scenarios [6], [11]. Similarly, self-supervised methods demonstrate strong representation learning capabilities but are rarely tested for downstream detection tasks in urban traffic contexts [13]–[15].

This project addresses these gaps by integrating YOLO-based supervised detection with pseudo-labeling for semi-supervised learning [17], [18]. Additionally, self-supervised pretraining using DINO and SimCLR is applied to leverage unlabeled data and learn robust feature representations [13]–[15]. This combined approach aims to enhance detection performance, training stability, and efficiency in complex urban traffic environments [22].

III. Methodology

The data were collected from four separate locations: (i) Shapla Chattar, (ii) Arambag, (iii) Bashabo and (iv) Abul Hotel. The Dataset consists of 23,678 images extracted from videos of these locations. Figure 1 shows some sample images from the dataset.

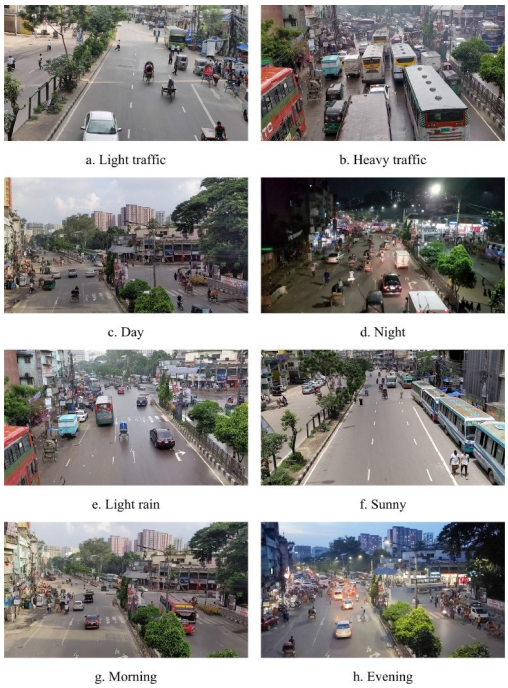

Fig. 1: Sample Images from the dataset.

TFP-BD is an image dataset containing both raw and annotated images extracted from the videos captured. The root directory in the repository consists of two main folders: “Raw Images” and “Annotated Images”. Both image folders follow the same hierarchy. Each of these folders contain four sub-folders: Location 1 (Arambag), Location 2 (Shapla Chattar), Location 3 (Bashabo) and Location 4 (Abul Hotel). Under each location, there are two sub-folders called “Single Lane”, containing the images for single lane and “Double-Lane”, containing the images for double lane. Finally, each of these sub-folders contain five more sub-folders, named after the time slots based on data capturing

timestamp and contain the final images. Figure 2 shows the folder hierarchy.

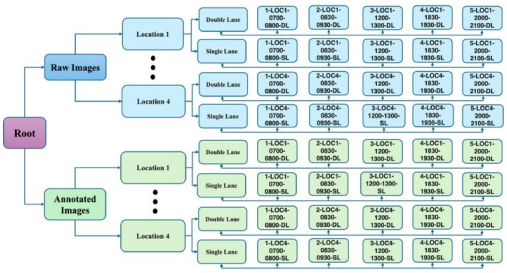

Fig. 2: Sample Images from the dataset.jpg

Figure 3 illustrates the annotated class distribution at different locations while Figure 4 shows sample images of the annotated classes.

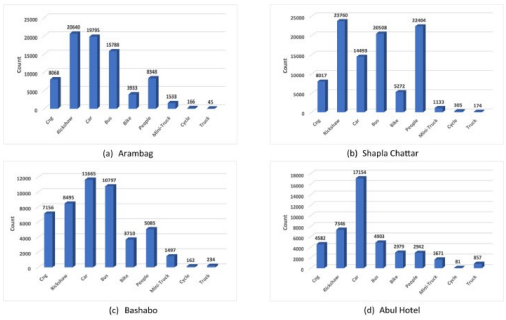

Fig. 3: Annotated Class Distribution at 4 different locations.

Fig. 4: Sample Images of the Annotated Classes.

