Class Imbalance Alleviation In Object Detection Via YOLOv11based Deep Dynamic Feature Fusion

Qi Yang[1], Bingkun Jiang[1], Jiatong Tang[1], Jianxi Huang [2], and Minghao Li[1*]

[1] *Shenyang Ligong University*
[2] *Fuzhou University*
[*] *Corresponding author. E-mail: lmh0315@126.com*

Received: Dec. 31, 2025; Accepted: Jan. 12, 2025

Object detection models often face significant performance limitations. These challenges include severe sample imbalance, background interference, and target occlusion. Such issues are particularly prevalent in complex industrial and medical imaging domains. While existing solutions typically focus on data resampling or loss function re-weighting to handle imbalance, a fundamental bottleneck within the network architecture itself is often overlooked. Traditional feature fusion necks, such as FPN and PANet, rely on static convolutions that inevitably become biased towards the majority class during training, leading to the marginalization or loss of minority-class features. To address this critical issue at the feature-fusion level, we propose the Semantic-Aware Fusion Neck (SAF-Neck), which replaces the static fusion paradigm with a dynamic, input-adaptive mechanism. By generating content-aware convolutional kernels for each input, SAF-Neck adaptively enhances the discriminative features of minority-class samples, preventing them from being suppressed by the majority class. We integrate this core innovation into a synergistic architecture with a Lightweight Probabilistic Spatial Attention-HGNetv2(LPSA-HGNetv2) and an imbalance-robust loss function , forming a comprehensive "front-end feature enhancement and back-end optimization" pipeline. We validate our model, SAF-YOLOv11, on a highly challenging industrial task of coal and gangue classification, characterized by a severe class imbalance ratio of up to 1 : 22. Experimental results show that our model achieves a 90.4% F1-score with a computational load of only 5.7 GFLOPs, outperforming the baseline by 4.1% in F1-score while being 13.6% more computationally efficient.

Keywords: Coal And Gangue Detection;Semantic-Aware Fusion;Lightweight Network;Class Imbalance;YOLOv11n

http://dx.doi.org/10.6180/jase.202608_31.019

1. Introduction

In recent years, deep learning, particularly Convolutional Neural Networks (CNNs), has driven significant progress in object detection, with applications spanning from medical diagnosis [1] to industrial manufacturing [2–4]. However, when deploying these advanced algorithms in complex, real-world industrial settings [5–7], their performance is often limited by three fundamental challenges: sample imbalance, background interference, and target occlusion. First, many industrial datasets exhibit a severe long-tail distribution where critical minority-class samples are vastly outnumbered, leading to a strong model bias towards the majority class. Second, a critical trade-off between accuracy and efficiency remains: highprecision models are often too computationally expensive for real-time deployment on edge devices, while lightweight models frequently sacrifice necessary feature representation capabilities, creating a persistent "accuracy-efficiency" bottleneck. Finally, industrial scenes are often cluttered with densely packed and overlapping objects, causing model attention mechanisms to fail to focus on salient features, which results in inac-

curate localization and misclassification. The automated sorting of coal and gangue is a quintessential industrial use case where all the aforementioned challenges manifest acutely [8–11].More recently, studies in 2025 have released large-scale datasets and explored advanced architectures specifically for complex industrial scenarios, such as intelligent raw coal sorting and infrastructure defect inspection [12, 13]. As a promising alternative to traditional labor-intensive methods, machine vision-based sorting systems demand exceptional real-time performance and accuracy from the underlying detection model. In real-world data from this task, the nature of coal production results in an inherent and severe class imbalance, with an observed ratio of gangue (majority class) to coal (minority class) as high as 22 : 1. This is a reflection of typical operational conditions, not a data collection artifact. While numerous studies have applied deep learning techniques to this problem, proposing models like RDB-YOLOv8n [14] and CSPNet-YOLOv7 [15] to handle these issues, we observe that they largely overlook a fundamental bottleneck at the core of the network architecture. To overcome these issues, our contributions are threefold:

- Lightweight Backbone: We design LPSA-HGNetv2, an improved lightweight backbone network that provides a high-quality feature foundation while significantly reducing computational overhead.
- Semantic-Aware Fusion Neck: We propose the SAF-Neck to address the static fusion bottleneck. By replacing static convolutions with a dynamic, input-adaptive routing mechanism, this core innovation generates specialized feature extraction strategies, preventing minorityclass features from being marginalized.
- Imbalance-Robust Optimization: We incorporate the SlideLoss function to specifically address class imbalance from the optimization perspective, further enhancing model robustness by focusing on hard-to-classify samples.

Through extensive experimentation, we demonstrate that SAF-YOLOv11 achieves a superior balance of accuracy and speed, outperforming the baseline YOLOv11n [16–18] and other state-of-the-art models, thus providing a robust and practical solution for automated coal and gangue sorting.

