International Journal on Semantic Web and Information Systems
Volume 20 • Issue 1

YOLO-DCNet:
A Semantic-Based Novel Flexible Lightweight Human Detection Algorithm

Yiheng Wu, China University of Mining and Technology, China*
Jiaqiang Dong, Xichang University, China
Jianxin Chen, China University of Mining and Technology, China

ABSTRACT

Enhanced processors empower edge devices like smartphones for human detection, yet their application is constrained by algorithmic efficiency and precision. This paper introduces YOLO-DCNet, a lightweight neural network detector built upon YOLOv7-tiny. Incorporating a dynamic multi-head structural re-parameterization (DMSR) module within its backbone network enables effective processing of the features utilized in the model. To improve multi-scale feature aggregation, the model integrates a channel information compression and linear mapping (CLM) module into its feature pyramid architecture. Moreover, the optimization of training and inference performance is achieved by employing RepVGG blocks between the main computational modules of the model. Experimental data reveal that the enhanced YOLOv7-tiny model achieves a 31.7% faster inference speed and marginal gains of 0.7% in mAP@0.5 and 0.5% in mAP@0.5:0.95 over the original. This underscores the model's improved performance and applicability for real-time human detection on edge devices across diverse applications.

KEYWORDS

Channel Information Process, Lightweight Neural Network, Personnel Detection, Reparameterization Module

The development of microprocessors and related software frameworks has significantly improved the performance of smartphones and embedded platforms, enabling them to process various data generated by their sensors (Krichen, 2021; Biglari, 2023). This has led to increased attention toward edge computing, which reduces network bandwidth consumption and latency and protects user privacy (Su et al., 2022). Concurrently, significant advancements have been made in the technologies related to image retrieval (Chu et al., 2022; H. Wang et al., 2020; D. Li et al., 2019; Nhi et al., 2022), generation (Qian et al., 2022; Chopra et al., 2022), encryption (Yu et al., 2018), as well as classification and recognition (Alsmirat et al., 2019; Zheng et al., 2022; Mandle et al., 2022; X. Wang et al., 2023; X. Li et al., 2023; Sun et al., 2023). For instance, within the realm of aviation and transportation system imagery applications, relevant data sets (Behera et al., 2023) and models (Jain et al., 2022) continue to be proposed by researchers. To enhance the application of emerging technologies, researchers have

DOI: 10.4018/IJSWIS.339000

*Corresponding Author

achieved valuable results in specific application scenarios, such as healthcare (Hunt et al., 2021), video surveillance systems (Patrikar et al., 2022), and agriculture (Krichen, 2021; Ji et al., 2022), by deploying carefully designed algorithms for edge platforms. Object detection algorithms play a crucial role in recognizing, analyzing, and processing human image features in various life-related fields (Ye et al., 2022; Xiao et al., 2021; Islam et al., 2020).

When designing object detection algorithms, it is important to consider specific requirements for key performance indicators such as classification accuracy, bounding box precision, and running speed in different application scenarios (Zaidi et al., 2022). To enhance performance and minimize manual effort, a design based on deep neural networks should be considered over traditional machine learning (ML) algorithms that necessitate manual feature extraction (Alzubaidi et al., 2021). As research has progressed, more complex network model structures and algorithmic approaches have been proposed. Pioneering works in staged detection, anchor box strategies, and multi-scale feature fusion have improved the reliability of detection algorithms in real-world applications in terms of precision, recall, parameter count, and inference speed (Kaur et al., 2022).

Deep detectors can be categorized as one-stage or two-stage detection algorithms based on the number of stages involved in the computation process. Two-stage detectors, such as Faster-RCNN (Ren et al., 2017), typically involve region proposals in the first stage and perform further detection based on these proposals. In contrast, one-stage algorithms like You Only Look Once (YOLO) (Jiang et al., 2022) and its derivatives can obtain object detection boxes and corresponding categories with a single regression, resulting in improved performance in terms of inference speed and computational efficiency (Xiao et al., 2021). Therefore, one-stage algorithms are more suitable for edge devices with limited computing power and tight resource allocation requirements.

Among the algorithms that meet the aforementioned criteria, the tiny variant within the YOLOv7 (C. Wang et al., 2022) object detection model family stands out by further reducing model size and computational demands while maintaining high detection performance, making it suitable for deployment on devices with limited computational resources. The YOLOv7-tiny, inheriting structures like ELAN from the series, demonstrates commendable precision and inference speed on publicly available data sets. Nevertheless, there is still room for improvement when it comes to edge deployment, which is the focus of this paper's work.

This study introduces YOLO-DCNet as a lightweight one-stage detector specifically designed for devices with limited computing capabilities. The model incorporates innovative computational modules. In the backbone network, a dynamic multi-head structural re-parameterization (DMSR) module is adopted to ensure high efficiency during both training and inference. By replacing the channel information compression and linear mapping (CLM) module into the feature pyramid, the model improves the aggregation capability for multi-resolution features and semantic information, while reducing computational cost. Experiments are conducted utilizing the publicly available PASCAL VOC data set, where multiple algorithms and schemes are evaluated. The experimental results suggest that YOLO-DCNet is the optimal selection when considering both precision and inference speed in task environments.

RELATED WORK

Ghost Module

In object detection using deep neural networks, the preprocessed image data is initially fed into the backbone network of the model for feature extraction. To improve the efficiency of this process, researchers have made significant advancements in exploring lightweight backbones. Han et al. (2020) visualized feature maps in convolutional neural networks and observed that there were redundant feature maps. They discovered that these redundant feature maps could be obtained through relatively cheaper operations, such as linear transformations or depth-wise separable convolutions. In the processing of their proposed ghost module, as depicted in Figure 1, a small number of convolutional kernels are initially employed to extract features from the input feature map, as illustrated by the brightly colored portion in the figure. Subsequently, the obtained features undergo additional cheaper computations, represented by the symbol Φ in the figure, such as linear transformations or depth-wise separable convolutions, resulting in the output shown in the grayscale portion of the figure pointed to by the arrows. Finally, the features obtained from these two processes are concatenated to generate the final output feature map. In their experiments, the Ghost-VGG-16 model, which incorporated this design, achieved the highest accuracy (93.7%) compared to other models, with a substantial reduction in floating point operations (FLOPs).

