MD-YOLOV12: Two-Stage Feature Injection for Robust Tool Wear Detection

Jiaxin Cao
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: caojiaxin@st.xatu.edu.cn

Yu Bai
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: baiyv@xatu.edu.cn

***Abstract*—Tool wear detection in mechanical machining is a critical link for ensuring product quality and improving production efficiency. However, this field faces challenges such as scarce annotated data and interference from complex working conditions, making it difficult to deploy advanced detection models. To address the fundamental mismatch between model capacity and data availability, this paper proposes a novel data-efficient hybrid detection architecture named MD-YOLOV12. This architecture ingeniously integrates the rich general visual representations learned by the self-supervised vision transformer model DINOv3 with the YOLOv12 object detection framework. Specifically, we perform feature enhancement at two key locations: input preprocessing and the middle layer of the backbone network, thereby enhancing the model's perception and recognition capability for tool wear features without relying on massive annotated data. To validate the method's effectiveness, we constructed a specialized tool wear detection dataset containing 8083 high-resolution images, meticulously annotated into three categories: "No Wear," "Moderate Wear," and "Severe Wear." Extensive experimental results demonstrate that the proposed MD-YOLOV12 method surpasses existing state-of-the-art techniques in the tool wear detection task, providing a viable technical pathway for data-efficient industrial vision applications.**

Keywords-Tool Wear Detection; DINOv; YOLOv12; Object Detection

I. INTRODUCTION

In the field of mechanical machining [1], tool wear detection [2] represents a critical application direction, where automated monitoring is essential for ensuring machining quality, improving production efficiency, and controlling costs. Three interconnected systems — Computer Numerical Control(CNC) [3] machine online monitoring, smart manufacturing unit tool management, and precision machining quality control systems — fully demonstrate the importance and challenges of computer vision in this domain. Accurate detection of tool flank wear is critical across modern manufacturing systems. At the equipment level, it enables the prediction of tool lifespan and ensures machining accuracy. For smart manufacturing units, real-time assessment of wear states facilitates the optimization of tool-change strategies and overall production cycles. Furthermore, in precision machining systems, the sensitive detection of micro-wear is indispensable for preventing surface quality defects on workpieces and enabling timely maintenance interventions. These applications pose unique challenges for computer vision technology: they not only demand high-precision detection capabilities at the micron level but also require accurate identification and segmentation of specific wear areas under complex working conditions such as coolant splashing, chip entanglement, and variable lighting. Since machining quality and equipment safety are at stake, decision errors could lead to batch workpiece scrap or machine tool damage. A fundamental challenge in developing visual detection models for tool wear is the inherent scarcity of industrial image data. Unlike general-purpose datasets with millions of samples, tool-specific wear data under varying materials and parameters is typically limited to merely hundreds or thousands of instances. This scarcity primarily stems from practical constraints: limited machine operation time for data collection, production confidentiality protocols, and the proprietary

DOI: 10.2478/ijanmc-2025-0040

nature of specific machining conditions. Consequently, building robust models under such severe data constraints remains particularly difficult.

However, in the general field, we have large datasets like Microsoft Common Objects in Context (COCO) [4] that contain hundreds of thousands of images and millions of annotations, which are sufficient to train very powerful detection models. However, in the field of mechanical engineering, obtaining and labeling a large amount of data is extremely difficult and expensive. For instance, marking minor wear and tear on props not only requires professional knowledge but is also extremely time-consuming. Therefore, when a large-capacity You Only Look Once (YOLO) [5] model is trained on a dataset with only a few hundred or a few thousand images, overfitting is very likely to occur. The generalization ability of the knowledge learned by the model is very poor, and it will "fail to adapt" to real scenarios.This fundamental mismatch between model capacity and data availability represents a critical bottleneck in deploying state-of-the-art object detection frameworks for practical industrial applications [6]. How to enable the model to learn robust and universal features on small datasets is a key challenge in this field.