2. Improvement plan

To address the challenges of low target-background contrast, severe class imbalance, and object occlusion in industrial sorting environments, we propose an improved model, SAFYOLOv11, based on the YOLOv11n [19] architecture. Our method is designed to balance detection performance with computational efficiency by systematically enhancing three key components of the detection pipeline: the backbone, the neck, and the loss function. The overall architecture of our proposed model is illustrated in Fig. 1.

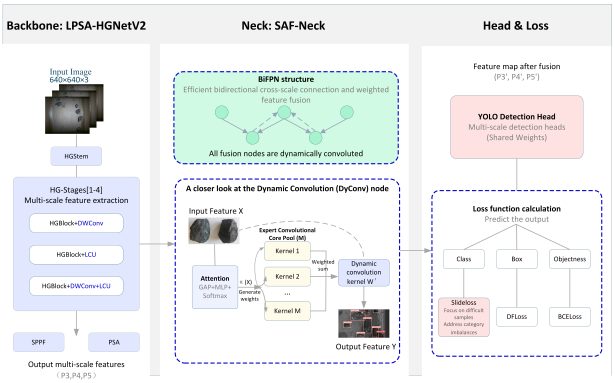

Fig. 1. The overall architecture of the proposed SAF-YOLOv11 model, highlighting the LPSA-HGNetv2 backbone, the SAF-Neck, and the integration of the SlideLoss function

2.1. Lightweight and High-Performance Backbone: LPSA-HGNetv2

To address the accuracy-efficiency trade-off, we propose LPSA-HGNetv2, a backbone network designed for a powerful and hierarchically structured feature representation. Its architecture, illustrated in Fig. 2, is built upon three core components: an HGStem module, four HG-Stages, and an efficient Probabilistic Spatial Attention (PSA) module.

The fundamental structure of LPSA-HGNetv2 is composed of HGBlock modules [20] and is optimized with Depthwise Separable Convolutions (DWConv) to reduce computational cost. To prevent information redundancy during multi-scale fusion, we introduce a Lightweight Convolution Unit (LCU) at different stages [21], following a hierarchical enhancement strategy: lower-level stages preserve semantic integrity, while higher-level stages focus on fine-grained details.

A key optimization is the replacement of the original C2PSA module with our PSA module. Inspired by the probabilistic outputs of Support Vector Machines (SVMs), PSA enhances computational efficiency by using Platt Scaling to map decision values to a probability distribution. This approach avoids complex branching computations and reduces the time complexity for kernel matrix calculations from $O\left(D^2\right)$ to $O(D \log D)$ in lower-dimensional feature spaces, making it highly effective and parameter-efficient.

HGStem and HG-Stages: The HGStem module processes the input via a dual-path mechanism (max-pooling and convolution) to generate a feature map at 1/4 of the original size, significantly reducing the initial computational load. Subsequently, the four HG-Stages extract multi-scale features. We strategically embed 1, 2, and 1 LCU modules into Stages 2, 3, and 4, respectively. These units are combined with Depthwise Separable Convolutions (DWConv) to construct a high-efficiency feature hierarchy that enriches representation without substantial overhead.

The efficiency of using DWConv over standard convolution is evident from its computational complexity. The parameter count (P) and floating-point operations (FLOPs, F) for a standard convolution are:

$$\begin{cases} P = C_{\text{in}} \times C_{\text{out}} \times K \times K \\ F = C_{\text{in}} \times C_{\text{out}} \times H \times W \times K \times K \end{cases} \tag{1}$$

For DWConv, these are:

$$\begin{cases} P_{DW} = C_{\text{in}} \times K^2 + C_{\text{in}} \times C_{\text{out}} \\ F_{DW} = C_{\text{in}} \times H \times W \times K^2 + C_{\text{in}} \times C_{\text{out}} \times H \times W \end{cases} \tag{2}$$

where C_{in} and C_{out} are the input/output channels, K is the kernel size, and H, W are the feature map dimensions. The ratio of their parameter counts is:

$$\frac{P_{DW}}{P} = \frac{F_{DW}}{F} = \frac{1}{C_{\text{out}}} + \frac{1}{K^2} \tag{3}$$

Eq. (3) shows that the parameter count and computational cost of DWConv are substantially lower than those of standard convolution.

