

Article

The Target Detection of Wear Particles in Ferrographic Images Based on the Improved YOLOv8

Jinyi Wong, Haijun Wei *, Daping Zhou and Zheng Cao

Merchant Marine College, Shanghai Maritime University, Shanghai 201306, China;
wangjinyi990622@163.com (J.W.); dpzhou@shmtu.edu.cn (D.Z.); cz809007607@163.com (Z.C.)

* Correspondence: haijun_welson@163.com

Abstract: An enhanced YOLOv8 algorithm is proposed in the following paper to address challenging issues encountered in ferrographic image target detection, such as the identification of complex-shaped wear particles, overlapping and intersecting wear particles, and small and edge-wear particles. This aim is achieved by integrating the main body network with the improved Deformable Convolutional Network v3 to enhance feature extraction capabilities. Additionally, the Dysample method is employed to optimize the upsampling technique in the neck network, resulting in a clearer fused feature image and improved precision for detecting small and edge-wear particles. In the head network, parameter sharing simplifies the detection head while enhancing convergence speed and precision through improvements made to the loss function. The experimental results of the present study demonstrate that compared to the original algorithm, this enhanced approach achieves an average precision improvement of 5.6% without compromising the detection speed (111.6FPS), therefore providing valuable support for online monitoring device software foundations.

Keywords: YOLOv8; wear particle detection; point sampling; optimization of loss function



Citation: Wong, J.; Wei, H.; Zhou, D.; Cao, Z. The Target Detection of Wear Particles in Ferrographic Images Based on the Improved YOLOv8.

Lubricants **2024**, *12*, 280. <https://doi.org/10.3390/lubricants12080280>

Received: 20 June 2024

Revised: 27 July 2024

Accepted: 29 July 2024

Published: 5 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

At present, many mechanical devices in service experience sequential failures, primarily due to wear and aging [1,2]. To appropriately monitor the wear status of such mechanical devices without causing downtime or operational disruptions, tribological analysis relies on signals generated by the equipment itself and substances produced during operation. This approach allows for the restoration of the internal working state of the equipment through methods such as analyzing faults using collected vibration and heat information. One technique that can be utilized for this purpose is ferrographic analysis, which constitutes one of the research methods developed within tribological informatics. This technique involves obtaining wear particles that have settled in used lubricating oil to analyze their morphology, size, color, and other characteristics. Through this form of analysis, it becomes possible to diagnose both the lubrication condition and degree of wear of the equipment. Furthermore, if there is any indication of potential failure trends, this technique may also enable early inference regarding both the location and cause of any impending issues, thus facilitating proactive troubleshooting [3–6]. While ferrographic analysis offers significant advantages in this context, its widespread application is constrained by substantial human resources and time costs associated with acquiring and interpreting ferrographic images. The development of artificial intelligence has provided new insights for tribological informatics by enabling computers to autonomously learn relationships within output data from tribological systems. This not only saves time and effort but also allows for integration across various data types, resulting in timely and accurate monitoring, prediction, and optimization capabilities [7,8]. In pursuit of realizing intelligent recognition techniques for iron particle images within ferrographical analysis, as well as providing software support for online oil monitoring device implementation efforts,

in the following paper, a YOLOv8 model-based algorithm for recognizing ferrographic images is proposed.

Upon the emergence of intelligent image recognition as a focal point and research hotspot, scholars in the field of tribology began to conduct extensive investigations on this topic [9]. First, Wang et al. integrated the traditional watershed algorithm with the gray clustering algorithm, resulting in significantly improved segmentation outcomes compared to those of previous algorithms [10]. Moreover, the clustering algorithm serves as an essential foundation for various convolutional neural network algorithms. Subsequently, Peng et al. attempted to construct a search tree model using SVDD (Support Vector Data Description), K-means clustering algorithm, and support vector machine (SVM) techniques, achieving notable success in classifying cutting wear particles, red and black oxides, and spherical models [11]. However, challenges persisted regarding classifying similar wear particles; furthermore, prior to inputting parameters into the algorithm, manual operation was required for data collection and processing, which limited the model's level of intelligent recognition.

Once convolutional neural networks (CNNs) started to revolutionize the field of computer vision, their impact began to garner attention from experts across various disciplines. Peng et al. proposed a CNN-based model called FECNN for ferrographic image identification and primarily focused on fatigue wear particles, oxidation wear particles, spherical wear particles, and cases where two or more types appeared simultaneously [12]. Their experimental results demonstrated the model's high accuracy; however, the model's image processing speed was relatively slow. In the same year, Peng et al. successfully validated an algorithm in a rolling bearing wear particle monitoring system by combining a Gaussian background mixture model with a spot detection algorithm and attempted to extract the three-dimensional features of wear particles from an online monitoring system [13]. In another study, Wang S. aimed to enhance model training accuracy by extracting three-dimensional information from ferrographic images and proposed constructing a two-level classification tree that a combined backpropagation (BP) neural network with CNNs for classifying single target wear particle images; nevertheless, due to the relatively simplistic construction of the CNNs, the model performed poorly in distinguishing similar wear particles [14–16]. Subsequently, Wang S. addressed this issue in their study by simplifying the processing of simple wear particle types using fuzzy selection methods and analyzing and training similar texture sample sets through principal component analysis and backpropagation neural networks based on existing classification structures [17]. Furthermore, Peng Y.P. advanced previous work by proposing the use of convolutional neural networks for detection along with support vector machines for classification, resulting in a suitable level of accuracy on a self-made four-class dataset [18]. However, the method still requires improvement in terms of its ability to identify overlapping wear particles. Fan S.L. introduced FFWR-Net, which made significant advancements in wear particle classification; however, it unfortunately lacked the ability of detection [19]. Vivek J. employed Alexnet for feature extraction from the ferrogram and utilized 15 classifiers for classification. Their study results demonstrated that instance-based K-nearest-neighbor classification yielded the best performance. It achieved a 95% accuracy rate [20].

