

Article

MSG-YOLO: A Lightweight Detection Algorithm for Clubbing Finger Detection

Zhijie Wang ^{1,2}, Qiao Meng ^{1,2,*}, Feng Tang ^{3,*}, Yuelin Qi ³, Bingyu Li ^{1,2}, Xin Liu ^{1,2}, Siyuan Kong ^{1,2} and Xin Li ^{1,2}

¹ School of Computer Technology and Application, Qinghai University, Xining 810016, China; ys230854040325@qhu.edu.cn (Z.W.); ys230854040300@qhu.edu.cn (B.L.); ys230854040328@qhu.edu.cn (X.L.); ys230854100347@qhu.edu.cn (S.K.); ys220854100361@qhu.edu.cn (X.L.)

² Intelligent Computing and Application Laboratory of Qinghai Province, Xining 810016, China

³ Research Center for High Altitude Medicine, Qinghai University, Xining 810016, China; ys221001020765@qhu.edu.cn

* Correspondence: 2010990037@qhu.edu.cn (Q.M.); 2012980002@qhu.edu.cn (F.T.)

Abstract: Clubbing finger is a significant clinical indicator, and its early detection is essential for the diagnosis and treatment of associated diseases. However, traditional diagnostic methods rely heavily on the clinician's subjective assessment, which can be prone to biases and may lack standardized tools. Unlike other diagnostic challenges, the characteristic changes of clubbing finger are subtle and localized, necessitating high-precision feature extraction. Existing models often fail to capture these delicate changes accurately, potentially missing crucial diagnostic features or generating false positives. Furthermore, these models are often not suited for accurate clinical diagnosis in resource-constrained settings. To address these challenges, we propose MSG-YOLO, a lightweight clubbing finger detection model based on YOLOv8n, designed to enhance both detection accuracy and efficiency. The model first employs a multi-scale dilated residual module, which expands the receptive field using dilated convolutions and residual connections, thereby improving the model's ability to capture features across various scales. Additionally, we introduce a Selective Feature Fusion Pyramid Network (SFFPN) that dynamically selects and enhances critical features, optimizing the flow of information while minimizing redundancy. To further refine the architecture, we reconstruct the YOLOv8 detection head with group normalization and shared-parameter convolutions, significantly reducing the model's parameter count and increasing computational efficiency. Experimental results indicate that the model maintains high detection accuracy with reduced parameter and computational requirements. Compared to YOLOv8n, MSG-YOLO achieves a 48.74% reduction in parameter count and a 24.17% reduction in computational load, while improving the mAP_{0.5} score by 2.86%, reaching 93.64%. This algorithm strikes a balance between accuracy and lightweight design, offering efficient and reliable clubbing finger detection even in resource-constrained environments.

Keywords: clubbing finger detection; YOLOv8; lightweight network; multi-scale feature extraction; feature fusion



Citation: Wang, Z.; Meng, Q.; Tang, F.; Qi, Y.; Li, B.; Liu, X.; Kong, S.; Li, X. MSG-YOLO: A Lightweight Detection Algorithm for Clubbing Finger Detection. *Electronics* **2024**, *13*, 4549. <https://doi.org/10.3390/electronics13224549>

Academic Editor: George A. Tsihrintzis

Received: 28 October 2024

Revised: 15 November 2024

Accepted: 18 November 2024

Published: 19 November 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

Clubbing finger is a critical clinical sign, often indicating severe underlying conditions such as lung cancer, pulmonary infections, interstitial lung disease, cystic fibrosis, or cardiovascular disorders [1–3]. Early detection of these conditions is crucial for timely intervention and prognosis, highlighting the importance of accurate identification of clubbing finger in assessing patient health. Traditional diagnostic approaches rely heavily on clinical experience, but individual differences, subtle symptoms, and the absence of standardized diagnostic tools often lead to misdiagnosis and inconsistent results [4]. Additionally, the increasing demand for multidisciplinary collaboration and imaging studies has prolonged diagnostic timelines and raised associated costs. With the growing prevalence of respiratory and cardiovascular diseases, there is an urgent need for non-invasive, rapid diagnostic

solutions. Therefore, an effective, accurate, and objective method for detecting clubbing finger is essential to enhance diagnostic accuracy and efficiency.

Unlike other diagnostic challenges, the changes associated with clubbing finger are subtle and localized, typically manifesting as soft tissue proliferation at the fingertips or toes, leading to an elevated nail bed and a characteristic club-like appearance [5,6]. These subtle changes require high-precision feature extraction and fusion. Current YOLO models may fail to capture critical diagnostic features or generate false positives when processing such fine details. To achieve rapid and accurate clinical diagnosis, particularly in resource-constrained settings, it is necessary to balance detection accuracy with computational efficiency. Designing a specialized model that captures these specific features while reducing computational load is crucial for practical deployment.

Recent advancements in medical imaging technology have enabled automated detection, and computer-aided diagnosis (CAD) systems have made significant progress in areas such as lung nodule detection, cardiac lesion identification, and dermatological disease diagnosis [7,8]. Deep learning algorithms based on convolutional neural networks (CNNs) are capable of quickly processing large volumes of image data to identify subtle features that might be missed through visual inspection alone. As deep learning techniques advanced, Jarallah et al. introduced transfer learning methods using pre-trained models such as DenseNet201 and ResNet50 to efficiently classify four nail disease types and enhance model generalization through data augmentation [9]. Karunarathne et al. further refined detection methods using InceptionV3 and DenseNet121, successfully detecting a range of nail abnormalities by combining color, shape, and texture features [10]. Soğukkuyu et al. employed VGG16 and VGG19 for transfer learning to classify melanonychia, Beau's lines, and nail clubbing, with their approach outperforming traditional CNN models in terms of accuracy and error metrics [11]. The Nail Insight system developed by Pathan et al. integrated VGG16 and GoogleNet, utilizing both feature- and decision-level fusion strategies to improve detection effectiveness for a variety of nail diseases [12]. Hsu et al. designed a system that combined YOLOv8 with U-Net for detection and segmentation, which excelled at identifying clubbing finger. However, its high computational complexity and resource demands hinder its application in resource-limited environments like mobile and embedded systems [13]. Despite demonstrating the potential of deep learning in nail disease detection, challenges remain for clubbing finger detection, including a small dataset with limited high-quality labeled data, which affects model training and generalization. Furthermore, the complex and variable morphology of clubbing finger, along with significant individual differences, complicates feature extraction and detection. Traditional YOLO models often fail to detect these fine features accurately, and their computational complexity limits their use in resource-limited platforms.

To overcome these challenges, we propose an improved lightweight YOLOv8 detection algorithm—MSG-YOLO. This algorithm integrates a multi-scale dilated residual module (C2f_MDR) and a lightweight selective feature fusion pyramid network (SFFPN), aiming to improve detection accuracy and real-time performance while reducing model parameters and computational complexity. The C2f_MDR module better extracts features across multiple scales, enhancing the model's ability to capture relevant contextual information for detecting clubbing finger. To optimize the feature fusion process, we designed the SFFPN, which uses a channel attention mechanism to focus on the most critical features, improving detection accuracy while reducing computational load. The group normalization shared parameter detection head (GNSCD) further minimizes the model's parameter count and computational complexity, making it more suitable for deployment on devices with limited resources. Furthermore, we collaborated with the Qinghai University Plateau Medical Research Institute to construct a larger clubbing finger dataset, providing high-quality samples for model training. The key contributions of this paper are as follows:

1. A novel multi-scale dilated residual module (C2f_MDR) is proposed, effectively improving multi-scale feature extraction while keeping the model lightweight.

2. A lightweight selective feature fusion network (SFFPN) is designed, which optimizes multi-scale feature fusion using a channel attention mechanism, enhancing detection accuracy and reducing computational complexity.
3. A group normalization shared parameter detection head (GNSCD) is introduced, significantly reducing model parameters and computational complexity, thereby increasing detection efficiency.

The structure of this paper is as follows: Section 2 reviews related work, particularly lightweight YOLO models and feature fusion strategies. Section 3 presents the architecture of the proposed MSG-YOLO model along with its technical details. Section 4 discusses the experimental setup and evaluation methods, providing a detailed analysis of model performance. Finally, Section 5 concludes this paper and outlines directions for future research.

2. Related Work

2.1. Lightweight YOLO Models

In recent years, deep learning technologies have made significant strides [14,15]. With the continued development of algorithms like Convolutional Neural Networks (CNNs), the performance of object detection tasks has dramatically improved. Among various object detection models, the YOLO (You Only Look Once) series [16–26] has been widely adopted due to its excellent real-time performance and efficiency. Over the years, from YOLOv1 to YOLOv11, improvements in model design have enhanced detection accuracy and speed. However, traditional YOLO models often have complex structures and high computational demands, making them unsuitable for deployment on resource-constrained devices. Consequently, recent research has focused on developing lightweight YOLO models. To address this issue, researchers have introduced several lightweight YOLO variants. For instance, PP-YOLOE [27] increases inference speed and reduces computational complexity while maintaining high detection accuracy by introducing an anchor-free mechanism, optimized CSPRepResNet structure, and a dynamic label assignment algorithm (TAL). The DsP-YOLO model proposed by Zhang et al. [28] features a lightweight, detail-sensitive feature fusion network (DsPAN) and an embedded attention mechanism (LCBHAM), significantly improving the model's ability to capture small target details and positional information. Zhou [29] introduced the YOLO-NL model, which innovates in the backbone network and feature fusion through dynamic label assignment strategies, enhancing performance in complex scenarios. Wang et al. [30] proposed the CTDD-YOLO model, which boosts feature extraction for complex textures and small targets using an improved CAACSPPELAN module and CGRFPN feature fusion network. Despite these advancements in inference efficiency, lightweight YOLO models still face challenges in terms of accuracy and detail capture, particularly when detecting subtle features like clubbing finger, where accuracy tends to decrease. Therefore, the key challenge in clubbing finger detection remains maintaining high precision while reducing computational load.