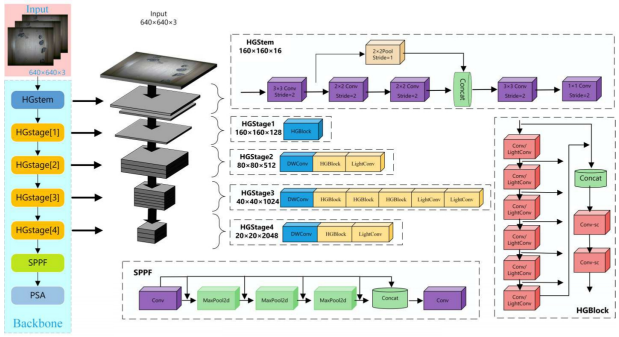

Fig. 2. The detailed architecture of the LPSA-HGNetv2 backbone, showing the HGStem, four HG-Stages with integrated DWConv and LightConv modules, and the final SPPF and PSA modules. The dimensions of the output feature maps at each stage are also indicated

2.2. Semantic-Aware Fusion Neck: A Paradigm Shift from Static to Dynamic Adaptive Fusion

2.2.1. *Motivation: The Fundamental Limitation of Conventional Feature Fusion*

The neck component of a detector is critical for fusing multi-scale features from the backbone. However, conventional necks like FPN [22], illustrated in Fig. 3A, and PANet [22],illustrated in Fig. 3B, rely on static convolutions. These networks use a fixed set of kernels for all inputs, which becomes a significant limitation in class-imbalanced scenarios. The learned kernels inevitably become biased towards the majority class, suppressing the unique features of the minority class.

Conventional networks rely on fixed, content-agnostic kernels that apply identical parameters to all inputs. Under severe class imbalance, these kernels inevitably bias toward the majority class during training, effectively becoming "majority experts". Consequently, subtle minority-class features are marginalized or lost. This structural "static fusion bottleneck" creates a representational deficit that cannot be fully resolved by back-end loss functions alone.

2.2.2. *SAF-Neck: A New Paradigm of "Dynamic, Tailored" Fusion*

To fundamentally break this bottleneck, we propose the SAF-Neck. Its core idea is to introduce a new paradigm of "dynamic, tailored" fusion, replacing the static convolution with a dynamic, input-adaptive mechanism. SAF-Neck is not a mere technical replacement but a new problem-solving philosophy: it empowers the network to dynamically "tailor" the most suitable convolutional kernel for each fusion operation based on the semantic content of the input [23]. The mechanism works by using a lightweight attention module to analyze the global semantic information of an input feature map. Based on this analysis, it linearly combines multiple base kernels from a pre-defined "expert kernel pool" to synthesize an optimal, dynamic kernel specifically for the current input.

This adaptive mechanism brings a qualitative change. For instance, when processing an image containing the minority-class "coal", the attention network can guide the generation of a kernel that specializes in capturing its unique texture and detailed features. Conversely, when processing the majority-class "gangue", it can generate a different kernel that focuses on its overall shape and contour. In this manner, SAF-Neck ensures that minority-class samples receive high-quality, discriminative feature representations, fundamentally preventing their features from being masked by those of the majority class during fusion and significantly mitigating the representational bias caused by class imbalance.

2.2.3. *Implementation and Synergy*

Our SAF-Neck architecture is built upon the efficient Bidirectional Feature Pyramid Network (BiFPN) structure, which utilizes weighted feature fusion with learnable weights as defined in Eq. (4).

$$O = \sum_i \frac{w_i}{\sum_j w_j + \epsilon} \cdot I_i \tag{4}$$

where w_i are learnable weights and ϵ is a small value for numerical stability.

In a typical BiFPN node, such as the one described in Eq. (5), Eq. (6), the primary limitation is the static Conv operation, which is content-agnostic.

$$P_i^{td} = \text{Conv}\left(\text{Swish}\left(\frac{w_1 \cdot P_i^{in} + w_2 \cdot \text{Resize}\left(P_{i+1}^{td}\right)}{w_1 + w_2 + \epsilon}\right)\right) \tag{5}$$

$$P_i^{out} = \text{Conv}\left(\text{Swish}\left(\frac{w_1' \cdot P_i^{in} + w_2' \cdot P_i^{td} + w_3' \cdot \text{Resize}\left(P_{i-1}^{out}\right)}{w_1' + w_2' + w_3' + \epsilon}\right)\right) \tag{6}$$

where P_i^{in} , P_i^{td} , and P_i^{out} denote input, top-down, and bottom-up features at level i, while w and w' are learnable weights. Our key innovation is to replace this static operation with a dynamic, content-adaptive mechanism using Dynamic Convolution (DyConv). Instead of a fixed filter, DyConv maintains a pool of M "expert" kernels and, as shown in Eq. (7), linearly combines them into a single, tailored dynamic kernel using input-specific coefficients generated by the attention mechanism detailed in Eq. (8).

$$W'(\alpha(X)) = \sum_{m=1}^{M} \alpha_m(X) W_m \tag{7}$$

where the attention coefficients $\alpha(X) = \{\alpha_1(X), \ldots, \alpha_M(X)\}$ are generated as follows:

$$\alpha(X) = \text{Softmax}(\text{MLP}(\text{GlobalAvgPool}(X))) \tag{8}$$

Specifically, input X undergoes GAP and MLP processing to extract global features. These are normalized via Softmax to generate weights α, which linearly combine M expert kernels into a unified content-adaptive convolution kernel.