Recent advances in the field of self-supervised learning [7] have demonstrated that rich visual representations can be effectively learned from large-scale unlabeled data. In particular, the DINO [8] (self-distillation [9] with no labels) framework, specifically its latest iteration DINOv3 [10], has exhibited remarkable semantic understanding capabilities through self-supervised pre-training on vast image collections. In industrial scenarios, while it may be challenging to obtain large volumes of labeled tool wear images, it is relatively easy to collect massive amounts of unlabeled tool images or production line monitoring videos. DINOv3 has been pre-trained on an extremely diverse dataset of internet images, enabling it to "recognize" and understand fundamental visual patterns of objects—such as edges, textures, shapes, shadows, and material variations. This general visual knowledge provides a crucial foundation for identifying wear features on tool surfaces (e.g., grinding marks, chipped edges, contour changes). This means that the DINOv3 model can be used directly without the need for training from scratch on a tool wear dataset. Its feature extraction capability is "built-in" and does not rely on any manually defined "wear" labels, thereby perfectly bypassing the bottleneck of data annotation.

Motivated by the aforementioned challenges, we make the following contributions to data-efficient object detection in mechanical machining:

We propose a novel hybrid detection architecture, termed MD-YOLOV12, which effectively combines the DINOv3 self-supervised vision transformer with the YOLOv12 detection framework. Departing from prior strategies such as full backbone replacement or single-point feature injection, our approach strategically integrates Mamba blocks to capture long-range dependencies and enhance feature representation under data-limited settings.

To mitigate the scarcity of annotated data in this domain, we introduce a specialized Tool Wear Detection Dataset. It consists of 8,038 high-resolution images collected from real-world machining environments, each meticulously labeled into three wear categories critical for predictive maintenance: No Wear, Moderate Wear, and Severe Wear. This dataset serves as a publicly available benchmark to foster research in data-efficient industrial visual detection.

Through extensive experiments, we demonstrate that MD-YOLOV12 consistently outperforms existing state-of-the-art methods, validating its effectiveness in real-world mechanical machining applications.

II. METHOD

A. Model Architecture

To address the limitations of existing approaches in integrating self-supervised visual models with supervised detectors—particularly the computational inefficiency and semantic constraints caused by full backbone replacement and single-point feature injection—this study introduces a novel hierarchical semantic injection strategy. This method integrates DINOv3’s rich semantic representations into the YOLO architecture at two critical stages: semantic grounding at the input layer enhances the representational capacity of low-level features, while the subsequent Mamba2 [11] module effectively captures important features extracted by DINOv3 and models long-range dependencies with linear complexity, thereby cooperatively improving the detector's semantic robustness and spatial localization accuracy under data-scarce training conditions.