The authors of previous studies have predominantly employed shallow neural networks for the simple classification and detection of worn particles. However, in recent years, the authors of previous studies have gradually grown interested in utilizing deep convolutional neural networks (DCNNs) for the intelligent detection of worn particles. Notably, the DCNN field primarily encompasses YOLO series, RCNN series, and DINO algorithms. Compared to earlier approaches, these algorithms have significantly simplified the preprocessing process and integrated detection and classification functions to achieve target detection. Moreover, some even offer video stream tracking capabilities. One of the most significant contributions is the YOLO (You Only Look Once) framework, first introduced in 2016 and subsequently refined through numerous revisions. The inception of YOLOv1 established the fundamental concept of treating object detection as a regression

problem, involving the following key steps for achieving object detection: ① dividing the image into uniform regions and predicting objects; ② generating multiple bounding boxes, confidence scores, and class probabilities within these bounding boxes; ③ calculating the loss function for bounding boxes, confidence scores, and class probabilities to expedite the convergence of bounding boxes and the enhancement of class probability during training; ④ retaining the most probable bounding boxes containing target objects through non-maximum suppression (NMS); and ⑤ validating results and iteratively refining training based on learning experiences until accuracy stabilizes. The distinctive features of the YOLO framework include end-to-end training, multi-scale prediction, and rapid detection [21]. The above factors have captured the attention of scholars. Jia et al. utilized YOLOv3 to test a self-built dataset comprising six types of wear particles and demonstrated excellent performance in detecting cutting wear particles, spherical wear particles, and severe sliding wear particles [22]. Nevertheless, this model suffers challenges relating to its large size and deployment. To address this issue, He et al. employed an optimized version of the YOLOv5 model with reduced dimensions while incorporating attention mechanisms. This modification yielded promising experimental results without compromising processing speed [23]. Fan H. W. employed a fusion of YOLOv3 and DarkNet53 models, following two rounds of transfer learning, and achieved a high classification recognition rate for the custom-made gearbox wear dataset [24]. Shi X.F. conducted experiments using YOLOv5s on a highly intricate custom iron spectrum dataset containing 23 types of abrasive materials, achieving an accuracy rate of 40.5% [25].

However, the detection of wear particles still poses challenges regarding identifying complex-shaped wear particles, overlapping wear particles, and edge-blurred and small target wear particles. In the following paper, we propose an enhanced YOLOv8 intelligent model for wear particle identification based on point sampling in ferrographic images. The main contributions and innovations of our work are outlined below:

1. To address the challenge of detecting edge-wear particles and small wear particles in ferrography, we introduce the C2f-DCNv3 module [26], inspired by ViT, into the YOLOv8 main network to enhance global perception and improve the feature representation of edge-wear particles and small wear particles, therefore enhancing recognition accuracy.
2. In the neck network, Dysample [27] upsampling is employed instead of the original nearest-neighbor upsampling method to generate higher-quality feature maps, further improving the ability to distinguish overlapping wear particles during information fusion.
3. Additionally, a shared parameter optimization strategy is utilized in the head network to reduce parameter count while achieving specified performance improvements.
4. Instead of using a single loss function as done previously, we adopted WISEPiou, which combines WISE-IoU [28] and POWERFUL-IoU [29] loss functions for faster model convergence and enhanced recognition accuracy.

2. Improved YOLOv8 Model Architecture

2.1. Introduction to YOLOv8

The UltraLytics team, known for their development of YOLO models from version 3 to version 5, has further enhanced the YOLOv8 by incorporating new technologies within the v5 framework [30–32]. Similar to its predecessor, YOLOv5, YOLOv8 offers model options of varying scales (N/S/M/L/X). In the study presented herein, we aimed to address the practical requirements of online detection and opted for the nanoscale model.

As shown in Figure 1. Compared to YOLOv7 ELAN, YOLOv8 incorporates C2f modules in both the main network and the neck network, introducing a more diverse range of gradient flows [33]. The head network is transformed into the prevalent decoupled structure, segregating the classification and detection heads while transitioning from an anchor-based to an anchor-free approach. Furthermore, the loss calculation adopts Task Aligned Assigner's positive sample assignment strategy and introduces distribution focal loss.

Through a series of structural reconstructions, YOLOv8 achieves significant improvements in detection accuracy, effectively addressing the limitations of single-stage algorithms.

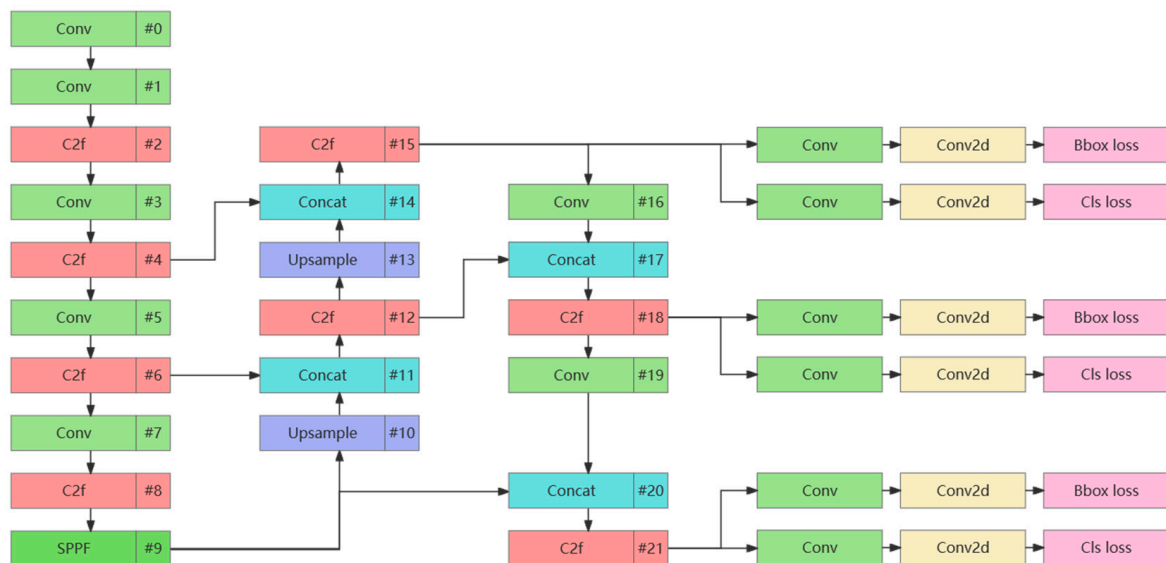


Figure 1. The structure of the YOLOv8 model. The model primarily comprises the backbone network spanning from layer #0 to #9 on the left side, the neck network spanning from layer #10 to #21, and the three head networks on the right side dedicated to detecting targets of varying scales.

2.2. DCNv3

Deformable Convolutional Networks (DCNs) are dynamic sparse convolution operators that differ from traditional CNNs. They enhance a model's ability to perceive target object shapes by introducing compensation values after convolution and pooling layers. These compensation values, determined by the shape of the target object, are updated in size and direction through learning to reduce interference from irrelevant factors. Consequently, DCNs can extract more flexible and accurate features related to wear particle size, contour, and texture. Unlike conventional CNN convolution operations, DCNs exhibit good generalization ability without requiring additional training data, such as image enhancement. To address variable features in wear particle images, such as size and shape, we employed the third-generation version of a DCN in the present study due to its higher computational efficiency, more stable gradient propagation, and richer spatial features.