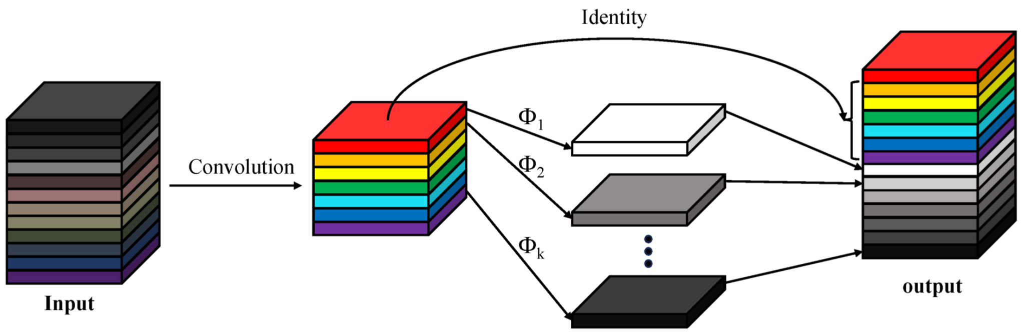
Building upon this approach, Deng et al. (2022) applied the ghost module to the classic single-stage object detection algorithm, reducing computational costs and improving detection performance. Similarly, Kong et al. (2022) reduced the model size to 1/6 of its original size and improved mean average precision (mAP) by 2.9%. To enhance the detection speed of the model, Cao et al. (2022) introduced the GhostNet module, and their proposed model, GhostNet-YOLOv5, achieved a 4.83% increase in precision. Considering memory and computational limitations of mobile devices, Han et al. (2022) proposed the CPU-efficient Ghost (C-Ghost) module and GPU-efficient Ghost (G-Ghost), which better balance accuracy and speed in models deployed on heterogeneous devices, including CPU and GPU. Furthermore, to address challenges posed by variations in target shapes, occlusions, and complex backgrounds in practical production applications, researchers (Qiu et al., 2022; H. Wang et al., 2022; Wei et al., 2022) implemented efficient processing of redundant information in channel features based on the ghost module. In their respective domains, these detectors exhibited significant improvements in terms of recall, precision, and mAP.

Notwithstanding the incremental advances in computer vision, the integration of efficient memory access cost (MAC) design with multi-scale edge detection methodologies has practical value, particularly within the ambit of the head part of edge detectors—a critical juncture for optimizing performance across scales. In light of this research lacuna, this paper presents a sophisticated design innovation for the head part of the model, which deftly incorporates the principles of GhostNet. This integration is meticulously crafted to process information from feature maps at varying scales with superior efficacy.

Structural Re-Parameterization Technique

In addition to effectively handling specific feature maps, researchers have also focused on refining the branch structure of network models. Ding et al. (2021) proposed RepVGG, which utilizes a structural re-parameterization technique. Its effect on the network structure is shown in Figure 2. This technique enables the network to retain its learning capabilities during the training phase through a multi-branch

Figure 1. An illustration of the ghost module. Φ represents the inexpensive operation



structure and, subsequently, it integrates the weights from these branches into a single convolutional layer with a 3x3 kernel size for deployment. Furthermore, the batch normalization layer's coefficients are re-engineered into the kernel and bias of a 1x1 convolutional layer, which permits the execution of convolutional computations. By applying zero padding within the 1x1 convolutional layer, the kernel is effectively transformed into an equivalent 3x3 kernel, thereby seamlessly transitioning into a 3x3 convolution. This ingenious structural transformation is pivotal in enabling the model to deliver highly efficient inference performance on graphics processing units.

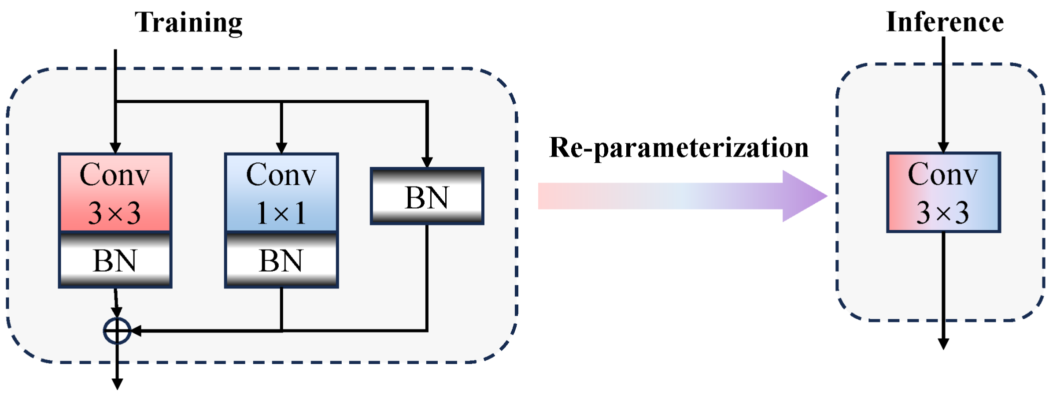
Feng et al. (2021) combined RepVGG with the spatial attention (SA) mechanism and achieved an accuracy of 95.10% on their test set. Qing et al. (2021) improved the detection performance of small objects from different angles by utilizing four target detection scales with a structurally adjusted and re-parameterized backbone network. Wan et al. (2023) applied re-parameterization techniques and bilinear feature extraction and fusion in the field of animal husbandry, enhancing the extraction of facial features and achieving recognition rates of 97.64% and 99.43% for front and full-face datasets, respectively. Yang et al. (2023) aimed to improve inference speed while maintaining model accuracy by using larger kernel sizes in the backbone network and an improved re-parameterization design in the head network, resulting in 97% mAP and 270 frames per second (FPS) on their data set. Additionally, researchers (Qi et al., 2021) have employed the design philosophy of RepVGG to simplify complexity and improve real-time detection in auxiliary perception applications, achieving high detection rates for objects in various ship compartments.