By integrating this dynamic process into the BiFPN framework, we derive the unified mathematical expression for the SAF-Neck node: first, input features are fused via a weighted scheme to produce X_{fused} as shown in Eq. (9); subsequently, this fused feature map is processed by our dynamically generated kernel, as detailed in Eq. (10).

$$X_{\text{fused}} = \text{Swish}\left(\frac{w_1 \cdot P_i^{\text{in}} + w_2 \cdot \text{Resize}\left(P_{i+1}^{\text{td}}\right)}{w_1 + w_2 + \epsilon}\right) \tag{9}$$

$$p_i^{\text{td-dynamic}} = \sum_{k=1}^{M} \alpha_k\left(X_{\text{fused}}\right) \cdot \left(W_k^* * X_{\text{fused}}\right) \tag{10}$$

The fused feature is then processed by our dynamically generated kernel. This design not only transforms the entire feature fusion process into a content-aware, adaptive operation but also achieves exceptional computational efficiency. Unlike complex modules that typically increase the computational burden, our ablation study demonstrates that the introduction of SAF-Neck is a key contributor to reducing the model's overall GFLOPs from 6.6 to 5.7. A standard convolutional layer has a computational cost (FLOPs) of:

$$\text{FLOPs} = W' \times H' \times C_{\text{out}} \times C_{\text{in}} \times K \times K \tag{11}$$

The computational cost for the SAF-Neck, incorporating the dynamic kernel generation, is given as:

$$\begin{aligned} FLOPs' = {} & M \times C_{\text{out}} \times C_{\text{in}} \times K \times K + \\ & W' \times H \times C_{\text{out}} \times C_{\text{in}} \times K \times K + \\ & C_{\text{in}}^2 + M \times C_{\text{in}} \end{aligned} \tag{12}$$

This efficiency stems from the inherent mechanism of dynamic convolution: while it leverages M expert kernels, these are aggregated into a single kernel before the convolution operation is performed, keeping its inference-stage computational cost comparable to a standard convolution. This unique combination of powerful performance and computational economy makes SAF-Neck particularly effective. Finally, this "front-end feature enhancement" paradigm forms a powerful synergy with the SlideLoss function we adopt at the optimization level, creating a complete "front-end feature enhancement and back-end optimization guidance" pipeline.

The implementation details are provided in algorithm 1 and in Fig. 3C.

2.3. Imbalance-Robust Loss Function: SlideLoss

The significant class imbalance in the dataset poses a major challenge to model training. To address this, we introduce the SlideLoss function [24] to replace the standard loss function. SlideLoss is specifically designed to mitigate the effects of class imbalance by dynamically reweighting samples based on their difficulty. Its core mechanism focuses the model's attention on hard examples that are difficult to classify correctly.

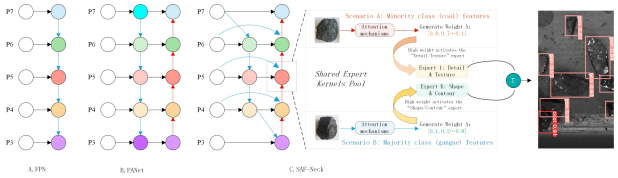

Fig. 3. Comparison of feature fusion network architectures. (A) FPN with a top-down pathway. (B) PANet adds a bottom-up pathway. (C) Our proposed SAF-Neck, which employs dynamic convolution to generate input-adaptive fusion strategies, illustrated by two different scenarios for coal and gangue class features

As a loss function, SlideLoss features an adaptive reweighting scheme that dynamically adjusts the contribution of each sample to the total loss. This is achieved by evaluating the model's prediction against a predefined threshold, μ. The formula is given as:

$$f(x) = \begin{cases} 1, & x \leq \mu - 0.1 \\ e^{\frac{1-x}{0.1}}, & \mu - 0.1 < x < \mu \\ e^{1-mu}, & x \geq \mu \end{cases} \tag{13}$$

Here, x denotes the prediction score (e.g., IoU) and μ is the threshold distinguishing easy from hard samples. SlideLoss assigns higher weights to "hard examples" ($x < \mu$) located near the decision boundary, such as visually similar coal and gangue. This prevents scarce minority samples from being overlooked, compelling the model to learn robust discriminative features for improved generalization in imbalanced scenarios.

3. Experiments

3.1. Dataset and Preprocessing

The dataset was collected at the Fuli Coal Mine in Hegang City, Heilongjiang Province. To simulate a controlled industrial environment, data was acquired using a light-blocking enclosure over a conveyor belt, illuminated by six uniform strip light sources. We used a Hikvision MV-CS050-60GC industrial camera. After filtering, we obtained a final dataset of 3,564 images, which were manually annotated using LabelImg. This yielded 18,552 object instances: 17,729 for gangue and 823 for coal. The dataset was divided into training (64%), validation (16%), and test (20%) sets.