The formula for DCNv3 is as follows:

$$y(p_0) = \sum_{g=1}^G \sum_{k=1}^K w_g m_{gk} x_g(p_0 + p_k + \Delta p_{gk}) \quad (1)$$

where G represents the total number of aggregation groups. For the g th aggregation group, $w_g \in R^{C \times C'}$ represents the projection weight without any information about the location, where $C' = C/G$ is the dimension of the group; $m_{gk} \in R$ is the modulation scalar for the k th sampling point in the g th aggregation group, which is normalized by the SoftMax function along k . $x_g \in R^{C' \times H \times W}$ represents the sliced input feature map and Δp_{gk} represents the offset corresponding to the grid sampling location p_k in the g th group.

2.3. Dysample

Dysample is primarily utilized in the upsampling module of the neck network. While YOLOv8 adopts nearest-neighbor sampling as its sampling strategy, this method's output image quality remains a concern despite its advantage of fast computation. To address the above issue, scholars have proposed operators such as CARAFE, FADE, and SAPA to enhance output image quality; however, the use of these operators sacrifices efficiency and

increases application thresholds. In contrast, Dysample is implemented solely through PyTorch's built-in functions with lower computational resource consumption and higher accuracy improvement than the aforementioned operators. Its working principle involves adding compensation values to sampling points during upsampling operations while determining compensation values via bilinear interpolation for dynamic sampling.

The process of feature map processing by Dysample is shown in Figure 2:

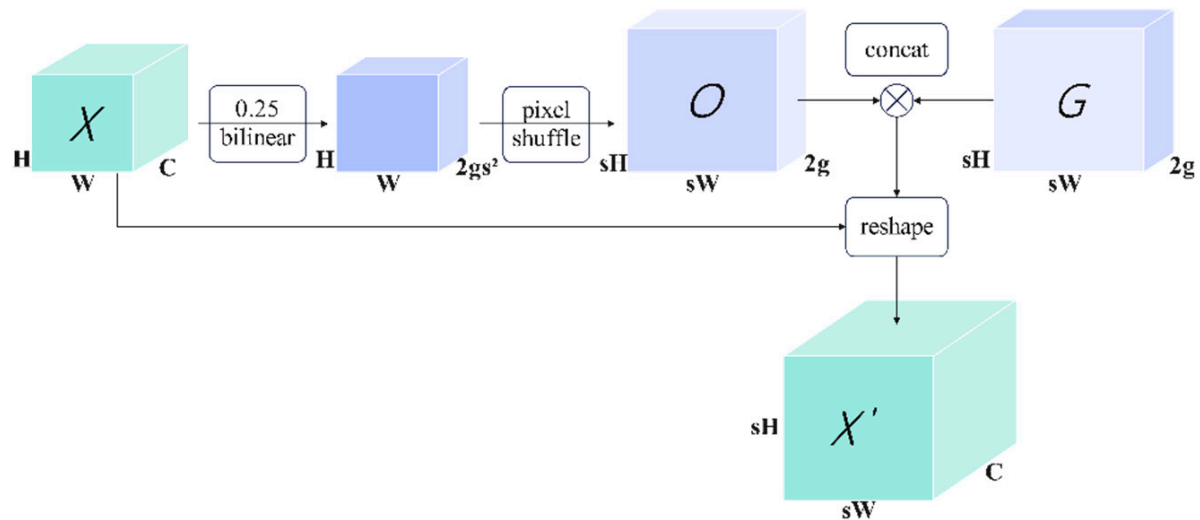


Figure 2. Flowchart of the upsampling principle in Dysample. By employing specialized sampling techniques, an additional compensation network O is derived, which is then combined with the original sampling network G to yield the dynamic sampling outcome.

The process of the flowchart can be explained using the following formula:

$$x' = \text{reshaping_sample}(x, g + \text{pixel_shuffle}(0.25\text{bilinear}(x))) \quad (2)$$

In this equation, x represents the input feature map, x' represents the output feature map, and g is the original sampling grid. x is sampled via 0.25-scaled bilinear interpolation and then pixel rearranged to obtain compensation values, which are added to the original sampling network and then reshaped to perform a dynamic upsampling.

2.4. Efficienthead

In YOLOv8, the computation of the detection head constitutes approximately 40% of the entire algorithm. Considering the potential future application of the wear particle recognition algorithm in mobile or online devices, it is imperative to minimize reliance on device computing power. Upon examining the detection head of YOLOv8, it becomes evident that it comprises two branches, each utilizing two 3×3 convolutions and one 1×1 2d convolution for information extraction. These branches are employed to calculate the bounding box loss function and classification loss function, respectively. Furthermore, traversing each channel necessitates substantial computational overhead. As shown in Figure 3. To mitigate this computational burden, we propose a shared parameter method in the present study, which simplifies three convolution kernels in both branches into one kernel within a single branch while retaining only two 3×3 convolution kernels for calculating the two loss functions. This approach not only reduces computational complexity but also demonstrates improved precision through experimental validation.



Figure 3. Diagram showing the improvement of the detection head before and after the aforementioned method is used. The original two branches are consolidated into a single branch, with parameter sharing for loss calculation and simplified convolutional kernels, therefore reducing the computational burden of the detection head.

2.5. WISEPiou

Boundary box regression (BBR) is one of the two crucial tasks in object detection, effectively guiding the algorithm to focus on the target task. In YOLOv8, the boundary box loss comprises intersection-over-union (IoU) loss and distribution focal loss, with the present study primarily emphasizing improvements in IoU loss. YOLOv8 employs the CIOU loss function, represented by the following formula:

$$L_{\text{CIOU}} = L_{\text{IoU}} + \frac{d^2}{c^2} + \alpha^* \nu, \quad 0 \leq L_{\text{CIOU}} \leq 2 \quad (3)$$

$$\alpha = \frac{\nu}{1 - \text{IoU} + \nu}, \quad \nu = \frac{4}{\pi^2} \left(\arctan \frac{W_{\text{gt}}}{h_{\text{gt}}} - \arctan \frac{W}{h} \right)^2 \quad (4)$$

Equation (3) consists of three parts, which, respectively, consider the overlapping area, center distance, and aspect ratio of the bounding box regression. In this formula, d represents the distance between the predicted box center and the true box center, c represents the distance of the diagonal of the smallest enclosing rectangle, α is a weight coefficient, and ν is a correction factor used to measure the consistency between the predicted box and the true box shape.

In the present study, we replaced the Ciou with WISEPiou, which is a combination of WISE-IoU and POWERFUL-IoU, and the loss functions of WISE-IoU and POWERFUL-IoU are given by Equations (5) and (6), respectively:

$$L_{\text{WIoU}} = R_{\text{WIoU}} L_{\text{IoU}} \quad (5)$$

$$R_{\text{WIoU}} = \exp\left(\frac{d^2}{c^2}\right) \quad (6)$$

Equation (5) multiplies the loss function by a penalty factor R_{WIoU} , which can promptly correct the deviation from the original loss function.

$$L_{\text{PIoU}} = L_{\text{IoU}} + R_{\text{PIoU}} \quad (7)$$

$$R_{\text{PIoU}} = 1 - \exp\left(-\left(\frac{dw_1}{w_{\text{gt}}} + \frac{dw_2}{w_{\text{gt}}} + \frac{dh_1}{h_{\text{gt}}} + \frac{dh_2}{h_{\text{gt}}}\right)/4\right)^2 \quad (8)$$

Equation (7) adds a penalty factor R_{PIoU} to the loss function, as shown in Equation (8). In this formula, dw_1 , dw_2 , dh_1 , and dh_2 represent the corresponding edge distances between the predicted box and the target box, and the denominators are the width and height of the target box.