Although re-parameterization has evidenced progress in a range of applications, its potential in navigating the complexities of elaborate branch structures and the nuances of lightweight scenarios remains underexplored. In response to this research gap, the current study innovates with a flexible multi-branch, multi-head structural re-parameterization module and judiciously integrates RepConv blocks, reminiscent of those in RepVGG, into strategic locations within the model. This tailored approach yields improved performance metrics for person detection, thereby illuminating the necessity of extending research endeavors that harness RepConv-like methodologies to advance the frontiers of lightweight detector.

Cross-Scale Connection

During the process of feature extraction in neural networks, downsampling modules enhance higher-level semantic information in the feature maps. However, effectively combining semantic information from different resolutions and intensities obtained during this process is a challenge beyond the design of the model backbone. To address this challenge, Wu et al. (2021) introduced spatial pyramid pooling (SPP) into the model, which improved the model's recognition capability in detection and increased

Figure 2. Network structure transformation via structural re-parameterization in RepVGG block



the inference speed by four times. Liang et al. (2022) established a feature pyramid network (FPN) for a lightweight anchor-free algorithm, with prediction heads for regression and classification, resulting in excellent precision performance. To enhance the detection of small and medium-sized objects, Zhang et al. (2022) introduced a coordinate attention module into the FPN structure, which reduced parameter scale while maintaining accuracy. Yao et al. (2023) incorporated efficient channel attention (ECA) blocks into the cascade feature fusion module (CFFM) to improve detection precision and fuse shallow detailed information with deep high-level semantic information. This approach resulted in a 2.95% increase in algorithm accuracy. Ruan et al. (2023) applied the SimAM (Yang et al., 2021) mechanism for spatial and channel-level feature fusion in a multi-scale feature fusion structure, achieving a fast inference speed of 69.3 FPS. Zong et al. (2021) proposed RCNet, a network that simplified the bidirectional pyramid inference architecture and achieved an average precision (AP) of 50.5 with a single-model single-scale on the COCO test set. Zhu et al. (2022) introduced an improved feature pyramid network (ImFPN), which better adapted to objects of varying sizes by fusing different features based on similarity, resulting in a 1.7% increase in AP on COCO.

In this investigation, the conceived detector is elegantly engineered to judiciously manage redundant features through operations that boast of diminished computational demands, thus striking a harmonious balance between the dual imperatives of learning efficiency and deployment performance. Complementing this, the model judiciously assimilates a structural re-parameterization block, meticulously calibrated to augment the detector's precision. The pivotal integration of cross-scale connection techniques, which amalgamate these innovative elements, empowers the model with enhanced adaptability, tailored to the specificities of the experimental task scene.

YOLO Algorithm

The YOLO series algorithms are renowned for their compact size and high-speed processing, placing them among the forefront of single-stage algorithms. These algorithms directly or indirectly regress the categories, sizes, and positions of target bounding boxes after processing the preprocessed image data once through the neural network. This efficiency is one of the key reasons for their popularity. Among them, YOLOv3 introduced both residual structures and the FPN structures, leading to a significant improvement in the detection performance of multi-scale objects, which has been highly valued by researchers (Mandle et al., 2022). YOLOv4 introduced noteworthy innovations such as spatial pyramid pooling (SPP), the mish activation function, and mosaic data augmentation, resulting in an approximate 10% enhancement in both mAP and FPS. YOLOv5 (Jocher, 2021) achieved impressive levels of accuracy while simultaneously reducing computational resources and storage space, showcasing its exceptional flexibility and efficiency. Moreover, YOLOv7 (C. Wang et al., 2022) introduced scaling strategies based on concatenation, the ELAN module, RepConv, and other techniques, leading to a 75% reduction in parameters, a 36% decrease in computation, and a 1.5% increase in AP. In the latest advancements, research on YOLOv8 (Ultralytics, 2023) by Ultralytics has yielded considerable improvements in tasks such as object detection, image segmentation, and pose estimation through the integration of the C2F module, varifocal loss (VFL), and task-aligned assigner—each constituting both efficient and innovative approaches to enhance the model's performance.

Moreover, researchers have developed various detectors for specific application scenarios by incorporating advantageous new designs into the YOLO series framework. For example, Chen et al. (2021) proposed YOLOv5-lite, a more lightweight version suitable for deployment on embedded platforms, by incorporating more efficient backbones. Derivative algorithms of the YOLO series have also been effective in domains such as bone fracture detection (Y. Li et al., 2022) and drones (Ma et al., 2018).

The YOLO series algorithms have made significant progress in recent years. However, they typically require dense convolutional operations across the entire image, which can result in longer inference times and higher energy consumption on resource-constrained edge computing platforms. Even the tiny versions of these algorithms often undergo algorithmic optimizations, including improvements to the model structure, to address this scenario. By incorporating the ghost operation, re-parameterization, and enhanced cross-scale connection discussed in the previous sections, performance improvements have been achieved in person detection tasks within the model.

IMPLEMENTATION OF THE NETWORK

Approach in the Research

This paper introduces YOLO-DCNet, an enhanced version derived from the YOLOv7-tiny framework, with a specific focus on improving performance on edge platforms. The backbone of YOLO-DCNet incorporates DMSR modules, which enhance feature extraction through multiple branches during training and are integrated into streamlined and efficient modules for inference. The head of the model includes a channel information CLM module, which efficiently aggregates input feature information. Additionally, the RepConv block is applied in the FPN to analyze complex content with different scale features and varying semantic intensity.