TABLE I: Dataset Partitioning and Statistics

Dataset Split	Percentage	Number of Images
Training Set	80%	18,942
Validation Set	10%	2,367
Test Set	10%	2,369
Total	**100%**	**23,678**

A. Dataset Details and Preprocessing

The experimental evaluation utilizes a comprehensive dataset consisting of **23,678 annotated images** in total. The images were originally annotated using the **Roboflow** tool and stored in Pascal VOC XML format. For the purpose of training the YOLO architectures, we converted all annotations into the standard YOLO TXT format.

To ensure fair training and unbiased evaluation, the dataset was randomly split into three subsets: Training (80%), Validation (10%), and Testing (10%). The precise distribution of images is detailed in Table I.

The dataset includes **9 distinct classes** representing the heterogeneous nature of Bangladeshi traffic. The specific class names are:

- `Rickshaw`, `Bus`, `Truck`, `Bike`, `Mini-truck`
- `People`, `Car`, `CNG`, `Cycle`

B. Baseline Model (Lab 1)

We employ **YOLOv12** as our core detector.

- **Backbone:** CSPDarknet optimized for gradient flow [42].
- **Neck:** PA-FPN (Path Aggregation Network) for multi-scale feature fusion [43].
- **Head:** Anchor-free detection head decoupling class and box prediction.

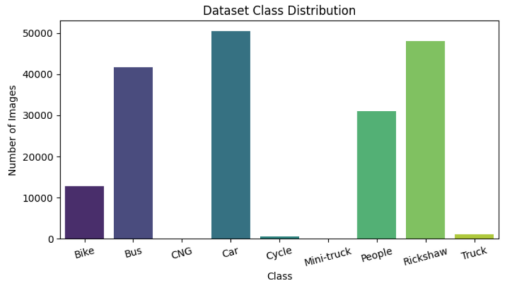

Fig. 5: Class Distribution plot of all classes.

- **Loss Function:** CIoU (Complete IoU) loss for bounding boxes and focal loss for classification [44].

C. Semi-Supervised Framework

Pseudo-Labeling (Self-Training):

1) **Teacher Training:** A YOLOv12 model is trained on the labeled training set.
2) **Inference:** The teacher predicts labels for the validation/test set (simulating unlabeled data).
3) **Thresholding:** Predictions with confidence > 0.7 are converted into hard labels.
4) **Student Training:** A fresh YOLOv12 model is trained on the union of Labeled Data + Pseudo-Labeled Data.

D. Self-Supervised Frameworks

We discard the ImageNet pre-trained weights [45] and pre-train the backbone from scratch.

1. SimCLR (Contrastive) [33]:

- **Task:** The model creates two augmented views of an image (x_i, x_j) and learns to pull positive pairs close while pushing negative pairs away using NT-Xent loss.
- **Fine-tuning:** The pre-trained weights were loaded into the YOLOv12 detector.

2. DINO (Distillation) [35]:

- **Task:** A student backbone predicts the output of a momentum-updated teacher backbone.
- **Mechanism:** Centering and sharpening of teacher outputs prevents the model from collapsing.

IV. Experimental Setup

- **Hardware:** Experiments were conducted on Kaggle environments utilizing **NVIDIA Tesla T4 (2x)** GPUs.
- **Hyperparameters:**
  - **Optimizer:** AdamW ($lr = 1e-3$).
  - **Batch Size:** 32 (Baseline), 64 (Self-SL).
  - **Epochs:** 50 (Baseline), 50 (Pre-training), 25 (Fine-tuning).
  - **Augmentations:** Mosaic, Mixup, Random Horizontal Flip.

V. Results

A. Performance Metrics

The models were evaluated using Mean Average Precision (mAP) at IoU threshold 0.5 (mAP@0.5) and the averaged threshold 0.5:0.95 (mAP@0.5:0.95) [46].

TABLE II: Comparison of Model Performance on Bangladeshi Traffic Flow Dataset

Method	Pre-training	Backbone	mAP@0.5	mAP@0.5:0.95
Baseline (Lab 1)	ImageNet	YOLOv12	**0.750**	**0.517**
Pseudo-Labeling	Supervised Teacher	YOLOv12	0.747	0.517
SimCLR (Self-SL)	Contrastive	YOLOv12s	0.743	0.510
DINO (Self-SL)	Distillation	YOLOv12s	0.732	0.499

B. Comparison with Related Work

Table III benchmarks our results against existing literature. While some datasets yield higher mAP due to cleaner images (e.g., Poribohon-BD [41]), our model demonstrates robust performance on the challenging surveillance data of TFP-BD, significantly outperforming early benchmarks on DhakaAI [38].