3.2. Implementation Details

Experiments were conducted on a workstation with an Intel Core i5-12400F CPU and an NVIDIA GeForce RTX 4060 GPU (8 GB VRAM). The software environment was Python 3.9.19 and PyTorch 1.11.0, with CUDA 11.3. All

Algorithm 1. SAF-Neck Implementation Pseudocode

Require: Feature maps $\{P_3, P_4, P_5, P_6, P_7\}$
Require: Number of dynamic experts X
Ensure: Refined multi-scale features

function DyConv(x)
  Initialize expert convolution kernels $\{K_i\}_{i=1}^{X}$
  Compute attention weights:

$$\alpha \leftarrow \text{Softmax}(\text{Conv}(\text{GAP}(x)))$$

  Aggregate dynamic kernel:

$$K \leftarrow \sum_{i=1}^{X} \alpha_i \cdot K_i$$

  return Conv2D(x, K)
function BiFPNBlock(P_3, P_4, P_5, P_6, P_7)
  Select convolution type:

$$\text{Conv} \leftarrow \begin{cases} \text{DyConv}, & \text{if dynamic fusion enabled} \\ \text{Conv2D}, & \text{otherwise} \end{cases}$$

  Perform top-down and bottom-up feature fusion
  Fuse $P_7 \rightarrow P_6$ using upsampling and convolution
  Fuse remaining feature levels iteratively
  return $\{P'_3, P'_4, P'_5, P'_6, P'_7\}$

input images were resized to 640×640 pixels. We used a batch size of 8 and trained all models for 300 epochs, with an initial learning rate of 0.01.Furthermore, the pixel values of the input images were normalized to the range of $[0, 1]$ by dividing by 255.0 to facilitate model convergence.

3.3. Evaluation Metrics

We adopted standard metrics including Precision, Recall, F1-Score, mean Average Precision (mAP), and GFLOPs. The formulas are as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{14}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{15}$$

$$F1 - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{17}$$

In our experiments, we report mAP at a fixed IoU threshold of 0.5 (denoted as mAP@0.5) and the mAP averaged

over IoU thresholds from 0.5 to 0.95 in steps of 0.05 (denoted as mAP@.5:.95). GFLOPs: Giga Floating Point Operations. This metric measures the computational complexity required for a single forward pass of the model on a given input size, indicating its inference speed.

3.4. Ablation Study

To validate the contribution of each component, we conducted a series of ablation experiments. The results are detailed in Table 1. The study confirms that while each component provides individual benefits, their true strength lies in their synergistic integration. The final SAF-YOLOv11 model outperforms the baseline by 4.1% in F1-score while being 9.5% more efficient in terms of GFLOPs.

To systematically validate the contribution of each proposed component, we conducted a series of ablation experiments. Starting with the YOLOv11n baseline, we incrementally integrated our proposed modules-LPSA-HGNetv2, SlideLoss, and SAF-Neck and evaluated the model's performance at each stage. The detailed results are presented in Table 1.

Effect of LPSA-HGNetv2 Backbone: When replacing the original backbone with LPSAHGNetv2 (Model 2 vs. Model 1), we observe a significant performance improvement. The mAP@0.5 increases by 1.1 percentage points (from 92.4% to 93.5%). More importantly, the precision and recall for the minority class "coal" rise by 3.0% (from 81.9% to 84.9%) and 3.5% (from 81.3% to 84.8%), respectively. This confirms that LPSA-HGNetv2 provides a superior feature representation. Simultaneously, the computational cost remains identical to the original model (both at 6.6 GFLOPs), establishing a stronger foundation for subsequent modules.

Effect of SlideLoss: Integrating SlideLoss directly into the baseline model (Model 3 vs. Model 1) addresses the class imbalance problem head-on. Without altering the network architecture, the introduction of SlideLoss boosts the precision and recall for the "coal" class by 4.7% (from 81.9% to 86.6%) and 4.7% (from 81.3% to 86.0%), respectively. This strongly demonstrates its effectiveness in forcing the model to focus on hard-to-classify, minority-class samples.

Effect of SAF-Neck: The most critical analysis comes from isolating the impact of our core innovation, the SAF-Neck. We compare our final model (Model 7) with a model that includes LPSA-HGNetv2 and SlideLoss but lacks the dynamic neck (Model 5). The addition of SAF-Neck not only elevates the final mAP@0.5 and F 1 -score to 94.1% and 0.904 , respectively, but also increases the recall for "coal" by a notable 2.3 percentage points (from 84.8% to 87.1%). Critically, it also significantly reduces the computational cost, lowering the GFLOPs from 6.6 to 5.7. This result is pivotal, as it shows that the dynamic, content-aware fusion mechanism of SAF-Neck enables a more efficient feature processing pipeline, more than compensating for the overhead of other modules.