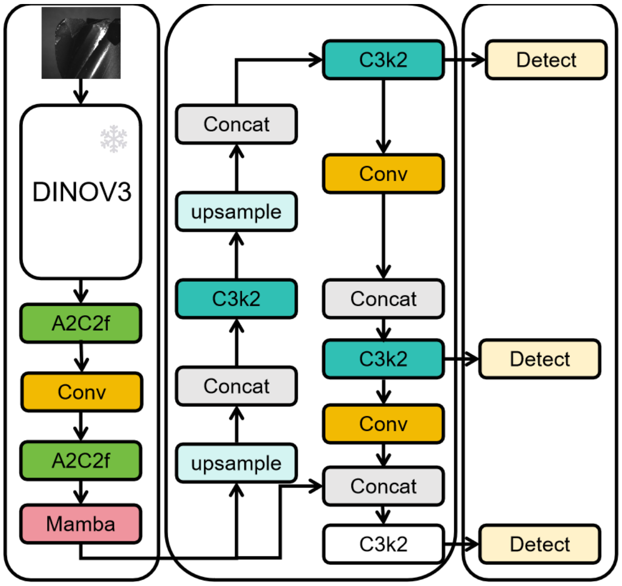

Figure 1. The main architecture of proposed DINO-YOLO model

Figure1, illustrates the YOLOv12 [12] architecture with DINOv3 as the backbone. The model processes 640×640 input images through a hierarchical pipeline consisting of the following sequential stages: input preprocessing, a DINOv3-enhanced backbone, deep backbone stages composed of Mamba2 and convolutional layers, a feature pyramid network that integrates top-down and bottom-up pathways, and a multi-scale detection head. The detection head operates at three different resolution levels: P3/8, P3/16, and P5/32. Two pre-trained DINOv3-ViT-B/16 modules (each with 86 million parameters and frozen weights) are incorporated. The entire architecture contains a total of 218 million parameters, of which 36 million are trainable, and requires 124 Giga Floating-point Operations (GFLOPs) for a 640×640 input.This dual-stage feature injection strategy is designed to address the diverse feature requirements across different abstraction levels in the object detection pipeline. By enhancing complementary aspects of semantic representation, it effectively improves the model's overall representational capacity and detection performance under data-scarce conditions.

B. DINOV3 Backbone

DINOv3, Meta AI's third-generation self-supervised vision transformer model, serves as an ideal backbone replacement for YOLO in tool

Wear detection applications due to its exceptional visual feature extraction capabilities, as illustrated in Figure 2. The model employs an efficient "self-distillation" framework that learns highly universal visual representations without relying on manually annotated data—a crucial advantage in industrial scenarios where tool wear data is often costly and challenging to label. Through its unique training mechanism involving dual enhanced views of the same image, DINOv3 develops robust multi-scale feature representations that naturally align with YOLO's multi-scale detection heads, enabling precise identification of wear regions across varying severity levels. The knowledge distillation process, formalized by the loss function:

$$\mathrm{L} = -\frac{1}{N}\sum_{i=1}^{N} P_t^{(i)}(x_2;T_t)\cdot\log P_s^{(i)}(x_1;T_s) \quad (1)$$

Here, L represents the total loss that the student network aims to minimize. The student's probability distribution P_s, generated from the first augmented view x_1, is trained to match the teacher's distribution P_t, which is generated from the second view x_2. The temperature parameters T_s and T_t control the sharpness of these distributions, helping the model focus on meaningful semantic relationships. Ensures the model captures subtle wear characteristics on tool surfaces while maintaining stability against environmental variations like lighting and scale changes. Furthermore, the teacher-student parameter update rule can be formurla as:

$$\theta_{\mathrm{t}} \leftarrow \lambda\theta_{\mathrm{t}} + (1-\lambda)\theta_{\mathrm{s}} \quad (2)$$

In this exponential moving average update, $\theta_t$and $\theta_s$denote the parameters of the teacher and student networks, respectively. The momentum coefficient λ, a value very close to 1, dictates the update speed.Guarantees reliable feature extraction during fine-tuning, allowing the pre-trained model to quickly adapt to specific tool wear domains even with limited samples. This integration creates a powerful tool wear detection system that accurately locates wear areas, quantifies degradation levels, and maintains consistent performance in complex industrial environments, thereby providing reliable technical support for tool condition monitoring in smart manufacturing applications.

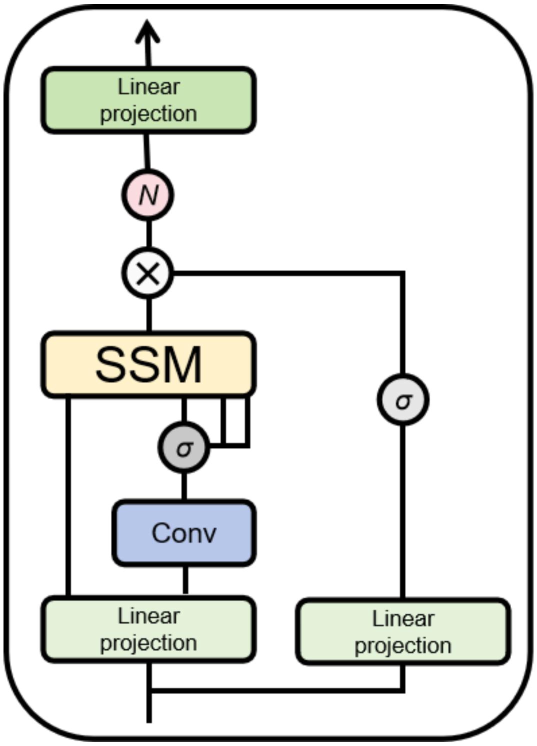

Figure 2. The architecture of Mamba block.