2.2. Feature Fusion Strategies

Multi-scale feature extraction plays a critical role in improving detection performance, especially when dealing with targets that vary in scale and possess subtle features. Classic YOLO models typically utilize Feature Pyramid Networks (FPNs) [31] for multi-scale feature extraction. However, FPNs may neglect crucial low-level features when processing complex targets such as small or irregularly shaped objects. To address these limitations, several improved multi-scale feature extraction techniques have been proposed. For example, Tan et al. [32] developed BiFPN (Bidirectional Feature Pyramid Network), which facilitates better interaction between high-level semantic and low-level detail features through bidirectional top-down and bottom-up paths. Additionally, a learnable weighted mechanism dynamically adjusts the contribution of each scale's features, improving fusion accuracy. Połap and Jaszcz [33] introduced a Twin Layer via Multiattention Networks with Feature Transfer, which optimizes feature selection and fusion by utilizing a two-layer structure to better capture the interactions and complementary information between modal-

ities. Li [34] proposed the LR-FPN model for multi-scale feature fusion in remote sensing images, which combines shallow position information extraction with a context interaction module (CIM), enhancing target localization and representation. Despite these advancements, these methods still fall short in addressing the challenges posed by clubbing finger detection. The highly individualized and complex details of clubbing finger require more specialized approaches, as traditional feature fusion methods may fail to adequately handle such challenges. Additionally, current methods struggle with precise feature selection and redundancy reduction, which impacts the accuracy of detail capture. Moreover, traditional YOLO models' complexity and computational demands lead to inefficiencies in real-time applications on resource-constrained devices. Thus, optimizing computational complexity and reducing redundant information, while maintaining high detection accuracy, remains a significant challenge in clubbing finger detection.

3. Method

3.1. Overall Structure of MSG-YOLO

To tackle the challenges of large parameter sizes and high rates of missed and false detections in existing clubbing finger detection models, we propose an improved lightweight model, MSG-YOLO, based on YOLOv8n. The revised network architecture is depicted in Figure 1.

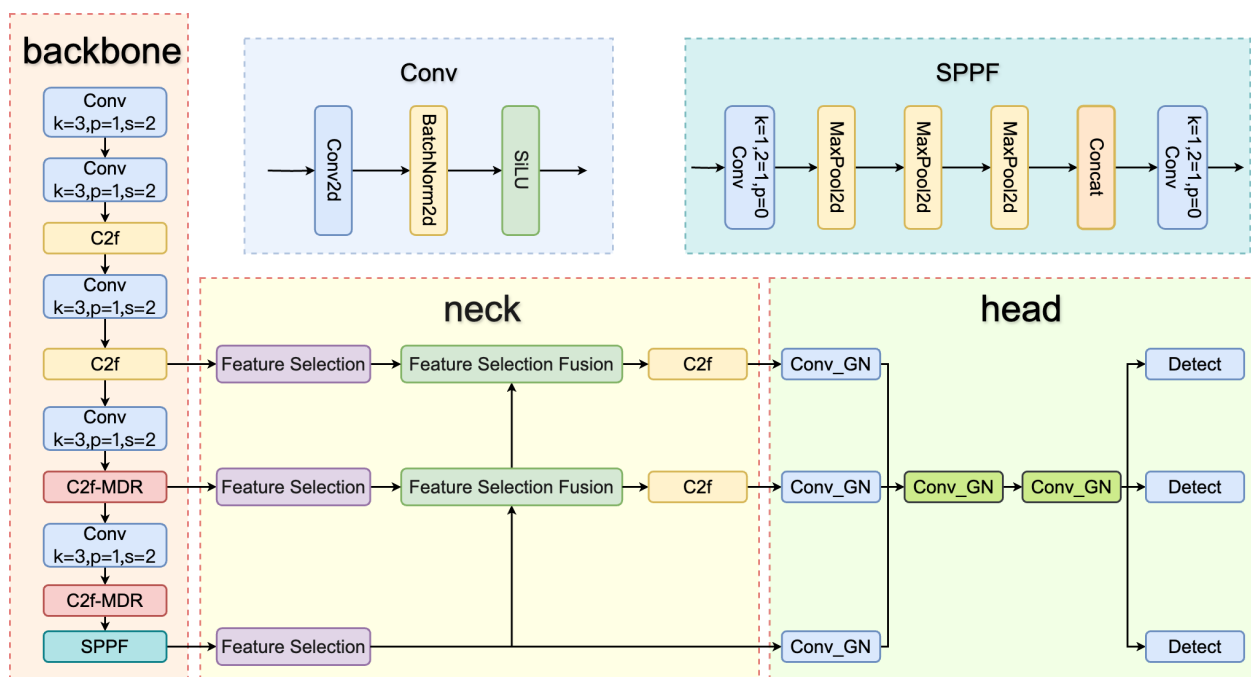


Figure 1. The overall architecture of the improved YOLO model, comprising three primary components: Backbone, Neck, and Head. The Backbone module extracts multi-level features from the input image, the Neck module merges and processes multi-scale features, and the Head module produces the target detection results. The SPPF module aggregates multi-scale information through repeated MaxPool2d operations and concatenation, and the Conv module consists of Conv2d, BatchNorm2d, and the SiLU activation function.

In MSG-YOLO, the C2f module in the Backbone is replaced with a multi-scale dilated residual module (C2f_MDR). This module combines convolutions with varying dilation rates, enabling the model to capture features across multiple scales. This approach enhances the detection of local details without increasing computational complexity, while also preserving extensive global contextual information.

For the Neck component, MSG-YOLO incorporates the Selective Feature Fusion Pyramid Network (SFFPN). The design of SFFPN specifically addresses the need for high

sensitivity when detecting subtle pathological features in clubbing finger. The selective fusion process is directed by a channel attention mechanism that dynamically adjusts the weights of each feature map channel according to the global context, ensuring that relevant features are emphasized while irrelevant or redundant ones are minimized. By integrating the channel attention mechanism with deconvolution operations, the model selectively enhances and efficiently fuses multi-scale features, improving detection performance while maintaining computational efficiency and a lightweight structure.

In the Head section, MSG-YOLO introduces a group normalization shared convolution detection head. This component combines shared convolution layers with Group Normalization (GN) [35], significantly reducing computational complexity and parameter count. The shared convolution layers allow multiple detection tasks to utilize the same feature map processing pathway, minimizing redundant computations and thereby lowering the model's parameter overhead while maintaining detection efficiency. Additionally, Group Normalization is implemented as an alternative to the traditional Batch Normalization (BN) method [36], optimizing for small batch training scenarios to achieve greater detection stability and accuracy in resource-constrained environments.

3.2. Multi-Scale Feature Extraction Residual Network (C2f_MDR)

Traditional convolutional neural networks (CNNs) are often limited by their fixed receptive fields, which restrict their ability to effectively capture target features of varying scales and deformations. To overcome this limitation, the MSG-YOLO model introduces an innovative multi-scale feature extraction residual network, known as C2f_MDR (Multi-Dilation Residual). By combining multi-scale dilated convolutions with residual connections, this structure effectively captures features across multiple scales, enhancing the model's sensitivity to fine details while keeping computational complexity low. The structural design of the C2f_MDR module is illustrated in Figure 2.