Conclusion: The ablation study confirms that while each component provides individual benefits, their true strength lies in their synergistic integration. The final SAF-YOLOv11 model is not merely a stack of modules but a systematic architectural redesign. In this design, a more powerful backbone (LPSA-HGNetv2) and a specialized loss function (SlideLoss) create the ideal conditions for our dynamic neck (SAF-Neck) to simultaneously achieve superior accuracy and computational efficiency. The final model (Model 7) outperforms the baseline (Model 1) by 4.1 percentage points in F1-score (from 0.863 to 0.904) while being 13.6% more computationally efficient (GFLOPs reduced from 6.6 to 5.7).

3.5. Comparative test

To comprehensively benchmark the performance of SAF-YOLOv11, we conducted a comparative analysis against several classic and state-of-the-art (SOTA) object detection models. These include SSD, YOLOv3-tiny, YOLOv4, YOLOv6, YOLOv8s, YOLOv10n, and YOLOv11s. All models were trained and evaluated under identical experimental conditions on our self-constructed coal and gangue dataset.

As summarized in Table 1, the results highlight the superior overall performance of our proposed SAF-YOLOv11 model. It achieves the highest scores across multiple key metrics, including an overall Precision of 92.3%, an mAP@0.5 of 94.1%, and an F1-Score of 0.904 . Notably, it also excels in identifying the challenging minority class, "coal," attaining the highest recall of 87.1%. This metric is critical for minimizing the loss of valuable resources (coal) during industrial sorting.

A core advantage of our model lies in its unparalleled balance between accuracy and computational efficiency. While delivering top-tier detection accuracy, SAF-YOLOv11 operates at a mere 5.7 GFLOPs. This is far more efficient than other high-performing models like YOLOv8s (28.4 GFLOPs) and YOLOv11s (21.3 GFLOPs), requiring only 20% and 27% of their respective computational resources. Its efficiency even surpasses well-known lightweight models such as YOLOv3-tiny (14.3 GFLOPs) and YOLOv10n (8.2 GFLOPs), making it the most computationally economical model among all competitors.

In conclusion, the comparative experiments validate that SAF-YOLOv11 establishes a new SOTA performance benchmark for this task. It offers a highly effective and powerful solution that is perfectly suited for deployment

Table 1. Ablation study results on the coal-gangue dataset

Model	P (%)	P (coal) (%)	R (%)	R (coal) (%)	mAP@0.5 (%)	GFLOPs	F1
(1) YOLOv11n	89.5	81.9	83.4	81.3	92.4	6.6	0.863
(2) YOLOv11n + LPSA-HGNetv2	90.8	84.9	87.5	84.8	93.5	6.6	0.891
(3) YOLOv11n + SlideLoss	91.5	86.6	87.7	86.0	93.5	6.3	0.895
(4) YOLOv11n + BiFPN	91.7	86.7	87.3	85.4	93.1	7.0	0.894
(5) YOLOv11n + SlideLoss + LPSA-HGNetv2	91.9	87.6	87.3	84.8	94.0	6.6	0.895
(6) YOLOv11n + LPSA-HGNetv2 + BiFPN + SlideLoss	90.9	85.5	87.8	85.4	93.8	6.5	0.893
(7) SAF-YOLOv11 (Ours)	**92.3**	**87.8**	**88.6**	**87.1**	**94.1**	**5.7**	**0.904**

in resource-constrained industrial environments.

3.6. Heatmap

To visually interpret model behavior, we used Grad-CAM [28] to generate class activation heatmaps. As illustrated in Fig. 4, the heatmaps for SAF-YOLOv11 are significantly more focused than the baseline. This improved focus is a direct result of the SAF-Neck. Furthermore, the detection results in Fig. 5 demonstrate that the SAF-YOLOv11 model successfully detects objects missed by the baseline and exhibits higher confidence scores, particularly for the minority "coal" class.

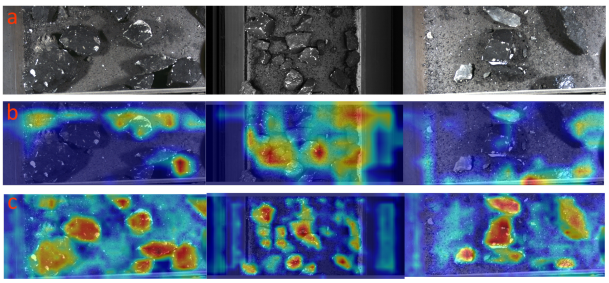

Fig. 4. Grad-CAM visualization comparing the feature localization capabilities of SAF-YOLOv11 and the baseline model. (a) Original input images. (b) Class activation heatmaps of the baseline YOLOv11n model, showing diffuse activation. (c) Heatmaps of our proposed SAF-YOLOv11, demonstrating precise focus on target objects

3.7. Limitations

Despite robust performance, limitations exist. Missed detections may occur under extreme occlusion or low illumination where visual features become indistinguishable. Additionally, high-speed motion blur can occasionally affect localization accuracy. Future work will explore multimodal fusion to resolve these physical constraints.