YOLO-DCNet achieves higher precision compared to the lightweight model YOLOv7-tiny, with mAP@0.5 and mAP@0.5:0.95 increasing by 0.7% and 0.5% respectively. Additionally, YOLO-DCNet achieves a 31.7% increase in inference efficiency. The main contributions of this work encompass:

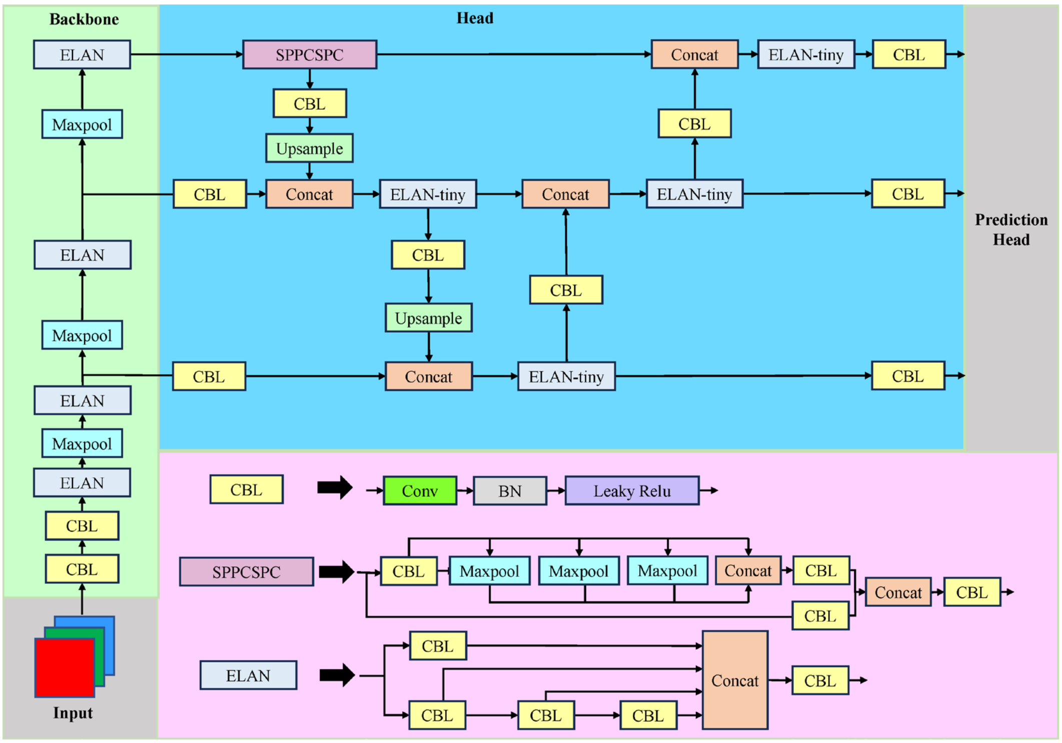
1) The proposal of the DMSR module and the CLM module, which effectively extract meaningful information from input images, as supported by experimental results.
2) The design of a compact backbone network and a streamlined FPN structure that demonstrate superior performance with increased inference efficiency, as evidenced by experimental results.
3) The demonstrated adaptability of YOLO-DCNet, as proposed in this paper, to human target detection in scenarios with limited computing capabilities, as verified through experiments.

Network Architecture of YOLO-DCNet

Clearly, among the various models of YOLOv7 with different scales, the compact "tiny" version is the primary candidate for lightweight deployment. Its model structure follows the same framework as other models in the YOLO series, but it adopts a more streamlined design in the main functional modules such as the feature extraction module, head structure, and training head usage strategy. The structure of the network and its principal modules are depicted in Figure 3. The CBL module, composed of convolution, batch normalization, and Leaky ReLU activation function, serves as an efficient computational unit. Within this tiny model, the SPPCSPC module is derived from a simplified version that utilizes pyramid pooling operations and the CSP structure, while still maintaining an adequate number of branches. Additionally, a more compact ELAN is used in the network structure instead of E-ELAN, resulting in better and more stable performance with fewer parameters. Additionally, a simpler maxpool layer is employed to connect the various main computational modules in the backbone, replacing the MP structure. By incorporating these advanced compact designs, YOLOv7-tiny reduces the parameter count by 39% and the computational load by 49% compared to YOLOv4-tiny-31, while maintaining a similar AP performance.

Although the reliability and efficiency of YOLOv7-tiny have been experimentally demonstrated, additional optimization is necessary when deploying detection on edge computing platforms with limited hardware capabilities. This paper conducts research based on the overall design concept of YOLOv7-tiny and the importance of detection accuracy, inference speed, and model size. Various methods such as designing computational modules, organizing data sets, optimizing parameters, and training models are employed. And on this basis, YOLO-DCNet is proposed as an efficient algorithm for person detection on edge devices (Figure 4) by considering research analysis and results. As shown in the figure, the CBL part remains consistent with YOLOv7-tiny, while the SPP module incorporates CBL at both ends and utilizes cascaded maxpool for multi-scale feature extraction. The extracted features are then aggregated through concatenation.

Figure 3. The structure of the YOLOv7-tiny



The YOLO-DCNet model structure consists of three main parts: the backbone, head, and prediction head. In the backbone section, responsible for feature extraction, the main computational module utilized is the DMSR module proposed in this paper. For example, with an input size of 640×640, the preprocessed image undergoes initial processing through two consecutive RepConv blocks, resulting in a feature map of size 160. This feature map is then processed alternately by the DMSR module and RepConv blocks. The features produced by the end of the backbone are further extracted using spatial pyramid pooling and fused with features from different scales in the FPN in the head section. The resulting information is then input into the prediction head section through three branches of different resolutions. Each branch's features are processed to obtain feature maps of size 20×20, 40×40, and 80×80, respectively, each representing class probabilities and position information. Finally, post-processing techniques like non-maximum suppression are applied to generate the final bounding boxes that correspond to the detected categories.