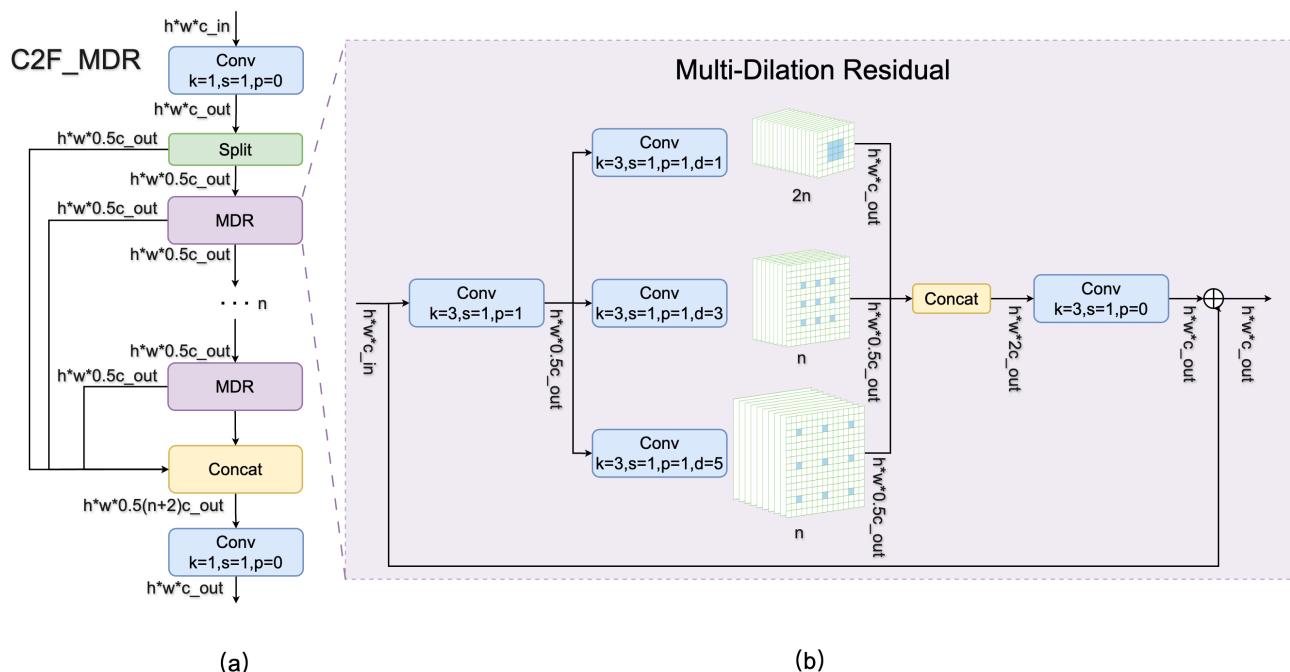


Figure 2. Structure diagram of the C2f_MDR module. (a) The overall structure of the C2f_MDR module. The input feature map undergoes convolution, segmentation, and multiple MDR module processes, followed by concatenation and convolution to restore the channel number and generate the output feature map. (b) The detailed structure of the MDR module, which consists of three convolutional layers with different dilation rates (dilation rates of 1, 3, and 5) to capture contextual information at different scales.

The MDR module includes three types of convolution operations: standard convolution ($d = 1$) and dilated convolutions with dilation rates of $d = 3$ and $d = 5$. The standard convolution ($d = 1$) is used twice as often as the dilated convolutions ($d = 3$ and $d = 5$), balancing the extraction of fine details and broader contextual information. The standard convolution focuses on capturing fine-grained local features such as edges and textures, which are crucial for detecting small or intricate targets. Meanwhile, the dilated convolutions, by expanding the receptive field, gather broader contextual information, though they may miss some finer details due to their sparsity. By increasing the number of standard convolutions, the MDR module compensates for this sparsity, ensuring that during the multi-scale information fusion process, the model maintains sensitivity to both local details and global features. This design enhances local detail capture while preserving rich global context without increasing computational complexity. The calculation process of the MDR module is expressed as follows:

$$\mathbf{F}_0 = \text{Conv}_{3 \times 3}(\mathbf{X}), \quad \mathbf{F}_0 \in \mathbb{R}^{\frac{C}{2} \times H \times W} \quad (1)$$

$\mathbf{F}_0$ is then processed through three parallel branches with dilation rates $d = 1, 3, 5$, producing outputs with channel sizes of $C, \frac{C}{2}$, and $\frac{C}{2}$. The outputs are concatenated along the channel dimension, followed by a 1×1 convolution to restore the original number of channels, and finally added back to the input $\mathbf{X}$ through a residual connection:

$$\mathbf{F}_1 = \text{Conv}_{3 \times 3}^{d=1}(\mathbf{F}_0), \quad \mathbf{F}_1 \in \mathbb{R}^{C \times H \times W} \quad (2)$$

$$\mathbf{F}_2 = \text{Conv}_{3 \times 3}^{d=3}(\mathbf{F}_0), \quad \mathbf{F}_2 \in \mathbb{R}^{\frac{C}{2} \times H \times W} \quad (3)$$

$$\mathbf{F}_3 = \text{Conv}_{3 \times 3}^{d=5}(\mathbf{F}_0), \quad \mathbf{F}_3 \in \mathbb{R}^{\frac{C}{2} \times H \times W} \quad (4)$$

$$\mathbf{F}_{\text{out}} = \mathbf{X} + \text{Conv}_{1 \times 1}(\text{Concat}(\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3)), \quad \mathbf{F}_{\text{out}} \in \mathbb{R}^{C \times H \times W} \quad (5)$$

Within the MSG-YOLO backbone, the C2f-MDR module is placed in the deeper network layers, as these layers are responsible for extracting more abstract and globally relevant information. As the network depth increases, feature maps become lower in resolution but higher in channel count, resulting in more abstract and semantically rich representations. At this stage, it is crucial for the model to integrate multi-scale information to efficiently capture features across different scales, ensuring effective detection. The C2f-MDR module expands the receptive field at these deep levels, allowing the model to capture both global context and local details. The deeper layers' complexity and abstraction require a broader perspective in feature extraction, making C2f-MDR's multi-scale perception highly advantageous. Furthermore, its residual connection design ensures smooth gradient flow, preventing vanishing or exploding gradients, thus enhancing the network's stability and trainability. Therefore, placing the C2f-MDR module in the deeper layers of the backbone significantly improves the model's ability to extract multi-scale features and enhances overall detection performance.

3.3. Selective Fusion Pyramid Network (SFFPN)

The Selective Fusion Pyramid Network (SFFPN) is integral to the MSG-YOLO architecture, designed to efficiently extract and fuse features across different scales while maintaining a lightweight structure suitable for resource-constrained environments. Traditional object detection models often utilize Feature Pyramid Networks (FPNs) to integrate multi-scale features. However, in medical imaging tasks like clubbed finger detection, where targets are typically small and exhibit subtle morphological changes, conventional feature fusion methods may not adequately capture these critical features. Therefore, we incorporate a channel attention mechanism and a deconvolution operation (ConvTranspose2d)

within the neck module to enhance the selectivity and effectiveness of feature fusion. The structural configuration of the SFFPN module is depicted in Figure 3.

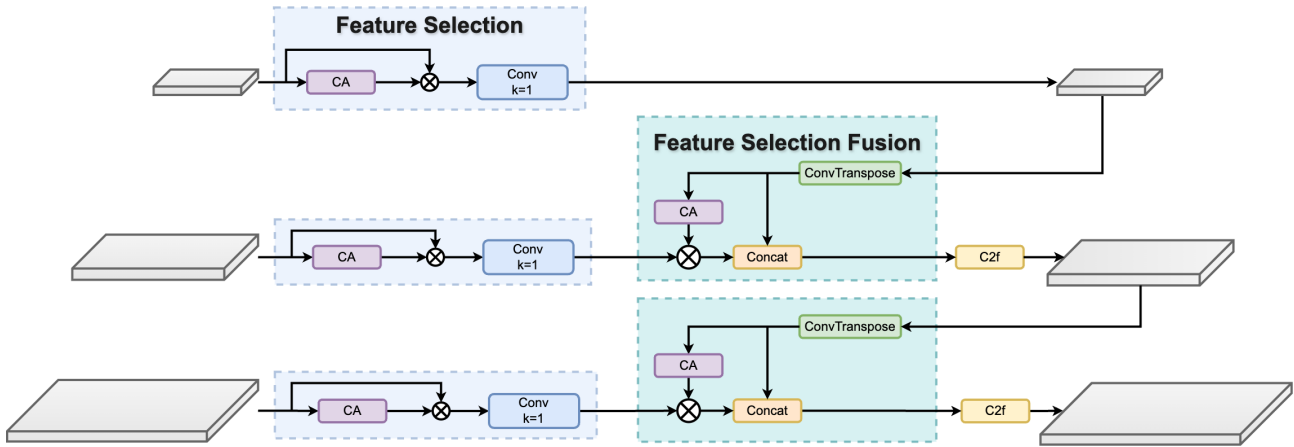


Figure 3. Schematic diagram of the SFFPN module. In the Feature Selection module, channel attention (CA) and element-wise multiplication ($\otimes$) are used to adaptively adjust the weights of the input features, followed by processing with a convolution layer (kernel size $k = 1$). The Feature Selection Fusion module then performs upsampling of features at different scales through convolution transpose (ConvTranspose), followed by concatenation (Concat) to fuse multi-scale features. The fused feature map is subsequently passed into the C2f module for further optimization, providing higher-quality features for object detection.

SFFPN builds upon the multi-scale advantages of traditional FPNs by introducing an adaptive selective fusion mechanism to optimize feature representation. Unlike conventional FPNs, which use a top-down approach to upsample high-level semantic features for merging with lower-level ones, SFFPN employs a channel attention mechanism to guide this fusion selectively. This mechanism dynamically adjusts the importance of each feature map channel based on the global context, enhancing features relevant to clubbed finger detection while suppressing unrelated or redundant information. This adaptive approach reduces the processing of unnecessary information, making feature processing more efficient. Specifically, given an input feature map $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$, where C is the number of channels, and H and W denote the height and width, respectively, we first derive the global spatial information of each channel using Global Average Pooling (GAP) and Global Max Pooling (GMP):

$$\mathbf{z}_{\text{avg}} = [\mathbf{z}_{\text{avg}}^1, \mathbf{z}_{\text{avg}}^2, \dots, \mathbf{z}_{\text{avg}}^C], \quad \text{where} \quad \mathbf{z}_{\text{avg}}^c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \mathbf{X}_{c,i,j}, \quad (6)$$

$$\mathbf{z}_{\text{max}} = [\mathbf{z}_{\text{max}}^1, \mathbf{z}_{\text{max}}^2, \dots, \mathbf{z}_{\text{max}}^C], \quad \text{where} \quad \mathbf{z}_{\text{max}}^c = \max_{i=1, \dots, H, j=1, \dots, W} \mathbf{X}_{c,i,j}.$$

Here, $\mathbf{X}_{c,i,j}$ indicates the pixel value of the input feature map at channel c , height i , and width j . The vectors $\mathbf{z}_{\text{avg}}$ and $\mathbf{z}_{\text{max}}$, obtained through global average and max pooling, encapsulate the global information for each channel.