4. Conclusion

In this paper, we addressed challenges in industrial coal and gangue sorting. We proposed SAF-YOLOv11, a redesigned architecture that balances detection performance and computational cost. Our main contributions are:

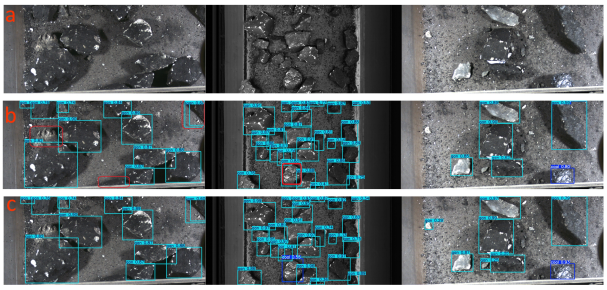

Fig. 5. Comparison of detection performance between SAF-YOLOv11 and the baseline model in complex industrial scenarios. (a) Challenging samples. (b) Detection results from the baseline YOLOv11n, showing missed detections. (c) Detection results from our proposed SAFYOLOv11, which successfully detects all targets

- We introduced the Semantic-Aware Fusion Neck (SAF-Neck), a core innovation using a dynamic, input-adaptive routing mechanism to mitigate class imbalance at the feature level.
- We developed a synergistic architecture where a high-performance backbone (LPSA-HGNetv2) and the SAF-Neck resolve the accuracy-efficiency bottleneck.
- We integrated the SlideLoss function, creating a comprehensive solution with front-end feature enhancement and back-end optimization.

Experimental results demonstrate that SAF-YOLOv11 outperforms the baseline, improving the F1-score by 4.1% and coal recall by 5.8% while reducing computational load by 13.6%. Benchmarks confirm it as a robust, efficient solution for automated coal and gangue identification in imbalanced industrial environments.

5. Funding

This work was supported by the Liaoning Provincial Graduate Education and Teaching Reform Research Project

Table 2. Performance comparison with other state-of-the-art models on the coal-gangue dataset

Model	P (%)	P (coal) (%)	R (%)	R (coal) (%)	mAP@0.5 (%)	GFLOPs	F1
SSD [25]	-	-	-	-	85.2	-	-
YOLOv3-tiny	91.3	88.5	84.3	76.3	93.6	14.3	0.876
YOLOv4 [26]	-	-	-	-	89.0	38.9	-
YOLOv6	86.6	77.7	88.1	84.8	92.4	11.5	0.873
YOLOv8s	91.9	88.5	88.6	86.0	92.5	28.4	0.902
YOLOv10n	90.8	86.7	88.4	84.8	93.9	8.2	0.895
YOLOv11s	91.8	86.8	85.8	84.2	93.9	21.3	0.886
Fine-tunedYOLOv5s [27]	-	-	-	-	93.0	6.0	-
SAF-YOLOv11 (Ours)	92.3	87.8	88.6	87.1	94.1	5.7	0.904

(GrantNo. LNYJG2023077); the Fundamental Research Funds for the Provincial Universities of Liaoning Province (Grant Nos.LJ212410144022 and LJ232410144074); and the Liaoning Provincial Science and Technology Plan Joint Plan (Natural ScienceFoundation - General Program) project "Research on robot active perception technology for intelligent assembly of XX productionline"(Grant No.2025-MSLH-589).

References

[1] J. Li and J. Wang, (2019) *"Comprehensive utilization and environmental risks of coal gangue: A review"* **Journal of Cleaner Production *239***: 117946.

[2] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, (2017) *"A survey on deep learning in medical image analysis"* **Medical image analysis *42***: 60–88.

[3] H. Jin, C. Yu, Z. Gong, R. Zheng, Y. Zhao, and Q. Fu, (2023) *"Machine learning techniques for pulmonary nodule computer-aided diagnosis using CT images: A systematic review"* **Biomedical Signal Processing and Control *79***: 104104.

[4] R. U. Modi, M. Kancheti, A. Subeesh, C. Raj, A. K. Singh, N. S. Chandel, A. S. Dhimate, M. K. Singh, and S. Singh, (2023) *"An automated weed identification framework for sugarcane crop: a deep learning approach"* **Crop Protection *173***: 106360.

[5] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, (2021) *"Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues"* **Array *10***: 100057.