C. Mamba Block

To further augment the model's capability in capturing long-range dependencies and fine-grained spatial relationships within tool wear images, we integrate the Mamba block. As a state space model (SSM) [13] with data-dependent selection mechanisms, Mamba overcomes the computational limitations of traditional transformers while excelling at modeling complex global contexts. This is particularly beneficial for tool wear detection, where a single localized wear region may be correlated with distant visual cues across the entire tool surface. The core of the Mamba block involves a structured state space sequence transformation. It was illustrated in Figure 2. The continuous-time state space is defined by the following equations:

$$h^{'}(t) = Ah(t) + Bx(t) \qquad (3)$$

$$\mathrm{y}(t) = Ch(t) + Dx(t) \qquad (4)$$

Here, h(t) represents the hidden state at time tt, x(t) is the input signal, and y(t) is the output signal. The matrices A, B, C, and D are parameters that govern the system's dynamics.

For digital computation, the continuous system is discretized using a zero-order hold method with a timescale parameter Δ . This is achieved by transforming the continuous parameters (A, B) into their discrete counterparts ($\overline{\mathrm{A}}, \overline{\mathrm{B}}$):

$$\overline{\mathrm{A}} = \exp(\Delta A) \qquad (5)$$

$$\overline{\mathrm{B}} = (\Delta A)^{-1}(\exp(\Delta A) - 1) \cdot \Delta B \qquad (6)$$

The discrete-time computation for a given input sequence $\mathrm{x_k}$ is then performed via a linear recurrence or a parallelized global convolution:

$$h_{\mathrm{k}} = \overline{A}h_{k-1} + \overline{B}x_k \qquad (7)$$

$$y_{\mathrm{k}} = Ch_k + Dx_k \qquad (8)$$

By integrating the Mamba block after the DINOv3 backbone and before the YOLO detection heads, the feature maps are enriched with globally-aware, contextually refined representations. This enables the detection system to not only identify local wear spots but also understand their significance within the broader context of the entire tool's condition, leading to more robust and accurate wear level quantification and localization in complex industrial environments.

D. Mechanistic Analysis of the Synergy Between Mamba Blocks and DINOv3

The significant performance enhancement achieved by our proposed architecture is not merely a result of stacking advanced modules, but stems from the fundamental theoretical complementarity between the static semantic extraction of DINOv3 and the dynamic contextual modeling of Mamba. This synergy functions, driven by three specific mechanistic pathways:

First, Global Contextualization via Linear Complexity. While DINOv3 excels at extracting object-centric features via its attention mechanism, standard Vision Transformers often lack efficient global modeling at high resolutions due to quadratic computational complexity O(N^2). In contrast, the Mamba block introduces a linear-complexity global receptive field O(N). By treating the flattened feature map as a sequence, Mamba utilizes its recurrent state space model to propagate information across the entire image. This allows the detection head to correlate distant visual cues—such as relating shank discoloration to tip wear—without computational bottlenecks, ensuring the model perceives the holistic tool state rather than isolated patches.

Second, The “Selection Mechanism” as a Semantic Rectifier. Industrial scenarios are often plagued by high-frequency noise that DINOv3 might encode as salient features due to its high sensitivity. The core innovation of Mamba—the Data-Dependent Selection Mechanism—acts as a dynamic semantic filter. Unlike static convolutional layers, Mamba's input-dependent parameters allow the model to selectively "forget" or "remember" information. When the scan encounters irrelevant background noise, the model generates a smaller time-step to suppress state updates; conversely, when it detects wear patterns characterized by DINO's high-level semantics, the gate opens to integrate this information. This process effectively rectifies the feature stream, filtering out environmental interference while preserving the structural integrity of wear defects.