These aggregated features are then passed through a shared Multi-Layer Perceptron (MLP) for non-linear transformation, yielding the following channel weights $\mathbf{s}$:

$$\mathbf{s} = \sigma(\mathbf{W}_2 \cdot \delta(\mathbf{W}_1 \cdot [\mathbf{z}_{\text{avg}}; \mathbf{z}_{\text{max}}])), \quad (7)$$

where $[\mathbf{z}_{\text{avg}}; \mathbf{z}_{\text{max}}]$ denotes the concatenation of the two C -dimensional vectors into a single $2C$ -dimensional vector; $\delta(\cdot)$ represents the ReLU activation, and $\sigma(\cdot)$ represents the Sigmoid activation. $\mathbf{W}_1 \in \mathbb{R}^{2C \times rC}$ and $\mathbf{W}_2 \in \mathbb{R}^{rC \times C}$ are the MLP weight matrices, with r being the reduction ratio. The output $\mathbf{s} = [s_1, s_2, \dots, s_C]$ provides channel-specific weights, where s_c denotes the weight for channel c .

Finally, these channel weights $\mathbf{s}$ are applied to the original feature map:

$$\mathbf{X}'_{c,i,j} = s_c \cdot \mathbf{X}_{c,i,j}, \quad (8)$$

where $\mathbf{X}'_{c,i,j}$ is the weighted element of the feature map. This method emphasizes channels associated with pathological features, suppressing irrelevant information, thus enhancing the model's sensitivity to subtle morphological changes in the fingers.

Another key feature of the SFFPN is the use of transposed convolution (ConvTranspose2d) for upsampling. Unlike traditional interpolation-based upsampling methods, transposed convolution uses learnable convolutional kernels to restore the spatial dimensions of the feature maps, thereby preserving and reconstructing the detailed information in low-resolution feature maps more effectively. Specifically, given an input feature map $\mathbf{X} \in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$, where C_{in} represents the number of input channels, and H_{in} and W_{in} denote the height and width of the input feature map, respectively, the output dimensions of the transposed convolution are calculated as

$$H_{out} = (H_{in} - 1) \times s - 2p + k + o, \quad (9)$$

$$W_{out} = (W_{in} - 1) \times s - 2p + k + o, \quad (10)$$

where H_{out} and W_{out} are the output feature map's height and width, respectively; k is the kernel size; s is the stride; p is the padding; and o is the output padding. The operation of the transposed convolution is formulated as

$$\mathbf{Y}_{c',i,j} = \sum_{c=1}^{C_{in}} \sum_{m=1}^k \sum_{n=1}^k \mathbf{X}_{c, \lfloor \frac{i+p-m}{s} \rfloor, \lfloor \frac{j+p-n}{s} \rfloor} \cdot \mathbf{K}_{c,c',m,n}, \quad (11)$$

where $\mathbf{Y} \in \mathbb{R}^{C_{out} \times H_{out} \times W_{out}}$ is the output feature map, with C_{out} as the number of output channels; $\mathbf{K} \in \mathbb{R}^{C_{in} \times C_{out} \times k \times k}$ represents the convolution kernel weights. The floor operation, $\lfloor \cdot \rfloor$, is applied to calculate indices. Here, i and j are the indices for the height and width of the output feature map, respectively, and m and n are the indices for the convolution kernel dimensions, respectively, with c and c' as the input and output channel indices, respectively. By learning the kernel weights, the transposed convolution operation not only enhances the model's capacity for precise spatial localization but also effectively integrates contextual information during multi-scale feature fusion. This precise feature reconstruction is crucial for accurately capturing subtle morphological details in clubbed finger detection.

In the feature fusion process, the high-level upsampled feature map and the corresponding low-level feature map first undergo a channel attention mechanism, extracting the most relevant feature channels. Subsequently, element-wise multiplication is performed to selectively combine these features, emphasizing shared activations in both feature maps to ensure effective information exchange. The fused result is then added to the upsampled feature map to enrich the feature representation, preserving the original semantic information during fusion. To further enhance the expression capability of the fused features, a modified C2f module is applied. This module efficiently extracts deep feature representations through cross-stage partial connections and feature fusion, enhancing the non-linear expressiveness while maintaining the model's lightweight nature. This provides high-quality feature support for the final detection head.

By integrating the channel attention mechanism and transposed convolution, the Multi-Scale Selective Fusion Pyramid Network achieves selective enhancement and efficient fusion of multi-scale features. This design is carefully optimized to meet the sensitivity requirements of detecting subtle pathological features in clubbed finger detection, improving detection performance while maintaining computational efficiency and lightweight characteristics.

3.4. Group Normalization Shared Convolution Detection Head (GNSCD)

To achieve a lightweight design, this study proposes the Group Normalization Shared Convolution Detection Head (GNSCD). In GNSCD, the convolutional layer is shared across all feature layers, using the same convolutional kernel and weights, as illustrated in Figure 4. This shared structure not only effectively reduces redundant parameters in the model but also enhances the consistency of features across different scales. By applying the same convolutional kernel to feature layers of varying scales, the model is compelled to learn consistent feature representations across these scales. This approach is particularly crucial for detecting targets, such as clubbed fingers, which possess similar morphologies but vary in size. The shared parameters enable the model to effectively capture common features across different scales, improving its target recognition capability and increasing detection accuracy.

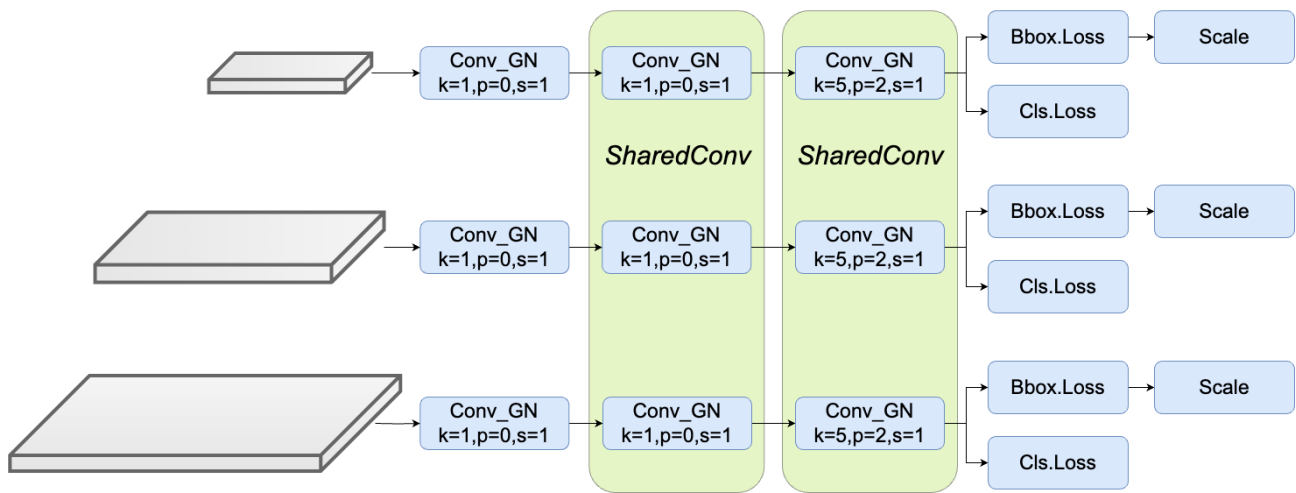


Figure 4. Structure diagram of the GNSCD module. Feature maps at different scales are first processed using group normalization convolution (Conv_GN), followed by shared convolution layers (with kernel sizes $k = 1$ and $k = 5$) to extract multi-scale features, thereby enhancing the model's ability to detect objects at different scales.

The GNSCD receives inputs in the form of multiple feature maps, denoted as $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_L\}$, where $\mathbf{X}_i \in \mathbb{R}^{C_i \times H_i \times W_i}$ represents the i th feature layer. To minimize redundancy in the convolutional layers, GNSCD adopts a shared convolution strategy, applying the same convolutional kernel and weights across multiple feature layers. In the conventional detection head, each feature layer has an independent convolutional layer, resulting in a parameter count of $\sum_{i=1}^L P_i$, where P_i is the parameter count for the i th feature layer. By employing shared convolution, the parameter count is reduced to a single P_{shared} , where $P_{\text{shared}} \ll \sum_{i=1}^L P_i$. The GNSCD network structure consists of several key components, which are detailed below.