Therefore, combining the characteristics of the two IoU loss functions, the WISEPiou loss function formula is given by Equation (9):

$$L_{\text{WisePiou}} = R_{\text{WPIoU}} L_{\text{IoU}} + R_{\text{PIoU}} \quad (9)$$

Based on the above improvements, the model structure used in the present study is shown in Figure 4.

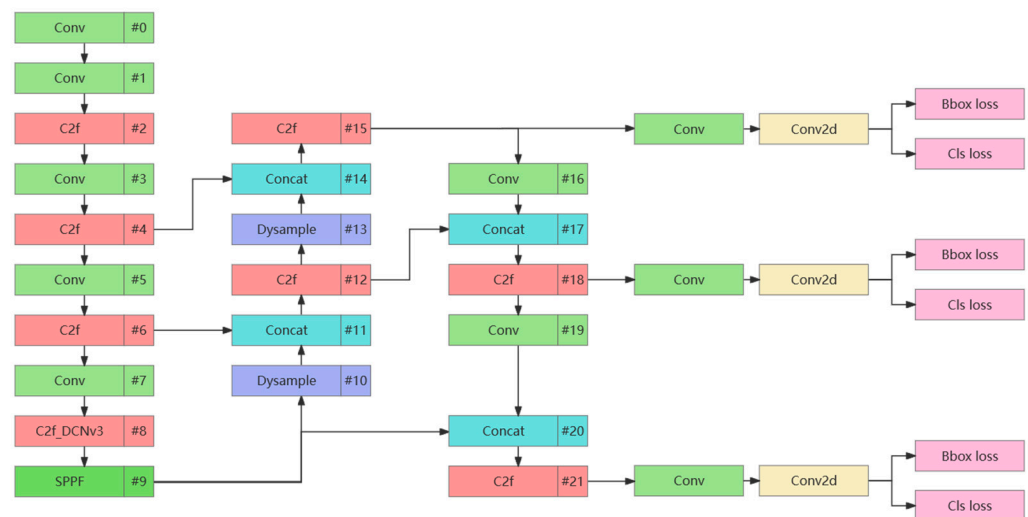


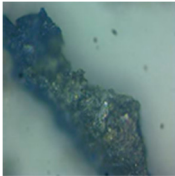
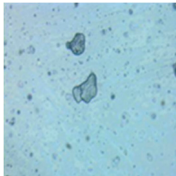
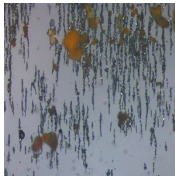
Figure 4. Structural diagram of the improved YOLOv8 model. Revisions to the original network include the integration of DCNv3 and C2f layer in #8, the implementation of Dysample in #10 and #13, and the streamlining of the detection head.

3. Experiment

3.1. Introduction to the Dataset and Experimental Equipment

The original images in the utilized dataset were selected from the friction and wear experiment library of the Institute of Safety and Energy Saving of Marine Power Plant at Shanghai Maritime University. The abrasive particles were generated using a universal mechanical performance testing machine by Bruker (Billerica, MA, USA), which was used to perform reciprocating sliding experiments (according to the ASTM G203-10 standard [34]), disc experiments (according to the ASTM G99-2005 standard [35]), and four-ball experiments (according to the ASTM D5183-21 standard [36]). The spectrograms were prepared using a SPECTRO-T2FM500 (Spectro Scientific, Chelmsford, MA, USA) ferrography analysis instrument, and the original images were captured using a CCD camera-equipped Olympus BX51 optical microscope (Olympus, Tokyo, Japan). High-resolution original images were cropped to obtain 640 images from the dataset with dimensions of 300×300 , and the images were labeled and organized into a YOLO-format abrasive particle dataset. Each category in the dataset was divided into a training set and a validation set in an 8:2 ratio, with the training set used for model training and evaluation of the validation set, ensuring consistent distribution across categories. The primary aim was to improve the recognition rate of similar wear particles in the dataset without external data augmentation. The distribution of wear particles in the dataset is shown in Table 1.

Table 1. The categories and classification of the wear particle dataset. The chunky surface has a noticeable three-dimensional texture, and the camera is only able to focus on part of it. The laminar surface has a smooth surface or small holes and cracks. The nonferrous surface comprises different colors, and the sliding surface contains scratches.

Wear debris	Chunky	Laminar	Nonferrous	Sliding
Image				
Training set	128	128	128	128
Test set	32	32	32	32

In the present study, Python v3.8.19 and PyTorch v1.13.1+cu117 were used to build a deep learning framework on the Ubuntu 20.0.6 LTS operating system. The relevant parameters of the experimental platform are listed in Table 2, and YOLOv8n version 8.1.9 was selected as the benchmark model. The model hyperparameters are listed in Table 3.

Table 2. Experimental platform hardware.

Hardware	Parameters
GPU	NVIDIA GeForce GTX 4060Ti
CPU	Intel i5-13400
CUDA	11.7
CuDNN	8.8.0

Table 3. Hyperparameters of YOLOv8.

Hyperparameters	Value
Momentum factor	0.9125
Initial learning rate	0.01
Final learning rate	0.01
Optimizer	AdamW
Weight attenuation	0.0005
Random number seed	128
Epochs	600

3.2. Evaluation Indicators

To verify the effectiveness and feasibility of the detection model in the experiment presented herein, we evaluated these factors from both qualitative and quantitative perspectives. In the qualitative evaluation, we assessed the model's performance by comparing the detection images between the proposed model and the control group model, including the precision of the target box location and the situation of false negatives and false positives. In the quantitative evaluation, we mainly chose the following indicators: precision (P), recall (R), average precision (AP), and mean average precision (mAP). The corresponding formulas are as follows:

$$P = \frac{T_P}{T_P + F_P} \times 100\% \quad (10)$$

$$R = \frac{T_P}{T_P + F_N} \times 100\% \quad (11)$$

$$mAP = \frac{1}{m} \sum_{j=1}^m \int_0^1 P(R) dR \quad (12)$$

$$FPS = \frac{1}{t_s} \quad (13)$$

In Equations (10)–(13), T_P represents the number of objects detected correctly by the detection model, F_P represents the number of misidentifications or missed detections, and F_N represents the number of times a target wear particle is misclassified as another category. P and R represent the accuracy rate and recall rate, respectively, and form the PR curve function. AP is the average score of the accuracy rate against the recall value, which is often used to evaluate model performance. mAP is the average of AP for m categories, which can be used to comprehensively evaluate the overall performance of the detection model; in comparison, mAP@0.5 refers to the average precision calculated at an IoU threshold of 0.5. FPS stands for the number of images processed per unit of time, which includes the time required for preprocessing, inference, and post-processing.