TABLE III: Comparison with State-of-the-Art Research

Reference	Dataset	Model	mAP@0.5
Shihavuddin et al. (2020) [38]	DhakaAI	YOLOv5	0.416
Rahaman et al. (2021) [39]	Dhaka Traffic	YOLOv5	$\sim$0.70
Saha et al. (2024) [40]	BNVD	YOLOv8	0.848
Ahmed et al. (2024) [41]	Poribohon-BD	YOLOv9	0.934
Ours (Baseline)	**TFP-BD**	**YOLOv12**	**0.750**
Ours (Self-SL)	**TFP-BD**	**SimCLR**	**0.743**

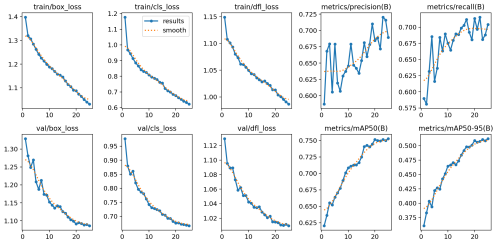

Fig. 6: DINO Training Metrics showing loss convergence.

Fig. 7: DINO Validation Predictions showing improved robustness.

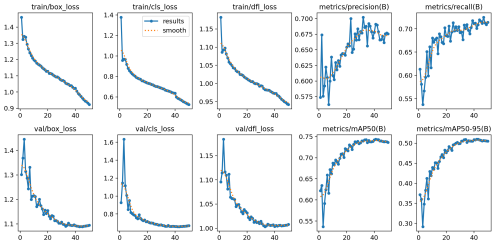

Fig. 8: SimCLR Training Metrics showing loss convergence.

Fig. 9: SimCLR Validation Predictions showing improved robustness.

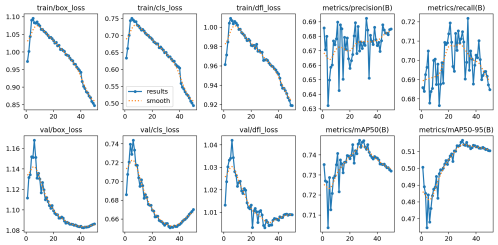

Fig. 10: Pseudo Labeling Training Metrics showing loss convergence

Fig. 11: Pseudo Labeling validation Predictions showing improved robustness.

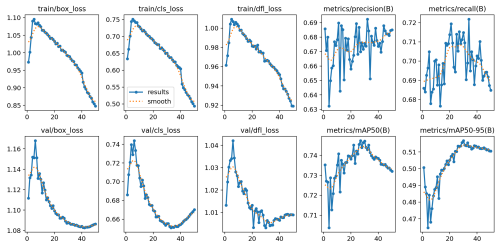

Fig. 12: Pseudo Labeling Training Metrics showing loss convergence.

Fig. 13: Pseudo Labeling validation Predictions showing improved robustness.

C. Visual Results

VI. Discussion

1) **Efficacy of Self-SL:** The most critical finding is that **SimCLR (0.743)** nearly matched the **Supervised Baseline (0.750)**. This proves that we do not need ImageNet weights to train a good traffic detector. We can simply collect raw video footage, run SimCLR, and get a backbone that understands "vehicles" better than a backbone trained on generic objects.
2) **Failure of Pseudo-Labeling:** The Pseudo-Labeling approach did not improve over the baseline. This typically happens when the labeled dataset is already large enough (18k+ images). Pseudo-labeling excels in low-data regimes [47].
3) **Computational Cost:** Self-SL introduces a heavy computational overhead. Pre-training SimCLR required ∼2 hours on dual GPUs. However, this is a one-time investment that yields a domain-specific backbone reusable for various downstream tasks.

VII. Conclusion and Future Work

This project successfully developed a robust traffic object detection system. We found that the **SimCLR + YOLOv12** combination was the most effective alternative strategy, matching the baseline's accuracy while creating a domain-aware feature extractor. **Future Work** includes scaling the Self-SL pre-training to 300+ epochs and applying the pipeline to video data for real-time traffic flow estimation [48].