Third, Modeling the Spatial Continuity of Wear. Tool wear is inherently a continuous physical phenomenon rather than a set of discrete pixels. While DINOv3 processes images in discrete patches, potentially leading to fragmented representations, Mamba is derived from continuous-time systems. This inductive bias makes it exceptionally capable of modeling the spatial continuity of wear marks. By processing DINO features through Mamba, discrete patch embeddings are smoothed into coherent “wear trajectories.” This ensures that the predicted bounding boxes align with the physical reality of the defect's shape, significantly improving localization accuracy and Intersection over Union (IoU) scores.

III. EXPERIMENT

A. Experimental Settings

Data and Evaluation. To evaluate the performance of the proposed MD-YOLOV12, We collected a dataset of approximately 8,038 tool wear images, called Tool Wear Detection Dataset (TWDD) , including normal wear, crater wear, flank wear, and groove wear. The worn tools were annotated, with 70% of the data used as the training set, 20% as the validation set for testing, and 10% as the test set. We employ mAP50 [14] as the evaluation metrics. Figure 3. Presents visual examples of each type of defect.

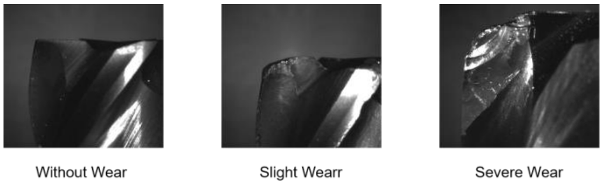

Figure 3. Examples of TWDD dataset.

Training Settings. We employ different variant of DINOv3 as the backbone of MD-YOLOv12. Since DINOv3 is already a powerful feature extractor, we freeze its parameters during training to reduce the number of trainable parameters and model complexity. It underwent training employing the Adam optimizer, lauded for its adaptive learning rate features that expedite model convergence. The initial learning rate was configured at 0.003, which diminishes the learning rate by a factor of 0.1 in the absence of performance enhancements on the validation set over 20 consecutive epochs. Training extended across 200 epochs, with early stopping mechanisms in place to avert overfitting. This halts the training process if the validation loss fails to improve over 20 consecutive epochs. To enhance model robustness against real-world variations, data augmentation techniques such as random rotations, translations, and scaling (detailed in the Dataset subsection) were applied to the training images. Our model is implemented using PyTorch [15] and trained on two RTX 3090 GPUs.

B. Ablation Study

As shown in Table 1, demonstrating that optimal integration strategies depend strongly on YOLO backbone scale, DINOv3 model variant.The ablation results demonstrate that optimal DINO-YOLO configurations vary systematically with YOLO model scales, refuting the hypothesis that a single universal integration strategy exists for all model sizes. When employing DINOv3 ViT-L/16 as the backbone with YOLOv12-M configuration, the model achieves its best performance of 58.78% mAP50. The large backbone network possesses sufficient capacity to effectively integrate semantic features from two DINOv3 injection points without causing architectural interference or gradient flow complications. Furthermore, we observe that the introduction of the Mamba block brings performance improvements to all configured models, with optimal performance consistently achieved when both DINOv3 and Mamba block are incorporated simultaneously.

Based on the ablation findings discussed above, practitioners should select configurations according to deployment requirements and resource constraints. For application scenarios where computational resources permit and maximum detection accuracy is prioritized, the optimal configuration when employing a large-scale backbone is YOLOv12-M with DINOv3 ViT-L/16. We show visual in Figure 4. This validates that the integration of high-quality DINO features with strategic architectural design enables efficient models to outperform larger counterparts. For resource-constrained edge deployment scenarios requiring small-scale architectures, practitioners should adopt YOLOv12-S with DINOv3 ViT-B/16.s

C. Comparison with State-of-the-Art Methods

To validate the adaptability of our model in complex industrial scenarios, specifically under conditions of strong interference and micro-wear, we conducted a comprehensive evaluation. We compared our proposed MD-YOLOv12 against seven state-of-the-art (SOTA) object detection methods: YOLOv3 [16], YOLOv5 [17], YOLOv7 [18], Gold-YOLO [19], YOLOv8 [20], YOLOv10 [21], and the baseline YOLOv12.