[6] R. Ameri, C.-C. Hsu, and S. S. Band, (2024) *"A systematic review of deep learning approaches for surface defect detection in industrial applications"* **Engineering Applications of Artificial Intelligence *130***: 107717.

[7] Y. Gao, J. Lin, J. Xie, and Z. Ning, (2020) *"A real-time defect detection method for digital signal processing of industrial inspection applications"* **IEEE Transactions on Industrial Informatics *17***(5): 3450–3459.

[8] J. Redmon and A. Farhadi, (2018) *"Yolov3: An incremental improvement"* **arXiv preprint arXiv:1804.02767**:

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, (2020) *"Yolov4: Optimal speed and accuracy of object detection"* **arXiv preprint arXiv:2004.10934**:

[10] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, et al., (2022) *"YOLOv6: A single-stage object detection framework for industrial applications"* **arXiv preprint arXiv:2209.02976**:

[11] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, et al., (2024) *"Yolov10: Real-time end-to-end object detection"* **Advances in Neural Information Processing Systems *37***: 107984–108011.

[12] Z. Lv, Y. Fan, T. Sha, Y. Cui, Y. Wu, H. Lv, M. Sun, Y. Tu, Z. Xu, and W. Wang, (2025) *"A large-scale open image dataset for deep learning-enabled intelligent sorting and analyzing of raw coal"* **Scientific Data *12***(1): 403. DOI: 10.1038/s41597-025-04719-0.

[13] R. Li, L. Zhao, H. Wei, G. Hu, Y. Xu, B. Ouyang, and J. Tan, (2025) *"Multi-defect type beam bridge dataset: GYU-DET"* **Scientific Data *12***(1): 1101. DOI: 10.1038/s41597-025-05395-w.

[14] Z. Liu, Y. Wang, L. Ma, Y. Wu, G. He, X. Liang, and F. Wang, (2025) *"CUs-YOLO: enhanced feature fusion model for coal and gangue recognition in complex environment of coal mine"* **Measurement Science and Technology *36***(6): 065012.

[15] X. WEI, F. WANG, D. HE, C. LIU, and D. XU, (2024) *"Coal gangue image recognition model based on CSPNet-YOLOv7 target detection algorithm"* **Coal Science and Technology *52***(S1): 238–248.

[16] N. Li, K. Qin, X. Li, and A. Zhang, (2025) *"A YOLOv7-based coal and gangue recognition model integrating super-resolution reconstruction"* **Computer Engineering and Applications *61***(15): 343–352.

[17] R. Khanam and M. Hussain, (2024) *"Yolov11: An overview of the key architectural enhancements"* **arXiv preprint arXiv:2410.17725**:

[18] X. Zhao, W. Zhang, H. Zhang, C. Zheng, J. Ma, and Z. Zhang, (2024) *"ITD-YOLOv8: An infrared target detection model based on YOLOv8 for unmanned aerial vehicles"* **Drones *8***(4): 161.

[19] Y. Ge, Z. Li, and L. Meng, (2025) *"YOLO-MSD: a robust industrial surface defect detection model via multi-scale feature fusion"* **Applied Intelligence *55***(12): 1–18.

[20] W. Lv, S. Xu, Y. Zhao, G. Wang, J. Wei, C. Cui, Y. Du, Q. Dang, and Y. Liu, (2023) *"Detrs beat yolos on real-time object detection"* **CoRR**:

[21] H. Shen, Z. Wang, J. Zhang, and M. Zhang, (2024) *"L-Net: A lightweight convolutional neural network for devices with low computing power"* **Information Sciences *660***: 120131.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 2117–2125.

[23] K. Han, Y. Wang, J. Guo, and E. Wu, (2023) *"ParameterNet: Parameters are all you need"* **arXiv preprint arXiv:2306.14525**:

[24] Z. Yu, H. Huang, W. Chen, Y. Su, Y. Liu, and X. Wang, (2024) *"Yolo-facev2: A scale and occlusion aware face detector"* **Pattern Recognition *155***: 110714.

[25] Z. Cao, L. Fang, Z. Li, and J. Li, (2023) *"Lightweight target detection for coal and gangue based on improved Yolov5s"* **Processes *11***(4): 1268.

[26] Y. Sui, L. Zhang, Z. Sun, W. Yi, and M. Wang, (2024) *"Research on Coal and Gangue Recognition Based on the Improved YOLOv7-Tiny Target Detection Algorithm"* **Sensors *24***(2): 456.

[27] D. Shang, Z. Lv, Z. Gao, and Y. Li, (2025) *"Detection of coal gangue by YOLO deep learning method based on channel pruning"* **International Journal of Coal Preparation and Utilization *45***(1): 231–243.

[28] H. Zhang and K. Ogasawara, (2023) *"Grad-CAM-based explainable artificial intelligence related to medical text processing"* **Bioengineering *10***(9): 1070.