4. Results Analysis of Each Improvement Part

4.1. Comparative Analysis of DCNv3

In the following study, we fused the DCNv3 module with the C2f module in the YOLOv8 backbone network, where the C2f module is located at the second, fourth, sixth, and fourth layers. Therefore, we replaced the fused C2f-DCNv3 module at the corresponding positions and tested its impact on model performance.

The experimental results, as shown in Table 4, demonstrate the effectiveness of replacing the four-layer C2f module in Model 1. Compared to the initial model, there was a 5.1% increase in accuracy and a 2.7% improvement in average precision. Subsequently, we gradually reduced the number of replaced layers and tested the performance of the C2f-DCNv3 module at different positions. From the table, it can be seen that although Model 2 achieved higher accuracy, its recall rate significantly decreased, further impacting the average precision metric. However, in Model 4, when only replacing the eighth layer's C2f-DCNv3 module, mAP@0.5 reached its optimal value of 74.3% while maintaining a recall rate of 65.8%. The above results indicate that distributing the fused C2f-DCNv3 module deeper within the backbone network allows for stronger feature extraction capabilities. Although overall replacement also yields good results, it introduces an additional computational burden. Therefore, we adopted an improved method by solely replacing the eighth layer's C2f module.

Table 4. Experimental comparison of the primary backbone network in C2f-DCNv3. The original C2f layer is substituted with a C2f-DCNv3 layer, denoted by a $\checkmark$ to indicate the corresponding replacement position. The specific corresponding layer can be found in Figure 1.

Model	#2	#4	#6	#8	P	R	mAP@0.5
0					65.8	68.0	70.8
1	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	70.9	67.5	73.5
2		$\checkmark$	$\checkmark$	$\checkmark$	73.1	63.1	71.2
3			$\checkmark$	$\checkmark$	69.3	68.0	73.7
4				$\checkmark$	73.6	65.8	74.3

4.2. Comparative Analysis of Dysample

Dysample is a modification to the neck network, replacing the nearest-neighbor upsampling technique in the original network with a parameter setting of 2 and a mode of lp. To validate the performance of this module, comparative experiments were designed for the neck network using BIFPN, GDFPN, and GFPN. The experimental results are shown in Table 5.

Table 5. Neck network horizontal comparison. For the improvement of the neck network, there are four different fusion strategies for the BIFPN module; the other modules, however, do not use these strategies. $\times$ means there are no different modes for this module.

Neck	Mode	P	R	mAP@0.5
BIFPN	Default	70.1	68.2	73.8
	Weight	67.6	67.0	71.5
	Adaptive	72.0	65.1	72.5
	Concat	70.3	68.8	73.5
GDFPN	$\times$	70.4	67.0	73.5
GFPN	$\times$	66.9	68.3	73.0
DYSAMPLE	$\times$	73.7	66.2	74.0

As shown in Table 5, Dysample achieves the highest mAP of 74% and an accuracy rate of 73.7%. The recall rate remains at a respectable 66.2%. Compared to other neck networks, Dysample demonstrates significant advantages, indicating its effective and accurate feature map restoration during upsampling. BIFPN has four different modes, with the default

mode achieving an mAP of 73.8%. Similarly, GDFPN and GFPN both exceed 73% but struggle to balance accuracy and recall; improving one metric often comes at the expense of the other. Only Dysample shows a relatively insignificant decrease in recall while achieving the highest accuracy rate. Taking all these factors into consideration, it can be concluded that Dysample is the best choice for improving neck networks.

4.3. Comparative Analysis of Efficienthead

To improve the header networks, we selected three comparable networks as controls, namely SEAMHEAD, DYHEAD, and DYHEAD fused with DCNv3. The experimental results are shown in Table 6.

Table 6. Head network horizontal comparison. DYHEAD is a detection head that incorporates DCNv2; DCNv3-HYDEAD, in comparison, is a detection head that incorporates DCNv3.

HEAD	P	R	mAP@0.5
SEAMHEAD	71.8	63.5	72.9
DYHEAD	69.4	66.5	72.9
DCNv3-DYHEAD	73.3	64.9	74.0
EFFICIENTHEAD	72.3	68.1	74.6

Although Efficienthead was designed to reduce the computational burden on the head, it is evident that sharing parameters has a positive impact on both the accuracy and recall of the model. Compared to the improved DYHEAD, Efficienthead performs slightly worse in terms of accuracy but better than the other models; however, its performance in terms of recall is more significant, exceeding the other models' average level by 3%. Therefore, it achieves an excellent mAP improvement of 74.6% among similar improvements. Additionally, due to improvements in backbone and neck networks leading to a decrease in recall rate, Efficienthead can balance the effects brought by these two factors and subsequently provide a detailed comparative analysis.

5. Comprehensive Results Analysis

5.1. Ablation Experiment

To better verify the impact of the improved modules and the combination of improved modules on the original model's performance, an ablation experiment was conducted. The results are presented in Table 7.

Table 7. Results of the ablation experiment. × means that this module is not used, and √ means that this module is used.

Dysample	DCNv3	Efficienthead	WISEPiou	P	R	mAP@0.5	FPS	FLOPs
×	×	×	×	65.8	68.0	70.8	133.7	8.9
√	×	×	×	73.7	66.2	74.0	104.4	8.1
×	√	×	×	73.6	65.8	74.3	116.7	8.0
×	×	√	×	72.3	68.1	74.6	132.1	8.1
√	×	√	×	69.0	71.4	74.2	112.6	8.1
√	√	√	×	71.8	70.9	75.9	110.5	8.0
√	√	√	√	74.2	68.2	76.4	111.6	8.0

According to the experimental results presented in Table 7, both individual and combined improvements of each module showed specific enhancement effects on the original model. When Dysample, DCNv3, and Efficienthead are used separately, they all contribute to a 3% performance improvement in the original model. However, their use leads to a decrease in FPS due to increased inference time. In terms of floating-point operations, these three modules were implemented based on simple underlying principles while maintaining comparable computational complexity with the original model.

By combining Dysample and Efficienthead, the recall rate significantly improves and effectively mitigates recall rate decline caused by other module enhancements in subsequent ablation experiments. After incorporating DCNv3 and WISEPIou, the improved model achieves optimal accuracy rates of 74.2% and mAP76.4%, respectively. Although loss function improvements may result in a slight decrease in recall rate compared to the original model level, the improved model still outperforms the original model overall.