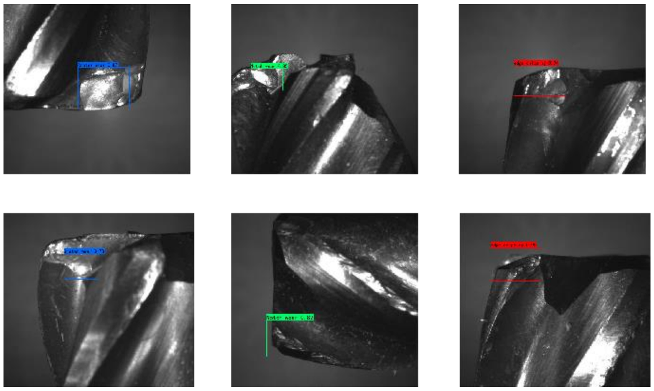

Figure 4. Visual result for MD-YOLOv12 in out test cases.

TABLE I. ABLATION STUDY OF DM-YOLOV12 ARCHITECTURAL CONFIGURATIONS ON TWDD DATASET.

Model Configuration	YOLOv12 Scale	DINOV Variant	mAP50
YOLOv12(baseline)	L	ViT-B/16	0.4325
YOLOv12+Mamba	L	ViT-B/16	0.4673
YOLOv12+Mamba+DINOV3	L	ViT-B/16	**0.5497**
YOLOv12(baseline)	L	ViT-L/16	0.4461
YOLOv12+Mamba	L	ViT-L/16	0.4871
YOLOv12+Mamba+DINOV3	L	ViT-L/16	**0.5831**
YOLOv12(baseline)	M	ViT-B/16	0.4152
YOLOv12+Mamba	M	ViT-B/16	0.4367
YOLOv12+Mamba+DINOV3	M	ViT-B/16	**0.5579**
YOLOv12(baseline)	M	ViT-L/16	0.4356
YOLOv12+Mamba	M	ViT-L/16	0.4651
YOLOv12+Mamba+DINOV3	M	ViT-L/16	**0.5878**
YOLOv12(baseline)	S	ViT-B/16	0.4051
YOLOv12+Mamba	S	ViT-B/16	0.4571
YOLOv12+Mamba+DINOV3	S	ViT-B/16	**0.5481**
YOLOv12(baseline)	S	ViT-L/16	0.4517
YOLOv12+Mamba	S	ViT-L/16	0.4401
YOLOv12+Mamba+DINOV3	S	ViT-L/16	**0.5591**

As shown in the experimental results, MD-YOLOv12 (specifically the M-scale variant) demonstrates significant superiority over baseline models. In terms of mAP50, our model achieves improvements of 24.53%, 19.05%, 16.81%, 15.62%, 8.24%, and 17.80% over YOLOv3, YOLOv5, YOLOv7, Gold-YOLO, YOLOv8, and YOLOv10, respectively. Notably, even compared to the latest YOLOv12 baseline, MD-YOLOv12 secures a 6.62% performance gain.

TABLE II. QUANTITATIVE COMPARISON WITH STATE-OF-THE-ART METHODS.

Model	Param.(M)	mAP50
YOLOv3-M	32.1	0.3425
YOLOv5-M	36.1	0.3973
YOLOv7-M	34.9	0.4197
Gold-YOLO-M	41.3	0.4316
YOLOv8-M	25.9	0.5054
YOLOv10-M	24.8	0.4098
YOLOv12-M	17.1	0.5216
MD-YOLOv12-M	16.4	**0.5878**

This substantial improvement highlights the advanced nature of MD-YOLOv12 in tool wear detection. While models like YOLOv10 offer a decent balance between speed and accuracy, MD-YOLOv12 exhibits better robustness in distinguishing tiny wear patterns from complex background noise. It effectively solves the problem of missed detections caused by micro-scale features and environmental interference, offering the most optimal trade-off for real-time industrial inspection.