Initially, each input feature layer $\mathbf{X}_i$ passes through a convolutional layer for channel adjustment. This layer uses a 1×1 convolutional kernel, taking C_i as the input channel count and producing C_{hid} (hidden channel count) as the output, ensuring that all feature layers have a unified channel count. This convolutional layer is followed by Group Normalization (GN) and the non-linear activation function SiLU (Sigmoid Linear Unit). This process is expressed as:

$$\mathbf{Y}_i = \text{SiLU}(\text{GN}(\text{Conv}_{1 \times 1}(\mathbf{X}_i))) \quad (12)$$

where $\text{Conv}_{1 \times 1}$ denotes the convolution operation with a kernel size of 1×1 , GN stands for the group normalization layer, and SiLU is the activation function.

Subsequently, the outputs $\mathbf{Y}_i$ from all feature layers are passed through the shared convolution module, ShareConv, which consists of two convolutional layers. The first shared convolution uses a 1×1 kernel, with both input and output channels set to C_{hid} . The second shared convolution employs a 5×5 kernel, also with input and output channels of C_{hid} . As with the previous steps, each convolutional layer is followed by GN and the SiLU activation function:

$$\mathbf{Z}_i^{(1)} = \text{SiLU}(\text{GN}(\text{Conv}_{1 \times 1}(\mathbf{Y}_i))) \quad (13)$$

$$\mathbf{Z}_i^{(2)} = \text{SiLU}(\text{GN}(\text{Conv}_{5 \times 5}(\mathbf{Z}_i^{(1)}))) \quad (14)$$

The shared convolution module is designed to expand the receptive field using larger convolutional kernels (e.g., 5×5) to capture more contextual information while simultaneously reducing the number of parameters through weight sharing, which enhances feature consistency across scales. This approach not only minimizes the storage requirements of the model but also boosts the efficiency of feature extraction, making the model more effective in processing multi-scale targets.

Following the output from the shared convolution layer, GNSCD applies Group Normalization (GN) to further normalize the feature maps' distribution. Compared to the conventional Batch Normalization (BN), GN offers superior stability and adaptability, particularly in small-batch training scenarios. The GN computation is defined as

$$\hat{\mathbf{Y}}_i^{(g)} = \frac{\mathbf{Y}_i^{(g)} - \mu_g}{\sqrt{\sigma_g^2 + \epsilon}} \gamma + \beta \quad (15)$$

where $\mathbf{Y}_i^{(g)}$ represents the features of the g th group; μ_g and σ_g^2 are the mean and variance of this group, respectively; ϵ is a small constant to prevent division by zero; and γ and β are learnable parameters for scaling and shifting.

The selection of GN is primarily due to its compatibility with small-batch training. Unlike BN, GN performs normalization based on the channel dimensions of the features and is independent of the batch size, thus maintaining stable performance even with small batches. This enhances the training efficiency and speeds up convergence. Moreover, GN normalizes features on a per-sample basis, eliminating dependencies between batches and promoting stable gradient flow.

By combining shared convolution layers with Group Normalization, GNSCD effectively balances model complexity and detection performance. The shared convolution layer reduces parameter redundancy through weight sharing, thereby improving feature consistency and enhancing multi-scale detection capabilities. Meanwhile, Group Normalization offers robust feature normalization during small-batch training, promoting effective gradient flow and improving both training efficiency and generalization. This design not only optimizes the use of computational resources but also ensures that the model remains efficient and accurate, even in resource-constrained environments.

3.5. EMASlideLoss

To further enhance the performance of the MSG-YOLO model, the EMASlideLoss function is introduced. This function combines the concepts of the Exponential Moving Average (EMA) and a sliding window to dynamically adjust loss weights during the training process, thereby improving the model's robustness and generalization capabilities. Traditional loss functions, such as cross-entropy loss or IoU loss, use fixed weights throughout training, which may not sufficiently adapt to the model's dynamic changes at different stages of training. To address this limitation, EMASlideLoss introduces a dynamic weight adjustment mechanism, enabling the loss function to adapt more effectively to these changes, thus improving detection performance.

In EMASlideLoss, the Exponential Moving Average (EMA) of the loss values is first computed at each training step. This smoothing method calculates a weighted average of past loss values, helping to stabilize fluctuations. The EMA calculation is given by

$$\text{EMA}_t = \alpha \cdot L_t + (1 - \alpha) \cdot \text{EMA}_{t-1} \quad (16)$$

where EMA_t represents the EMA value at time t , L_t is the current loss value, and α is the smoothing factor, typically ranging between 0 and 1. This parameter controls the influence of recent versus past loss values, allowing the model to react smoothly to changes over time.

To further stabilize and smooth the loss values, EMASlideLoss incorporates a sliding window average of the loss values. The sliding window method computes the average loss over a specified time period, enabling dynamic adjustment of the loss weight. The sliding window calculation is

$$\text{SW}_t = \frac{1}{N} \sum_{i=t-N+1}^t L_i \quad (17)$$

where SW_t is the sliding window average at time t , N is the window size (representing the number of past time steps considered), and L_i is the loss value at each time step i . This approach helps to smooth out short-term fluctuations and provides a baseline for dynamic adjustment.

By combining the EMA and sliding window techniques, EMASlideLoss can dynamically adjust the loss weight at each time step, allowing the model to adaptively modify the loss function based on the training stage. The process of dynamic weight adjustment is defined as

$$w_t = \frac{\text{EMA}_t}{\text{SW}_t} \quad (18)$$

$$\text{EMASlideLoss}_t = w_t \cdot L_t \quad (19)$$

where w_t is the dynamically computed weight and L_t is the current loss value. By adjusting the weight dynamically, EMASlideLoss effectively balances the influence of different loss values throughout training, enhancing the model's stability and generalization ability.

EMASlideLoss achieves adaptive balancing of loss values across different training phases through dynamic weight adjustment. This approach not only ensures better convergence during training but also optimizes the model's response to varying conditions, maintaining high detection accuracy and efficiency.

4. Experimental Detail

4.1. Dataset and Preprocessing

The dataset used in this study was provided by the Highland Medical Research Center of Qinghai University. Data collection involved 42 volunteers of varying genders and ages living at an altitude of 2400 m. The clinical diagnosis of clubbed fingers primarily relies on features such as nail appearance, nail-fold angles, and the presence of the Schamroth Sign. To clearly capture these diagnostic features, researchers used handheld mobile devices to photograph the lateral view of fingers under natural lighting conditions, resulting in a total of 207 images. Examples of the collected images are shown in Figure 5. To enhance the model's generalization capability and robustness, and to optimize its learning efficiency, data augmentation techniques such as random rotations and the addition of Gaussian noise were applied, increasing the total number of images to 495. This expanded dataset includes 225 images of clubbed fingers and 286 of normal fingers. Each image was annotated by experts from the Highland Medical Research Center at Qinghai University, ensuring high accuracy and consistency of the labels.

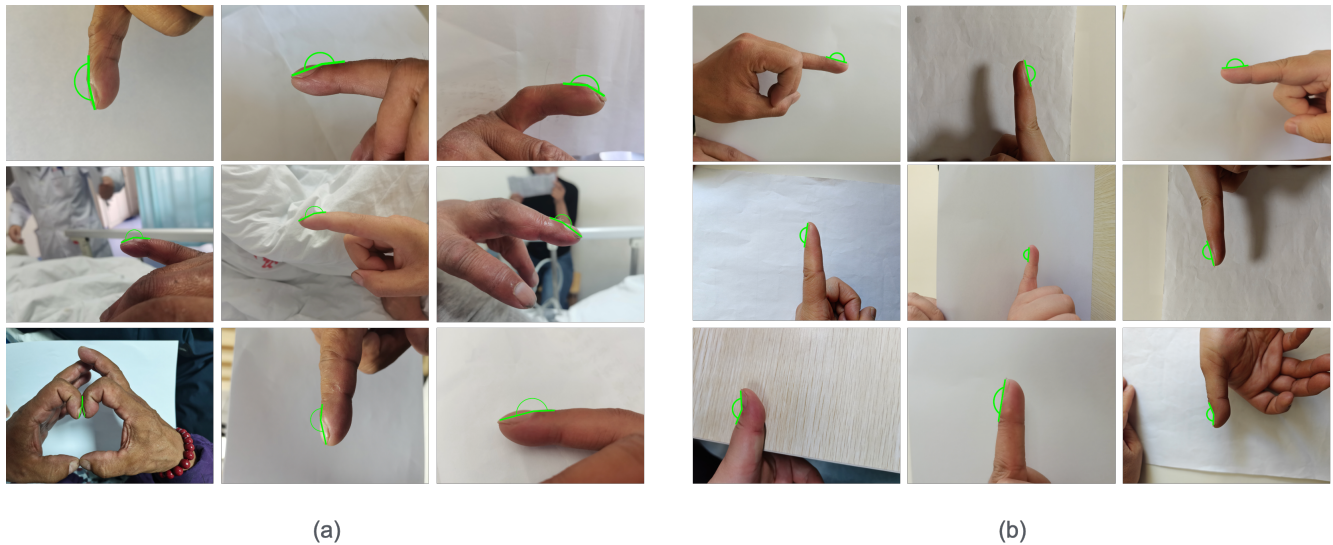


Figure 5. Dataset example: image (a) displays a clubbed finger, while image (b) shows a normal finger. Characteristics of clubbed fingers include nail-fold angles greater than 180° , whereas normal fingers generally have nail-fold angles less than 180° .