The decrease in FPS is not severe as it maintains real-time detection requirements at 111.6 FPS. Furthermore, a reduction of 0.9 in floating-point operations compared to the original model was determined, which further alleviates reliance on computational power.

5.2. Comparative Experiments

As shown in Table 8. In terms of accuracy, recall, and mAP@0.5, the enhanced model demonstrates superior performance, exhibiting a 2.6% increase in accuracy over the second-best improved YOLOv5n, a 1.1% improvement in recall compared to the second-best YOLOv9-S, and a 3.8% enhancement in mAP@0.5 relative to the second-best YOLOv9-S. Furthermore, the enhanced model achieves an FPS of 111.6, meeting online monitoring requirements with an 18.4% advantage over YOLOX-Tiny within the same series and comparable performance to other models in the experiment, notably outperforming models with an FPS below 30. In terms of FLOPs, although not minimal at 8.0 compared to YOLOX's rate of 7.578, it remains only marginally higher by 0.5G, presenting a significant advantage when contrasted with non-YOLO models requiring over 100G FLOPs.

Table 8. Experimental results of the comparison of the different algorithms.

Algorithms	P	R	mAP@0.5	FPS	FLOPs
Faster-RCNN [37]	64.0	56.1	66.1	28.7	208
TOOD [38]	59.8	55.5	67.1	26.3	199
YOLOX-Tiny [39]	62.5	55.5	69.1	93.2	7.578
ATSS-Dyhead [40]	64.4	46.1	65.8	16.3	110
DINO [41]	68.9	51.0	69.6	15.1	274
Improved YOLOv5n	71.6	65.6	72.2	112.3	7.1
YOLOv9-S	68.8	67.1	72.6	108.7	38.7
Our algorithm	74.2	68.2	76.4	111.6	8.0

Through comparative experiments, it is possible to verify that this improved model achieves optimal detection accuracy with a relatively small size and meets high frame-rate video detection requirements.

5.3. Visualization Chart

To better demonstrate the advantages of our model in the present study, we selected three types of images: complex-shaped wear particles, overlapping wear particles, and edge target wear particles. We used the GradCAMPlusPlus visualization technique for heatmap visualization analysis. Figure 5 illustrates a heatmap comparison among the algorithms, where the first row depicts the original image, the second row presents the enhanced YOLOv5 heatmap, the third row displays the YOLOv8 heatmap, the fourth row showcases the YOLOv9 heatmap, and finally, in the fifth row is depicted an improved version of YOLOv8 heatmap. In each row's first image lies an atypical severe sliding abrasive grain image with obvious scratches on the abrasive grain but not on the texture-rich grain in the image—a complex scenario. The second image portrays two overlapping abrasive grains to validate the model's capability to handle overlapping relationships. The third image showcases distributed abrasive grains of varying sizes and positions to assess the model's perception ability from both scale and positional perspectives.

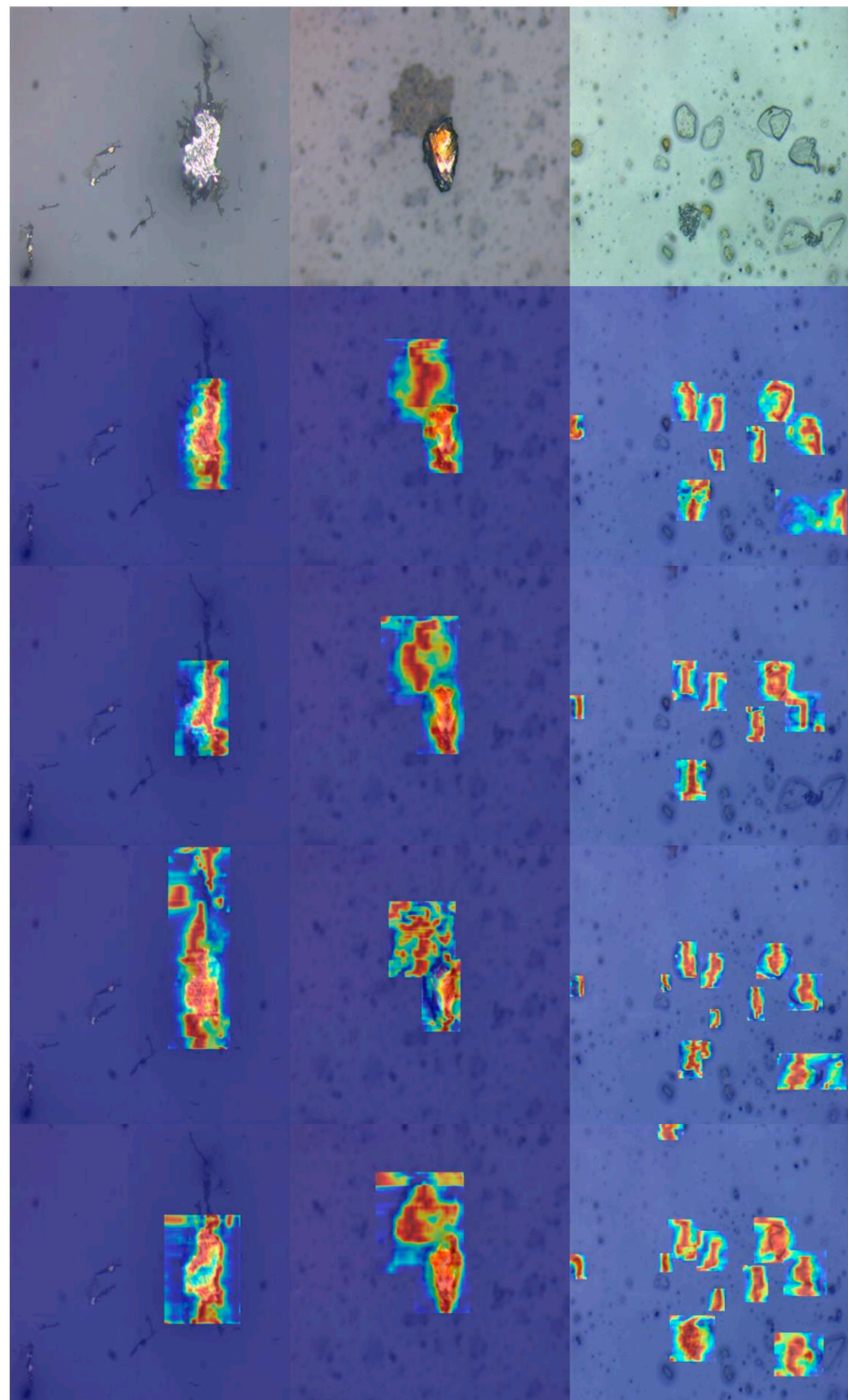


Figure 5. A comparison of heatmaps generated by different algorithms. The five rows, from top to bottom, correspond to the original image, the enhanced YOLOv5 (featuring the v8 head network), YOLOv8, YOLOv9, and the enhanced YOLOv8.