DMSR Module

The RepVGG block used in this work is processed as depicted in Figure 5. As illustrated in the figure, this re-parameterization technique encompasses three distinct branching scenarios. In this progressive process, batch normalization (BN) is transformed into a 1x1 convolution, while the 1x1 convolution can be further transformed into a 3x3 convolution. Finally, the structure is consolidated into an efficient 3x3 convolution.

Each branch of this block contains classic convolution layers and BN layers, which can be integrated using the structural re-parameterization technique. The equations (1), (2), and (3) involved in this process can be described as follows: x represents the input feature, the function weight represents the convolution operation performed by using the weights of the convolutional kernel on the input

Figure 4. The structure of the YOLO-DCNet where DMSR module is the equal-scale sampling unit and CLM module is another type of primary computing unit

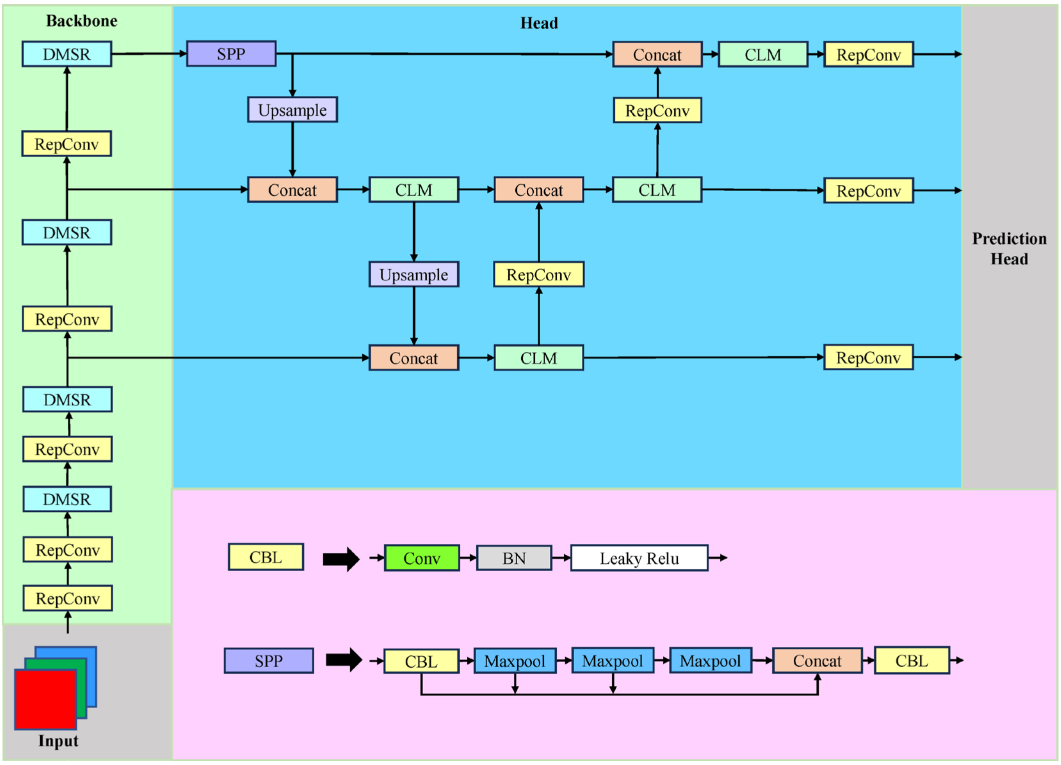
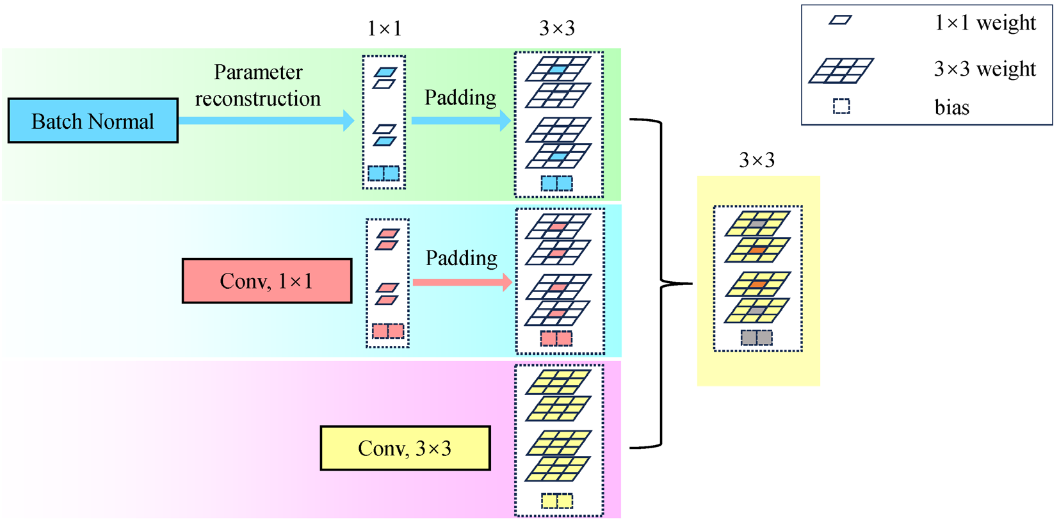