4.2. Experimental Environment

The operating system used in this study is Red Hat Enterprise Linux 7.6, with an Intel Xeon Gold 6348 processor, an NVIDIA A100 80GB PCIe GPU, and 251GB of RAM. The programming language is Python 3.9, and the deep learning framework is PyTorch 2.0.1, with GPU acceleration provided by CUDA 11.7. The experimental settings are as follows: input images are sized 640×640 , the batch size is set to 16, and the model is trained for 300 iterations, with the training and validation datasets split in an 80:20 ratio.

To ensure consistency and reproducibility, we adopted the default hyperparameter configuration of YOLOv8 as the baseline for our experiments. For fairness and comparability, the same hyperparameter settings were applied across all comparison models, thus minimizing the influence of differing hyperparameters on the results. The detailed training parameters are provided in Table 1.

Table 1. Training Parameter Settings.

Parameter	Value
Optimizer	SGD
Initial learning rate (lr0)	0.01
Final learning rate (lrf)	0.01
Weight decay	0.0005
Momentum	0.937
Warmup epochs	3
Warmup momentum	0.8
Close mosaic	10

This table summarizes the primary parameter settings used during the training process.

4.3. Evaluation Metrics

In this study, we evaluate the model's accuracy using Mean Average Precision (mAP), specifically mAP@0.5 and mAP@0.5:0.95. Additionally, we use the number of model parameters (Params) and computational complexity (FLOPs) as quantitative metrics to assess the lightweight nature of the model. These metrics provide a comprehensive and objective evaluation of the lightweight YOLO model's performance in the task of detecting clubbed fingers. Specifically, mAP@0.5 represents the mean average precision when the IoU threshold is set to 0.5, while mAP@0.5:0.95 calculates the mean average precision over a

range of IoU thresholds from 0.5 to 0.95, with an interval of 0.05. The number of parameters (Params) reflects the total count of parameters within the model, indicating the model's storage requirements. The computational complexity (FLOPs) measures the number of floating-point operations per second executed by the model and serves as an indicator of its computational demands. The formulas for calculating mAP are as follows:

$$P = \frac{TP}{TP + FP} \quad (20)$$

$$R = \frac{TP}{TP + FN} \quad (21)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (22)$$

4.4. Experimental Results and Analysis

4.4.1. Ablation Study

To validate the effectiveness of each module and strategy in the MSG-YOLO model for the clubbed finger detection task, we conducted a detailed ablation study. By sequentially removing or substituting key components, we analyzed the contributions and impacts of the Multi-Scale Dilation Residual (C2f-MDR) module, the Selective Feature Fusion Pyramid Network (SFFPN), the Group Normalization Shared Convolution Detection Head (GN-SCD), and the EMASlideLoss function on the overall model. These ablation experiments aimed to evaluate the independent performance contributions and synergistic effects of each component, providing a comprehensive understanding of the model design. The experimental results are presented in Table 2.

Table 2. The ablation study of the MSG-YOLO algorithm. In the table, Bolded content represents the best performance metric.

Model	C2f-MDR	SFFPN	GN-SCD	EMA-SlideLoss	mAP50	mAP50-95	FLOPs/G	Params/M
YOLOv8n					90.78%	76.36%	8.195	3.01
1	✓				91.28%	77.02%	8.1	3.01
2		✓			90.56%	75.67%	6.946	1.94
3			✓		89.02%	75.78%	7.169	2.4
4				✓	90.98%	77.96%	8.195	3.01
5	✓	✓			91.49%	76.58%	6.85	1.93
6	✓	✓	✓		93.07%	77.27%	6.214	1.6
Ours	✓	✓	✓	✓	93.64%	77.71%	6.214	1.6

As shown in Table 2, replacing the deep C2f module in the backbone with the C2f-MDR module increased mAP50 by 0.5% and mAP50-95 by 0.66%, while slightly reducing the computational cost to 8.1 GFLOPs, demonstrating that C2f-MDR effectively improves accuracy without increasing computation. Modifying the Neck to the SFFPN module reduced computational cost by 15.24% and the parameter count by 35.66%. Although mAP50 and mAP50-95 decreased by 0.22% and 0.69%, respectively, the model's efficiency and resource utilization were significantly optimized, making it suitable for low-resource environments. When the Head was replaced with GN-SCD, computational cost decreased by 12.5%, and the parameter count was reduced by 20.31%. While mAP50 dropped by 1.76%, mAP50-95 increased by 0.12%. GN-SCD's shared convolution structure effectively reduces parameters and computational load while maintaining multi-scale feature capture. Introducing EMASlideLoss improved mAP50 by 0.2% and mAP50-95 by 1.6% without changing the parameter count or computation. EMASlideLoss dynamically adjusts loss

weights, allowing the model to focus more effectively on critical features. Applying the C2f-MDR and SFFPN modules together increased mAP50 by 0.72% and mAP50-95 by 0.22%, while computational cost and parameter count dropped by 16.38% and 35.83%, respectively, showing their complementary effects in improving performance and reducing overhead. In Experiment 6, combining C2f-MDR, SFFPN, and GNSCD reduced the computational cost by 24.17% and parameter count by 46.89%. Meanwhile, mAP50 increased by 2.29% and mAP50-95 by 0.91%, demonstrating the significant synergistic effects of the modules in enhancing overall accuracy while minimizing computational demands.

Finally, adding C2f-MDR, SFFPN, and GNSCD modules and incorporating EMASlide-Loss reduced the computational cost and parameter count by 24.17% and 48.74%, respectively, while mAP50 increased by 2.86% and mAP50-95 by 1.35%. This multi-module design optimizes feature extraction, lightweight structure, and loss optimization strategies, demonstrating the MSG-YOLO model's effectiveness and efficiency for clubbed finger detection.

4.4.2. Comparative Experiments

To further validate the detection performance of the improved MSG-YOLO model proposed in this study, we conducted comparative experiments against several popular object detection algorithms. The models selected for comparison include Faster R-CNN, RTMDet-Tiny, TOOD, DDOD, and several versions of the YOLO series (YOLOv5n, YOLOv5s, YOLOv8n, YOLOv8s, YOLOv9t, YOLOv10n, and YOLOv11n). The comparative results are summarized in Table 3.

Table 3. Comparative results of various detection models.

Model Name	mAP@0.5	mAP@0.5:0.95	FLOPs (G)	Params (M)
Faster-Rcnn [37]	88.20%	64.60%	90.903	41.353
Rtmdet-Tiny [38]	86.10%	68.10%	8.026	4.873
TOOD [39]	85.20%	72.20%	78.857	32.021
DDOD [40]	87.00%	67.70%	71.146	32.199
YOLOv5n [20]	87.04%	73.96%	7.18	2.51 M
YOLOv5s [20]	88.77%	77.66%	64.36	25.07 M
YOLOv8n [23]	90.78%	76.36%	8.20	3.01 M
YOLOv8s [23]	92.34%	76.13%	28.65	11.14 M
YOLOv9t [24]	91.19%	76.37%	11.00	2.66 M
YOLOv10n [25]	89.77%	75.99%	8.40	2.71 M
YOLOv11n [26]	91.63%	75.33%	6.4	2.59 M
ours	93.64%	77.71%	6.214	1.6 M

As presented in Table 3, the improved MSG-YOLO model demonstrates significant advantages compared to current popular object detection algorithms. Compared with models like Faster R-CNN, RTMDet-Tiny, TOOD, DDOD, and the various YOLO versions, the proposed MSG-YOLO algorithm achieves the highest mAP@0.5 (93.64%) and mAP@0.5:0.95 (77.71%) with the smallest parameter count and computational load, highlighting its superior detection performance.

For instance, Faster R-CNN reaches an mAP@0.5 of 88.20% and mAP@0.5:0.95 of 64.60%, but its parameter count and computational load are 41.353M and 90.903G, respectively, which is significantly higher than MSG-YOLO, resulting in inferior overall performance. While RTMDet-Tiny has a relatively low parameter count (4.873M) and computational load (8.026G), its mAP@0.5 and mAP@0.5:0.95 are 86.10% and 68.10%, respectively—7.54% and 9.61% lower than MSG-YOLO's scores. Similarly, TOOD and DDOD do not surpass MSG-YOLO in accuracy and have higher parameter counts and computational loads, making them less suitable for resource-limited environments.

Although lightweight models such as YOLOv5n, YOLOv8n, YOLOv9t, YOLOv10n, and YOLOv11n have similar parameter sizes and computational complexity to MSG-YOLO, there is a gap in detection accuracy. Taking YOLOv8n as an example, its mAP@0.5 and mAP@0.5:0.95 are 90.78% and 76.36%, respectively, still lower than MSG-YOLO. YOLOv11n has an mAP@0.5 of 91.64%, which is 2.00% lower than MSG-YOLO, and an mAP@0.5:0.95 of 75.33%, which is 2.38% lower than MSG-YOLO. In addition, the parameter size and computational complexity of these models are still higher than those of MSG-YOLO, and their overall performance is inferior to MSG-YOLO.