It can be inferred from the heatmap of the first column that the other models exhibit either a conservative or radical interpretation of the edge of the abrasive grain profile, potentially leading to gradual error accumulation in feature extraction and a subsequent decline in detection capability. In contrast, the high heatmap value (red area) generated by

our proposed model accurately characterizes the abrasive grain profile without significant overshooting, effectively disregarding atypical textures of complex abrasive grains to mitigate potential model fragility resulting from excessive focus on special cases. This aspect is primarily attributed to dynamic downsampling and upsampling, which enhance feature selection rather than incorporating all features into calculations.

Observing the heatmap in the second column, it can be seen that the original model cannot accurately perceive the overlapping area when two worn particles overlap, which affects the extraction of nearby features. It can be observed that in positions close to the overlapping area, the original model presents a relatively dim heatmap. In contrast, our proposed improved model effectively suppresses this trend and provides more comprehensive and accurate perception results. Even in the presence of overlapping areas, it is not significantly affected.

The conclusion drawn from the heatmaps displayed in the third column is that the original model can only perceive worn particles located in the center of images with high integrity; however, it does not perform well in perceiving worn particles located at edges or with fewer quantities. In contrast, our improved model adopts the DCNv3 module and has a wide dynamic receptive field. The model can detect more cases of worn particles existing at edges and possesses richer and more accurate perception ability for small-sized worn particles near image edges. Therefore, this improvement effectively enhances detection accuracy and recall rate.

6. Conclusions

In the following paper, an improved YOLOv8 model is proposed based on dynamic point sampling to address the challenges of complex shapes, overlapping, and edge target wear particles in the intelligent classification and online detection of spectrogram wear particles. Through training and validation on a series of original wear particle images, the following conclusions can be drawn:

(1) The improved YOLOv8 model effectively mitigates detection difficulties such as complex shapewear particles, overlapping wear particles, and edge target wear particles based on the principle of dynamic point sampling, even with a limited dataset and no additional data augmentation. These features reduce the occurrence of undetected objects, missed detections, and false alarms. The lightweight module usage and improved detection head reduce the model's floating-point computations while simultaneously optimizing the loss function to accelerate model convergence speed and improve detection box accuracy.

(2) Through repeated experiments on a limited dataset, the text model achieves an accuracy rate of 76.4% in terms of the mAP@0.5 metric and demonstrates high-speed detection capabilities at 111.6 FPS, surpassing recently developed algorithms such as TOOD, YOLOX-Tiny, and DINO.

(3) The current algorithm achieves real-time detection speeds above 30 FPS but is limited by dataset size and quality. After training on higher-quality datasets, its accuracy will further improve, with significant application prospects in fields such as aviation engine diagnostics [42], industrial product defect detection [43], and medical imaging [44].

Author Contributions: Conceptualization, J.W. and H.W.; methodology, J.W.; software, J.W.; validation, J.W. and H.W.; formal analysis, J.W.; investigation, J.W.; resources, J.W.; data curation, J.W.; writing—original draft preparation, J.W. and Z.C.; writing—review and editing, J.W. and D.Z.; visualization, J.W.; supervision, J.W.; project administration, J.W.; funding acquisition, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Shanghai Engineering Research Center of Intelligent Ship Operation and Energy Efficiency Monitoring, Shanghai Science and Technology Program, grant number 20DZ2252300.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SVDD	Support Vector Data Description
SVM	Support Vector Machine
DCNNs	Deep Convolutional Neural Networks
BP	Back Propagation
YOLO	You Only Look Once
ViT	Vision Transformer
DCNs	Deformable Convolutional Networks