Figure 5. Process of structural re-parameterization of the RepVGG block



feature, and bias represents the bias term of the convolution. Additionally, in the context of batch normalization calculations, γ denotes the scale factor, β represents the bias coefficient, σ represents the standard deviation of the features, and μ_x represents the mean of the features. The traditional convolution computation formula is given by equation (1), where the input feature x is convolved with the weight of the convolutional kernel and then added to a bias term to generate the convolution output. By substituting and reconstructing the components in equation (2), which represents the basic computation formula of the BN layer, it is possible to transform them into parameters of an equivalent 1×1 convolution layer based on the convolution principle. In practical implementation, by substituting equations (1) and (2), the fusion of convolution weights and BN layer parameters can be achieved, resulting in the reconstructed parameters of a new convolution layer, as depicted in equation (3). Furthermore, when the stride and feature map size remain unchanged after convolution, the 1×1 convolution kernel can be padded with surrounding zeros to obtain a new 3×3 convolution kernel that produces equivalent results. By flexibly utilizing these methods, the various network layers in the branches of the RepConv block can be transformed into convolutions with the same kernel size. By directly integrating their weights and biases according to the distributive law of multiplication, a single, computationally equivalent 3×3 convolution is obtained.

$$Conv\left(x\right)=weight\left(x\right)+bias \tag{1}$$

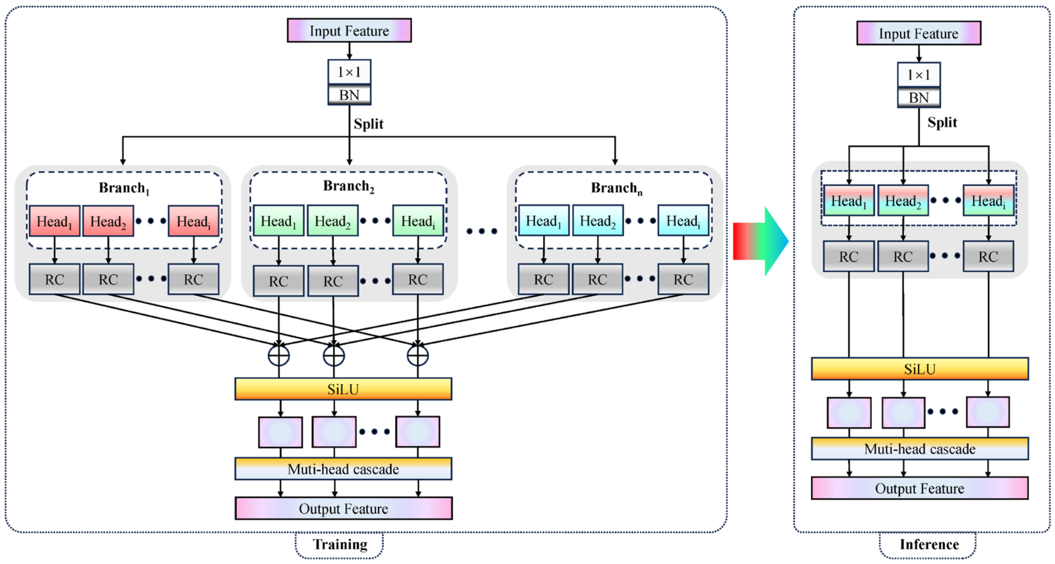
$$BN\left(x\right)=\gamma\times\frac{\left(x-\mu_x\right)}{\sigma}+\beta=\frac{\gamma}{\sigma}\times x+\left(\beta-\frac{\gamma}{\sigma}\times\mu_x\right) \tag{2}$$

$$BN\left(Conv\left(x\right)\right)=\frac{\gamma}{\sigma}\times weight\left(x\right)+\left(\beta-\frac{\gamma}{\sigma}\times\mu_x\right) \tag{3}$$

Based on this principle, the DMSR module is designed, as illustrated in Figure 6. From the figure, it can be observed that the input feature is divided into multiple heads upon entering the module, and each head is further divided into multiple branches along the channel using the same partitioning method. Each branch undergoes re-parameterization through RepConv (RC), and is then re-aggregated based on the head index. Subsequently, the output is obtained by passing through SiLU activation and multi-head cascade, as shown in the figure.

Building upon the aforementioned principles and formulations, the equations pertinent to the DMSR module are articulated as follows:

Figure 6. Structure and process of the DMSR module



$$Conv_{DMSR-Head_i}\left(x\right) = Conv_{Branch_1}\left(x\right) + Conv_{Branch_2}\left(x\right) + \ldots + Conv_{Branch_n}\left(x\right) \tag{4}$$

$$\begin{aligned} &Concat\left(SiLU\left(Conv_1\left(x\right)\right), SiLU(Conv_2\left(x\right)), \ldots SiLU\left(Conv_n\left(x\right)\right)\right) = \\ &\quad SiLU\left(Concat(Conv_1\left(x\right), Conv_2\left(x\right), \ldots Conv_n\left(x\right)\right)) \end{aligned} \tag{5}$$

$$\begin{aligned} &DMSR_{main-proc}\left(x\right) = \\ &SiLU\left(Concat\left(Conv_{DMSR-Head_1}\left(x\right), \ldots, Conv_{DMSR-Head_i}\left(x\right)\right)\right) \end{aligned} \tag{6}$$

In this module, a multi-branch, multi-head structure is introduced to increase the complexity of its architecture and enhance the feature analysis capabilities during training by incorporating more parameters. Internally, the input feature map is processed by different branches and, within each branch, the features are further divided into multiple heads for re-parameterized convolution along the channel dimension, as shown in equation (4). The computation results are added at their respective positions, activated using the SiLU function, and then concatenated via the pointwise operation to obtain the final output. The procedure is refined by resequencing the operations to stack initially and then proceed to activation, as delineated in equation (5). This reorganization facilitates the derivation of equation (6), which articulates the configuration of the entire module, with x representing the input features before being split into individual branches.