While YOLOv5s and YOLOv8s achieve detection accuracies close to or slightly higher than some other models in the comparison, their parameter counts and computational loads are substantially higher. YOLOv5s has 25.07 M parameters and 64.36 G in computational load; YOLOv8s has 11.14 M parameters and 28.65 G in computational load, both of which are significantly higher than MSG-YOLO's values, making them unsuitable for resource-constrained environments. The MSG-YOLO model excels across various metrics, outperforming mainstream object detection algorithms in detection accuracy while optimizing resource requirements, making it suitable for deployment on devices with limited computational resources, thus meeting both real-time and high-accuracy needs.

4.5. Visualization Analysis

To assess the performance of the improved MSG-YOLO model in the clubbed finger detection task, we present a visual comparison with the YOLOv8n baseline model in Figure 6 to validate MSG-YOLO's accuracy and robustness. Figure 6a illustrates the detection results of MSG-YOLO, while Figure 6b shows the results from YOLOv8n. It is evident from Figure 6 that MSG-YOLO outperforms the baseline model in distinguishing between clubbed and normal fingers under various conditions, demonstrating higher confidence scores and greater stability.

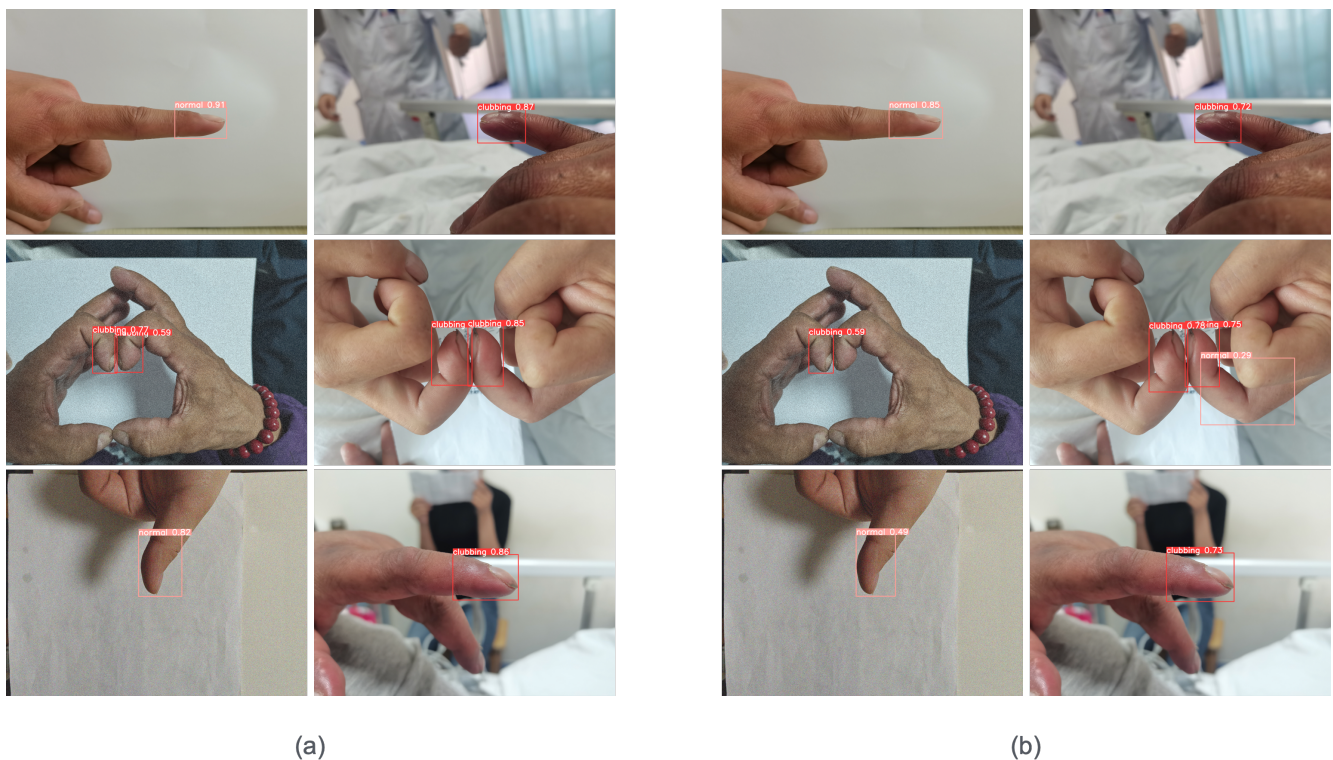


Figure 6. Comparison of detection results between MSG-YOLO and YOLOv8n models. (a) Performance of the MSG-YOLO model in the clubbed finger detection task. (b) Results from the YOLOv8n model on the same task.

Specifically, MSG-YOLO maintains high confidence levels even in complex backgrounds, accurately distinguishing the two classes. This indicates that the model's sensitivity and accuracy in identifying subtle features such as nail curvature and thickness have significantly improved. In contrast, the YOLOv8n model, shown in Figure 6b, displays lower confidence levels under the same conditions and occasionally misclassifies the fingers. These results highlight how MSG-YOLO, through its incorporation of the Multi-Scale Dilation Residual (C2f-MDR) module and the Selective Feature Fusion mechanism, significantly enhances detection stability and accuracy in diverse scenarios.

To further evaluate MSG-YOLO's performance in object detection, we analyzed several images and visualized the detection outcomes using heatmaps. The heatmap employs a color gradient to illustrate the model's attention levels across pixels: red areas indicate the highest target presence probability, yellow and green areas indicate moderate focus, and blue areas represent the least attention.

Figure 7 clearly show that the model consistently focuses on the fingertip region across all images, indicating that the fingertip serves as a prominent feature. The model accurately identifies and localizes the fingertip, with the red areas representing high-confidence regions where the probability of detecting the target is highest. This consistent observation across all images underscores the model's efficiency and reliability in detecting this particular feature. The gradient changes in the heatmap further reveal differences in attention: red regions are concentrated at the fingertip and its boundary with the background, while the rest of the finger and background mostly appear in blue or green. This demonstrates the model's capability to effectively differentiate the target from the background and focus on key features. The variation in attention suggests that the model possesses strong capabilities in handling complex backgrounds and multi-object detection scenarios.

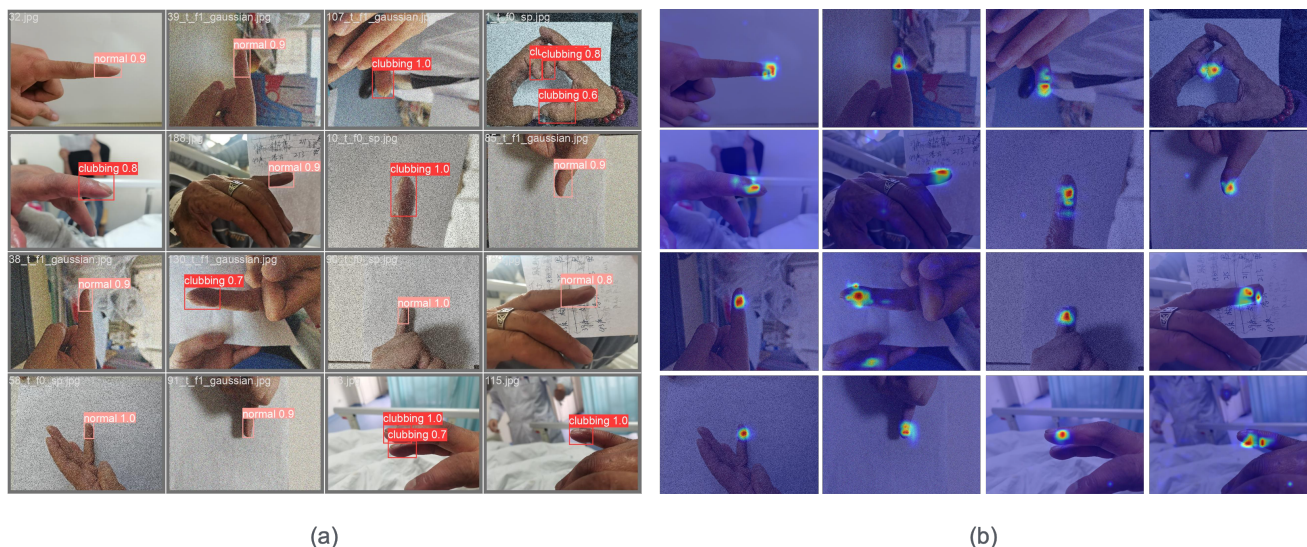


Figure 7. Detection Figure results of MSG-YOLO on different samples. The left side (a) shows the bounding boxes for normal and clubbed finger samples, with each box labeling the detected category and confidence score. The right side (b) presents the corresponding heatmaps, highlighting the areas of focus for the model during detection. The high-intensity red and yellow regions indicate significant features that the model has identified in these areas.

Overall, the MSG-YOLO model is capable of accurately distinguishing between clubbed and normal fingers in most images, with high confidence scores. The model shows exceptional performance in clubbed finger detection, surpassing other baseline models not only in detection accuracy and confidence but also in its ability to handle multiple objects and resist noise interference.