IV. CONCLUSIONS

In the field of mechanical machining, tool wear detection is crucial for ensuring machining quality and efficiency, yet it faces severe challenges such as scarce annotated data and complex working conditions. Traditional large-capacity detection models (e.g., YOLO) trained on large-scale annotated data are prone to overfitting in such few-shot scenarios, resulting in insufficient generalization capability. This study innovatively proposes a hybrid detection architecture named MD-YOLOV12, which strategically integrates the powerful visual semantic understanding capability of the self-supervised pre-trained model DINOv3 with the YOLOv12 detection framework. By injecting general features at two complementary locations—input preprocessing and the middle layer of the backbone network—the proposed method effectively overcomes the data bottleneck and enables robust feature learning without relying on large amounts of annotated data. Extensive experiments on a specialized tool wear dataset containing 8083 high-resolution images demonstrate that this solution outperforms existing advanced methods, providing an effective approach for data-efficient object detection in industrial scenarios.

REFERENCES

[1] Suh S H, Kang S K, Chung D H, et al. Theory and design of CNC systems[M]. London: Springer London, 2008.

[2] Liang S Y, Dornfeld D A. Tool wear detection using time series analysis of acoustic emission[J]. 1989.

[3] Xu X W, Newman S T. Making CNC machine tools more open, interoperable and intelligent—a review of the technologies [J]. Computers in Industry, 2006, 57(2): 141-152.

[4] Lin T Y, Maire M, Belongie S, et al. Microsoft coco: Common objects in context[C]//European conference on computer vision. Cham: Springer International Publishing, 2014: 740-755.

[5] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[6] Kleijnen J P C. Validation of models: statistical techniques and data availability[C]//Proceedings of the 31st conference on winter simulation: Simulation---a bridge to the future-volume 1. 1999: 647-654.

[7] Liu X, Zhang F, Hou Z, et al. Self-supervised learning: Generative or contrastive[J]. IEEE transactions on knowledge and data engineering, 2021, 35(1): 857-876.

[8] Caron M, Touvron H, Misra I, et al. Emerging properties in self-supervised vision transformers[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 9650-9660.

[9] Zhang L, Song J, Gao A, et al. Be your own teacher: Improve the performance of convolutional neural networks via self distillation[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 3713-3722.

[10] Sim éoni O, Vo H V, Seitzer M, et al. Dinov3[J]. arXiv preprint arXiv:2508.10104, 2025.

[11] Dao T, Gu A. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality[J]. arXiv preprint arXiv:2405.21060, 2024.

[12] Tian Y, Ye Q, Doermann D. Yolov12: Attention-centric real-time object detectors[J]. arXiv preprint arXiv:2502.12524, 2025.

[13] Hamilton J D. State-space models[J]. Handbook of econometrics, 1994, 4: 3039-3080.

[14] Yue Y, Finley T, Radlinski F, et al. A support vector method for optimizing average precision[C]//Proceedings of the 30th annual

international ACM SIGIR conference on Research and development in information retrieval. 2007: 271-278.

[15] Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library[J]. Advances in neural information processing systems, 2019, 32.

[16] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[17] Zhang Y, Guo Z, Wu J, et al. Real-time vehicle detection based on improved yolo v5[J]. Sustainability, 2022, 14(19): 12274.

[18] Wang C Y, Bochkovskiy A, Liao H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 7464-7475.

[19] Wang C, He W, Nie Y, et al. Gold-YOLO: Efficient object detector via gather-and-distribute mechanism[J]. Advances in Neural Information Processing Systems, 2023, 36: 51094-51112.

[20] Sohan M, Sai Ram T, Rami Reddy C V. A review on yolov8 and its advancements[C]//International Conference on Data Intelligence and Cognitive Informatics. Springer, Singapore, 2024: 529-545.

[21] Wang A, Chen H, Liu L, et al. Yolov10: Real-time end-to-end object detection [J]. Advances in Neural Information Processing Systems, 2024, 37: 107984-108011.