It is worth noting that, to improve the model's performance on the specific tasks addressed in this study, the DMSR module is dynamically designed based on its position within the backbone. The specific parameters pertaining to the branches and heads within the network architecture are delineated in Table 1. The index of DMSR module in this tabulation is predicated upon the model's structural topological sequence, describing the order in which features flow through the network structure. Additionally, the structure containing the 1×1 convolution in the RepConv block and the branch with only BN are also integrated into the DMSR module flexibly to enrich its structure and feature extraction capabilities, although they are not included in the statistical table.

The integration of these meticulously crafted DMSR modules has been instrumental in enhancing the comprehensive performance of the algorithm. During the backward propagation process, where the network learns parameters, the branch structure facilitates the learning of distinct feature representations along divergent paths, concentrating on capturing various aspects of the input data, thereby enriching the feature representation and augmenting the learning efficacy of the module. Upon deployment and practical application, the resultant cheap operations, transformed from the original structure, inherit exceptional feature extraction capabilities while simultaneously accelerating inference speed and reducing memory consumption. These characteristics enable the lightweight model to seamlessly transition between training and inference without the necessity for retraining. This dual capability of smooth transitioning, devoid of additional training, bestows upon the model a practical flexibility in deployment.

CLM Module

To efficiently and appropriately handle the fused complex information, the design of the processing module incorporated the concept of the ghost module, which involves dividing the features into two parts and simplifying the treatment of the redundant part. Based on this approach, the Gr-Ghost block is designed to efficiently process the features. After the features enter this block, as shown in Figure

Table 1. The branches and heads corresponding to different DMSR modules

Index of DMSR Module	B1	B2	B3	B4
Number of branches	2	2	4	2
Number of heads	1	2	4	8

7, they are first processed by a convolutional layer. The output of the convolutional layer is then split into two branches for further processing. In one branch, the features are sequentially subjected to BN and the SiLU activation function. In the other branch, they undergo grouped convolution and BN. The results from both branches are aggregated by concatenation to obtain the output of this block.

Equations (7) aptly encapsulate the principal computational processes inherent in the Gr-Ghost block, where x represents the input features:

$$GrGhost\left(x\right) = Concat\left(BN\left(Conv_{5\times5}\left(Conv_{1\times1}\left(x\right)\right)\right), SiLU\left(BN\left(Conv_{1\times1}\left(x\right)\right)\right)\right) \tag{7}$$

In the Gr-Ghost block, the input features are initially compressed along the channel dimension using a 1×1 kernel convolutional layer, resulting in an output with C channels. One of the outputs goes through BN and SiLU processing, while the other undergoes connected group convolution and BN. The results from these two branches are then concatenated in a channel-wise manner to obtain the final output of this module. During this process, the group convolution has a kernel size of 5×5, a stride of 1, and the number of groups is equal to the input channels, which is also C. This enables the convolutional kernel of this branch to have a larger receptive field and, during inference, the BN and convolution are merged into a more "cost-effective" grouped linear operation. Considering the rationale behind the design of the Gr-Ghost block, this study proposes the CLM module, which is applied to the head part of the model to handle the complex information from aggregated shallow and deep features. The structure of the CLM module is depicted in Figure 8. As depicted in the figure, this module can be primarily divided into four blocks, consisting of two 1×1 blocks composed of convolutional layers, BN layers, and SiLU activation, as well as two Gr-Ghost blocks mentioned earlier.

Equation (8) correspondingly delineates the process, wherein $x_{\mathrm{Re}\,p1\times1}$ denotes the output following the initial processing of the input x by the 1×1 block.

$$CLM_{main-proc}\left(x\right) = Concat\left(x_{1\times1}, GrGhost_1\left(x_{1\times1}\right), GrGhost_2\left(GrGhost_1\left(x_{1\times1}\right)\right)\right) \tag{8}$$

In the CLM module, when features are inputted, they are scaled along the channel dimension using pointwise convolution, followed by BN and SiLU activation to obtain the output. In the subsequent two stages, one branch of the output is efficiently processed by cascaded Gr-Ghost blocks, resulting in the generation of an output with stacked redundant features. Finally, the feature maps from each

Figure 7. Structure of the Gr-ghost block

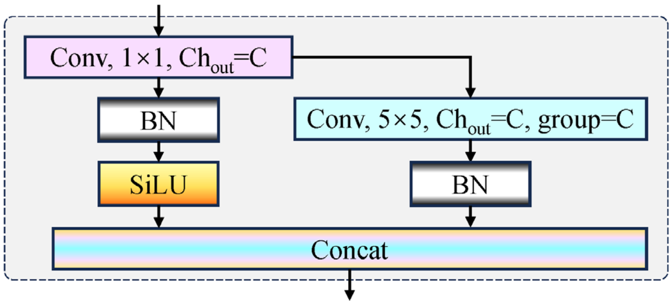
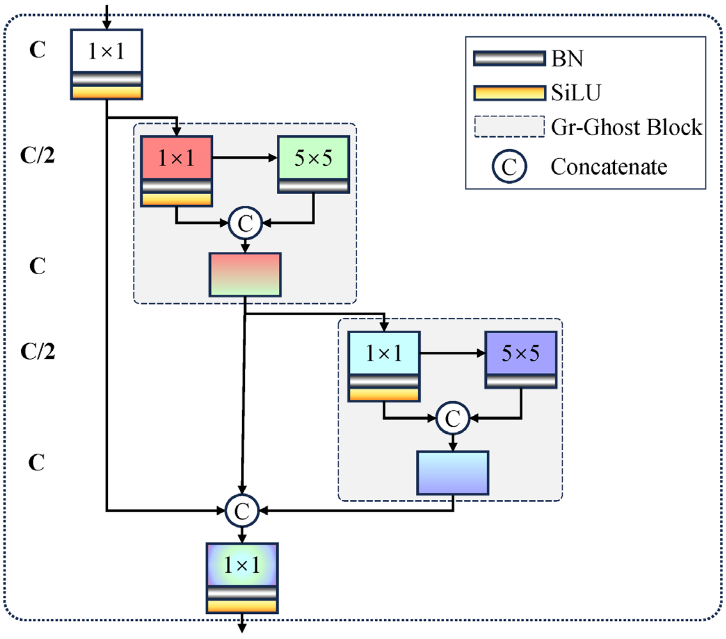


Figure 8. Structure of the CLM module



stage undergo compression through a 1×1 convolution layer and the activated outputs are obtained, similar to the previous stage. Using the one-shot aggregation design, the model becomes more efficient in terms of MAC performance (Ma et al., 2018; Lee et al., 2019), and this module draws inspiration from such a design. Additionally, by introducing the Gr-Ghost block and appropriately compressing the channels in the CLM module, it becomes feasible to efficiently refine robust elements from the feature information from the backbone with reduced computational cost.