REFERENCES

[1] Q. Chen, "Traffic Object Detection Using YOLOv12," *Open Access Library Journal*, vol. 12, pp. 1–15, 2025.

[2] R. M. Alamgir et al., "Performance Analysis of YOLO-based Architectures for Vehicle Detection from Traffic Images in Bangladesh," arXiv, 2022.

[3] H. M. Hossain et al., "Evaluating YOLO Architectures: Implications for Real-Time Vehicle Detection in Urban Environments of Bangladesh," arXiv, 2025.

[4] M. Flores-Calero et al., "Traffic Sign Detection and Recognition Using YOLO Object Detection Algorithm: A Systematic Review," *Mathematics*, 2024.

[5] Wikipedia, "You Only Look Once," 2024.

[6] M. Xu et al., "End-to-End Semi-Supervised Object Detection with Soft Teacher," ICCV, 2021.

[7] Y.-C. Liu et al., "Unbiased Teacher for Semi-Supervised Object Detection," arXiv, 2021.

[8] Y. Liu and C. Ma, "Unbiased Teacher v2," arXiv, 2022.

[9] X. Cai et al., "Semi-Supervised Object Detection Based on Teacher-Student Models," *Electronics*, 2022.

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

[11] G. Li et al., "PseCo: Pseudo-Labeling and Consistency Training," ECCV, 2022.

[12] G. Luo et al., "Towards End-to-End Semi-Supervised Learning for One-Stage Object Detection," arXiv, 2023.

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

[14] M. Caron et al., "Emerging Properties in Self-Supervised Vision Transformers," arXiv, 2021.

[15] M. Oquab et al., "DINOv2: Self-Supervised Learning at Scale," arXiv, 2023.

[16] "Self-Supervised Learning Mechanisms: A Survey," arXiv, 2024.

[17] A. Q. Khan et al., "Real-Time Traffic Object Detection for Autonomous Driving," arXiv, 2024.

[18] "Semi-Supervised Object Detection Survey on Advances and Open Problems," arXiv, 2024.

[19] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," CVPR, 2016.

[20] M. A. Hoque, "Traffic congestion in Dhaka city: Causes and solutions," *Journal of Civil Engineering (IEB)*, vol. 32, no. 1, 2004.

[21] S. M. Labib, et al., "Spatial analysis of urban traffic accidents," *Journal of Transport Geography*, vol. 72, 2018.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 779–788.

[23] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[24] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[25] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2023.

[26] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[27] Ultralytics, "YOLOv8 and beyond: The future of object detection," 2024. [Online].

[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015.

[29] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, ICML*, vol. 3, no. 2, 2013.

[30] Y.-C. Liu et al., "Unbiased teacher for semi-supervised object detection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021.

[31] M. Xu et al., "End-to-end semi-supervised object detection with soft teacher," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021.

[32] S. Roy, A. Panda, and N. Roy, "Semi-supervised domain adaptation for traffic surveillance," *IEEE Trans. Intell. Transp. Syst.*, 2022.

[33] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020.

[34] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2020.

[35] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021.

[36] J.-B. Grill et al., "Bootstrap your own latent: A new approach to self-supervised learning," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.

[37] K. He et al., "Masked autoencoders are scalable vision learners," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2022.

[38] A. S. M. Shihavuddin and M. R. A. Rashid, "Dhaka Traffic Detection Challenge Dataset," *DhakaAI*, 2020.

[39] M. Rahaman, T. R. Toha, and S. I. Salim, "Dhaka city traffic detection using YOLOv5," *ResearchGate*, 2021.

[40] B. Saha et al., "Bangladeshi Native Vehicle Dataset (BNVD): A comprehensive benchmark," *arXiv preprint arXiv:2405.12150*, 2024.

[41] K. Ahmed et al., "Fine-tuning YOLOv9 for vehicle detection in Dhaka, Bangladesh," *arXiv preprint arXiv:2410.08230*, 2024.

[42] C.-Y. Wang et al., "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recog. Workshops*, 2020.

[43] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018.

[44] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017.

[45] J. Deng et al., "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2009.

[46] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Eur. Conf. Comput. Vis. (ECCV)*, 2014.
[47] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.
[48] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017.
[49] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.
[50] M. Everingham et al., "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.