References

- Jardine, A.K.S.; Lin, D.; Banjevic, D. A review on machinery diagnostics and Prognostics Implementing Condition-Based Maintenance. *Mech. Syst. Signal Process.* **2006**, *20*, 1483–1510. [\[CrossRef\]](#)
- Han, W.; Mu, X.; Liu, Y.; Wang, X.; Li, W.; Bai, C.; Zhang, H. A Critical Review of On-Line Oil Wear Debris Particle Detection Sensors. *J. Mar. Sci. Eng.* **2023**, *11*, 2363. [\[CrossRef\]](#)
- Kumar, M.; Mukherjee, P.S.; Misra, N.M. Advancement and Current Status of Wear Debris Analysis for Machine Condition Monitoring: A Review. *Ind. Lubr. Tribol.* **2013**, *65*, 3–11. [\[CrossRef\]](#)
- Raadnui, S. Wear Particle Analysis—Utilization of Quantitative Computer Image Analysis: A Review. *Tribol. Int.* **2005**, *38*, 871–878. [\[CrossRef\]](#)
- Ebersbach, S.; Peng, Z.; Kessissoglou, N.J. The Investigation of the Condition and Faults of a Spur Gearbox Using Vibration and Wear Debris Analysis Techniques. *Wear* **2006**, *260*, 16–24. [\[CrossRef\]](#)
- Roylance, B.J. Ferrography—Then and Now. *Tribol. Int.* **2005**, *38*, 857–862. [\[CrossRef\]](#)
- Hasan, M.S.; Nosonovsky, M. Triboinformatics: Machine Learning Algorithms and Data Topology Methods for Tribology. *Surf. Innov.* **2022**, *10*, 229–242. [\[CrossRef\]](#)
- Paturi, U.M.R.; Palakurthy, S.T.; Reddy, N.S. The Role of Machine Learning in Tribology: A Systematic Review. *Arch. Comput. Methods Eng.* **2022**, *30*, 1345–1397. [\[CrossRef\]](#)
- Cao, W.; Dong, G.; Xie, Y.-B.; Peng, Z. Prediction of Wear Trend of Engines Via On-Line Wear Debris Monitoring. *Tribol. Int.* **2018**, *120*, 510–519. [\[CrossRef\]](#)
- Wang, J.Q.; Yao, P.P.; Liu, W.L.; Wang, X.L. A Hybrid Method for the Segmentation of a Ferrograph Image Using Marker-Controlled Watershed and Grey Clustering. *Tribol. Trans.* **2016**, *59*, 513–521. [\[CrossRef\]](#)
- Peng, Y.P.; Wu, T.H.; Cao, G.Z.; Huang, S.D.; Wu, H.K.; Kwok, N.; Peng, Z.X. A Hybrid Search-Tree Discriminant Technique for Multivariate Wear Debris Classification. *Wear* **2017**, *392*, 152–158. [\[CrossRef\]](#)
- Peng, P.; Wang, J. FECNN: A Promising Model for Wear Particle Recognition. *Wear* **2019**, *432–433*, 202968. [\[CrossRef\]](#)
- Peng, Y.; Cai, J.; Wu, T.; Cao, G.; Kwok, N.; Zhou, S.; Peng, Z. Online Wear Characterisation of Rolling Element Bearing Using Wear Particle Morphological Features. *Wear* **2019**, *430*, 369–375. [\[CrossRef\]](#)
- Wang, S.; Wu, T.H.; Yang, L.F.; Kwok, N.; Sarkodie-Gyan, T. Three-Dimensional Reconstruction of Wear Particle Surface Based on Photometric Stereo. *Measurement* **2019**, *133*, 350–360. [\[CrossRef\]](#)
- Wang, S.; Wu, T.H.; Shao, T.; Peng, Z.X. Integrated Model of BP Neural Network and CNN Algorithm for Automatic Wear Debris Classification. *Wear* **2019**, *426*, 1761–1770. [\[CrossRef\]](#)
- Wang, S.; Wu, T.; Zheng, P.; Kwok, N. Optimized CNN Model for Identifying Similar 3D Wear Particles in Few Samples. *Wear* **2020**, *460–461*, 203477. [\[CrossRef\]](#)
- Wang, S.; Wu, T.; Wang, K.; Sarkodie-Gyan, T. Ferrograph Analysis with Improved Particle Segmentation and Classification Methods. *J. Comput. Inf. Sci. Eng.* **2020**, *20*, 021001. [\[CrossRef\]](#)
- Peng, Y.P.; Cai, J.H.; Wu, T.H.; Cao, G.Z.; Kwok, N.; Peng, Z.X. WP-DRnet: A Novel Wear Particle Detection and Recognition Network for Automatic Ferrograph Image Analysis. *Tribol. Int.* **2020**, *151*, 9. [\[CrossRef\]](#)
- Fan, S.L.; Zhang, T.H.; Guo, X.X.; Wulamu, A. FFWR-Net: A Feature Fusion Wear Particle Recognition Network for Wear Particle Classification. *J. Mech. Sci. Technol.* **2021**, *35*, 1699–1710. [\[CrossRef\]](#)
- Vivek, J.; Venkatesh, N.S.; Mahanta, T.K.; Sugumaran, A.; Amarnath, M.; Ramteke, S.M.; Marian, M. Wear Particle Image Analysis: Feature Extraction, Selection and Classification by Deep and Machine Learning. *Ind. Lubr. Tribol.* **2024**, *76*, 599–607. [\[CrossRef\]](#)
- Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [\[CrossRef\]](#)
- Jia, F.G.; Wei, H.J.; Sun, H.Y.; Song, L.; Yu, F.L. An Object Detection Network for Wear Debris Recognition in Ferrography Images. *J. Braz. Soc. Mech. Sci. Eng.* **2022**, *44*, 67. [\[CrossRef\]](#)
- He, L.; Wei, H.; Gao, W. Research on an Intelligent Classification Algorithm of Ferrography Wear Particles Based on Integrated ResNet50 and SepViT. *Lubricants* **2023**, *11*, 530. [\[CrossRef\]](#)
- Fan, H.W.; Gao, S.Q.; Liu, Q.; Ma, N.G.; Zhang, X.H.; Cao, X.G. Intelligent Wear Debris Identification of Gearbox Based on Virtual Ferrographic Images and Two-Level Transfer Learning. *Int. J. Pattern Recogn.* **2022**, *36*, 2251012. [\[CrossRef\]](#)

25. Shi, X.F.; Cui, C.; He, S.Z.; Xie, X.P.; Sun, Y.H.; Qin, C.D. Research on Recognition Method of Wear Debris Based on YOLO V5S Network. *Ind. Lubr. Tribol.* **2022**, *74*, 488–497. [[CrossRef](#)]
26. Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; et al. InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions. *arXiv* **2023**, arXiv:2211.05778.
27. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to Upsample by Learning to Sample. *arXiv* **2023**, arXiv:2308.15085.
28. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IOU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
29. Liu, C.; Wang, K.; Li, Q.; Zhao, F.; Zhao, K.; Ma, H. Powerful-IOU: More Straightforward and Faster Bounding Box Regression Loss with a Nonmonotonic Focusing Mechanism. *Neural Netw.* **2024**, *170*, 276–284. [[CrossRef](#)]
30. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
31. Bochkovskiy, A.; Wang, C.-Y.; Mark Liao, H.-Y. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
32. He, L.; Wei, H.; Wang, Q. A New Target Detection Method of Ferrography Wear Particle Images Based on ECAM-YOLOv5-BiFPN Network. *Sensors* **2023**, *23*, 6477. [[CrossRef](#)] [[PubMed](#)]
33. Wang, C.-Y.; Bochkovskiy, A.; Mark Liao, H.-Y. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. *arXiv* **2022**, arXiv:2207.02696.
34. ASTM G203-10; Standard Guide for Determining Friction Energy Dissipation in Reciprocating Tribosystems. ASTM: West Conshohocken, PA, USA, 2020.
35. ASTM G99-2005; Standard Test Method for Wear Testing with a Pin-on-Disk Apparatus. ASTM: West Conshohocken, PA, USA, 2010.
36. ASTM D5183-21; Standard Test Method for Determination of the Coefficient of Friction of Lubricants Using the Four-Ball Wear Test Machine. ASTM: West Conshohocken, PA, USA, 2022.
37. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2016**, arXiv:1506.01497. [[CrossRef](#)]
38. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. TOOD: Task-Aligned One-stage Object Detection. 2021. Available online: <https://ui.adsabs.harvard.edu/abs/2021arXiv210807755F> (accessed on 10 April 2024).
39. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**, arXiv:2107.08430.
40. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; Zhang, L.J. Dynamic Head: Unifying Object Detection Heads with Attentions. 2021. Available online: <https://ui.adsabs.harvard.edu/abs/2021arXiv210608322D> (accessed on 10 April 2024).
41. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.-Y. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. *arXiv* **2022**, arXiv:2203.03605.
42. Wang, X.; Hong, W.; Liu, Y.; Hu, D.; Xin, P. SAR Image Aircraft Target Recognition Based on Improved YOLOv5. *Appl. Sci.* **2023**, *13*, 6160. [[CrossRef](#)]
43. Lv, H.; Zhang, H.; Wang, M.; Xu, J.; Li, X.; Liu, C. Hyperspectral Imaging Based Nonwoven Fabric Defect Detection Method Using LL-YOLOv5. *IEEE Access* **2024**, *12*, 41988–41998. [[CrossRef](#)]
44. Zhou, T.; Liu, F.; Ye, X.; Wang, H.; Lu, H. CCGL-YOLOV5: A Cross-Modal Cross-Scale Global-Local Attention YOLOV5 Lung Tumor Detection Model *. *Comput. Biol. Med.* **2023**, *165*, 107387. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.