5. Conclusions

This paper presents MSG-YOLO, a lightweight model for finger clubbing detection based on YOLOv8, designed to address the limitations of existing methods in terms of accuracy, computational efficiency, and resource requirements. By incorporating the multi-scale dilated residual module (C2f_MDR), the lightweight selective feature fusion pyramid network (SFFPN), and the group normalization shared parameter detection head (GNSCD), MSG-YOLO significantly enhances both detection accuracy and efficiency. Experimental results show that MSG-YOLO reduces the number of parameters by 48.74% and the computational complexity by 24.17%, while increasing mAP@0.5 by 2.86% to 93.64%. It significantly outperforms YOLOv8n in terms of detection accuracy while maintaining a minimal computational overhead. This demonstrates MSG-YOLO's ability to balance high precision with efficiency in automated finger clubbing detection, making it highly suitable for deployment in resource-constrained environments, such as mobile devices and embedded systems.

While the MSG-YOLO model has achieved significant success in finger clubbing detection, there remain areas for further improvement. First, the current dataset's diversity and scale limit the model's training effectiveness and generalization capability. Future research should focus on developing larger and more diverse datasets that cover a wider range of case characteristics, particularly finger clubbing images from different ethnicities, genders, ages, and pathological conditions, to enhance the model's robustness and adaptability. Furthermore, finger clubbing detection relies not only on image data but could also benefit from the integration of other clinical information, such as patient medical history and related symptoms. The fusion of multi-modal data would further improve the accuracy and stability of detection. Future studies could consider combining image data with clinical data to create a more comprehensive and accurate automated diagnostic system, enhancing the model's practical value in real-world clinical environments. In conclusion, MSG-YOLO provides an effective solution for the automated detection of finger clubbing, achieving a significant improvement in detection accuracy while reducing computational complexity. With the expansion of datasets, further algorithmic refinements, and the integration of multi-modal technologies, the automation and intelligence of finger clubbing detection are expected to continue improving.

Author Contributions: Conceptualization, Z.W., Q.M. and F.T.; methodology, Z.W. and Y.Q.; software, Z.W. and B.L.; validation, Z.W., X.L. (Xin Liu) and S.K.; data curation, Z.W. and Y.Q.; writing—original draft preparation, Z.W. and X.L. (Xin Li); writing—review and editing, Z.W. and Q.M.; visualization, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Qinghai Province, grant number 2023-ZJ-989Q.

Data Availability Statement: The data used in this study are available upon request from the corresponding author via email. Data are contained within this article.

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

References

1. Ritter, E.; Itach, T.; Paran, D.; Gaskin, A.; Havakuk, O.; Ablin, J.N. Cardiac Sarcoma Mimicking Libman–Sacks Endocarditis in a Patient with Systemic Lupus Erythematosus (SLE): A Case Report and Literature Review. *J. Clin. Med.* **2024**, *13*, 4345. [\[CrossRef\]](#)
2. Arshad, M.; Athar, Z.M.; Hiba, T. Current and Novel Treatment Modalities of Idiopathic Pulmonary Fibrosis. *Cureus* **2024**, *16*, e56140. [\[CrossRef\]](#)
3. Burcovschii, S.; Aboeed, A. Nail Clubbing. In *StatPearls*; StatPearls Publishing: Treasure Island, FL, USA, 2024.
4. Arnal, C.; Richert, B. Examination of the nails: Main signs. *Hand Surg. Rehabil.* **2024**, *43*, 101639. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Rutherford, J.D. Digital clubbing. *Circulation* **2013**, *127*, 1997–1999. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Goldsmith, L.A.; Freedberg, I.M.; Eisen, A.Z.; Wolff, K.; Goldsmith, L.A.; Katz, S. *Fitzpatrick's Dermatology in General Medicine*; McGraw Hill Professional: New York, NY, USA, 2003.
7. Azad, R.; Kazerouni, A.; Heidari, M.; Aghdam, E.K.; Molaei, A.; Jia, Y.; Jose, A.; Roy, R.; Merhof, D. Advances in medical image analysis with vision transformers: A comprehensive review. *Med. Image Anal.* **2024**, *91*, 103000. [\[CrossRef\]](#) [\[PubMed\]](#)

8. Zhou, S.K.; Greenspan, H.; Davatzikos, C.; Duncan, J.S.; Van Ginneken, B.; Madabhushi, A.; Prince, J.L.; Rueckert, D.; Summers, R.M. A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises. *Proc. IEEE* **2021**, *109*, 820–838. [CrossRef] [PubMed]
9. Abdulhadi, J.; Al-Dujaili, A.; Humaidi, A.J.; Fadhel, M.A.R. Human nail diseases classification based on transfer learning. *ICIC Express Lett.* **2021**, *15*, 1271–1282.
10. Karunarathne, H.; Senarath, G.; Pathirana, K.; Samarawickrama, H.; Walgampaya, N. Nail Abnormalities Detection and Prediction System. In Proceedings of the 2023 5th IEEE International Conference on Advancements in Computing (ICAC), Colombo, Sri Lanka, 7–8 December 2023; pp. 394–399.
11. Soğukkuyu, D.Y.C.; Ata, O. Classification of melanonychia, Beau’s lines, and nail clubbing based on nail images and transfer learning techniques. *PEERJ Comput. Sci.* **2023**, *9*, e1533. [CrossRef]
12. Pathan, S.K.; Jatoth, S.; Narisetty, P.; Pulari, S.V.; Vadithya, A. Nail Insight: Enhanced Nail Image Analysis for Early Disease Detection. In Proceedings of the 2024 5th IEEE International Conference for Emerging Technology (INCET), Belgaum, Karnataka, India, 24–26 May 2024; pp. 1–9.
13. Hsu, W.S.; Liu, G.T.; Chen, S.J.; Wei, S.Y.; Wang, W.H. An Automated Clubbed Fingers Detection System Based on YOLOv8 and U-Net: A Tool for Early Prediction of Lung and Cardiovascular Diseases. *Diagnostics* **2024**, *14*, 2234. [CrossRef]
14. Hittawe, M.M.; Harrou, F.; Togou, M.A.; Sun, Y.; Knio, O. Time-series weather prediction in the Red sea using ensemble transformers. *Appl. Soft Comput.* **2024**, *164*, 111926. [CrossRef]
15. Harrou, F.; Zeroual, A.; Hittawe, M.; Sun, Y. Chapter 6—Recurrent and convolutional neural networks for traffic management. In *Road Traffic Modeling and Management*; Harrou, F., Zeroual, A., Hittawe, M.M., Sun, Y., Eds.; Elsevier: Amsterdam, The Netherlands, 2022; pp. 197–246.
16. Redmon, J. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
17. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
18. Redmon, J. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
19. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
20. Jocher, G. YOLOv5 by Ultralytics. Available online: <https://github.com/ultralytics/yolov5> (accessed on 17 November 2024).
21. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. *arXiv* **2022**, arXiv:2209.02976. [CrossRef]
22. 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. *arXiv* **2022**, arXiv:2207.02696. [CrossRef]
23. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLO. Available online: <https://github.com/ultralytics/ultralytics/tree/v8.2.103> (accessed on 17 November 2024).
24. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv* **2024**, arXiv:2402.13616. [CrossRef]
25. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *arXiv* **2024**, arXiv:2405.14458. [CrossRef]
26. Jocher, G.; Qiu, J.; Chaurasia, A. Ultralytics YOLO. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 17 November 2024).
27. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y.; et al. PP-YOLOE: An evolved version of YOLO. *arXiv* **2022**, arXiv:2203.16250.
28. Zhang, Y.; Zhang, H.; Huang, Q.; Han, Y.; Zhao, M. DsP-YOLO: An anchor-free network with DsPAN for small object detection of multiscale defects. *Expert Syst. Appl.* **2024**, *241*, 122669. [CrossRef]
29. Zhou, Y. A YOLO-NL object detector for real-time detection. *Expert Syst. Appl.* **2024**, *238*, 122256. [CrossRef]
30. Wang, D.; Peng, J.; Lan, S.; Fan, W. CTDD-YOLO: A Lightweight Detection Algorithm for Tiny Defects on Tile Surfaces. *Electronics* **2024**, *13*, 3931. [CrossRef]
31. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
32. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
33. Połap, D.; Jaszcz, A. Sonar digital twin layer via multi-attention networks with feature transfer. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 4206910. [CrossRef]
34. Li, H.; Zhang, R.; Pan, Y.; Ren, J.; Shen, F. Lr-fpn: Enhancing remote sensing object detection with location refined feature pyramid network. *arXiv* **2024**, arXiv:2404.01614.
35. Wu, Y.; He, K. Group Normalization. In Proceedings of the European Conference on Computer Vision (ECCV 2018), Munich, Germany, 8–14 September 2018; pp. 3–19.
36. Ioffe, S. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
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] [PubMed]

38. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Zhang, S.; Chen, K. RtmDET: An empirical study of designing real-time object detectors. *arXiv* **2022**, arXiv:2212.07784.
39. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. Tood: Task-aligned one-stage object detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; IEEE Computer Society: Washington, DC, USA, 2021; pp. 3490–3499.
40. Chen, Z.; Yang, C.; Li, Q.; Zhao, F.; Zha, Z.J.; Wu, F. Disentangle your dense object detector. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual Event, 20–24 October 2021; pp. 4939–4948.

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.