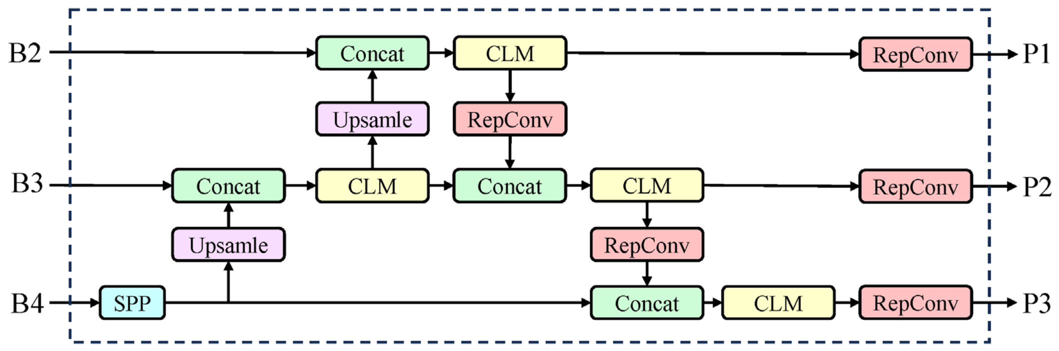
The CLM module constitutes a lightweight neural network architecture that, by leveraging the Gr-Ghost block, significantly reduces parameters and computational load while preserving requisite performance levels, thereby accelerating inference speed. Furthermore, the cascaded and interleaved configuration of computational units enhances the training process by enabling more effective gradient signal transmission, which in turn facilitates improved updates of the underlying features. These advancements allow depth networks incorporating the CLM to more effectively learn and represent intricate features, resulting in a model that is both lightweight and efficient and making it particularly suitable for resource-constrained devices and applications.

The Designed Backbone Network and FPN

In the backbone network, the DMSR module mentioned earlier is utilized to replace the ELAN module at the corresponding position. This replacement aims to enhance the information extraction capability of the module during training, improve its speed during inference, and increase its applicability when deployed on target platforms. Furthermore, the RepConv block, which incorporates structural re-parameterization, is employed to replace the original CBL block and maxpool layer. This substitution enhances the ability of these connecting modules to analyze and transmit features without significantly increasing the computational cost.

Additionally, in the head part of the model, an FPN is constructed using both bottom-up and top-down approaches, as depicted in Figure 9. In this structure, the inputs B2, B3, and B4 correspond to the indices of the DMSR modules as indicated in Table 1 and, at the end of this structure, outputs P1,

Figure 9. Structure of the designed FPN



P2, and P3 are generated. Similarly, within the FPN structure, the CLM module replaces the ELAN module. Furthermore, certain CBL blocks are removed, and the remaining blocks are substituted to establish connection with RepConv blocks.

The lightweight designs incorporated in the improved model make it more suitable for deployment and application on resource-constrained devices. These designs ensure that the model's size undergoes only limited changes, thereby minimizing the impact on storage and transmission costs. Additionally, they enable faster execution of object detection tasks under similar hardware conditions, enhancing real-time performance while maintaining high detection accuracy. In summary, the reduced model size and increased inference speed make it particularly well-suited for utilization in resource-constrained edge computing scenarios and similar environments.

EXPERIMENTAL DESIGN

The data used in this study is sourced from publicly available datas ets that have been extensively tested by various researchers to validate the effectiveness of the modules for person detection tasks. Comparative experiments were conducted to evaluate the performance of YOLO-DCNet against several state-of-the-art object detection algorithms known for their excellence in this task, thereby verifying the efficiency of the proposed model. In addition, different combinations of computational units were designed and scientifically evaluated to assess their performance, rationality, and effectiveness in various configurations.

Experimental Conditions

The experimental setup for this study is as follows. Pre-training weights are not utilized during the training process. The model is trained for 200 epochs, with the first 3 epochs serving as warm-up training. Stochastic gradient descent (SGD) optimizer is employed with an initial learning rate of 1x10-2. The momentum and weight decay parameters are set to 0.937 and 5x10-4, respectively. Image pre-processing includes random translation, horizontal flip, HSV color transformation, and mosaic augmentation. During testing, a confidence threshold of 1x10-2 and an IOU threshold of 0.5 are set to evaluate the accuracy of object detection. These settings ensure reliable detection results.

The experimental computations were conducted on a computer running Windows 10, equipped with an Intel i7 12700KF CPU and 32GB of memory. The main hardware unit utilized was an NVIDIA GeForce RTX 3090 with 24GB of VRAM. The system was configured with CUDA 11.1 and cuDNN 8.0.5. The deep learning framework PyTorch 1.9.0, based on Python 3.9, was employed for the experiments.

Data Set

The data set utilized in the experiments is sourced from the publicly available PASCAL VOC data set, which is well-suited for the human detection task. This data set comprises 5,240 images that include a total of 13,319 human targets, each annotated with ground truth bounding boxes. The data set encompasses individuals from various age groups, including the elderly, young, and children, as well as diverse scenarios such as riding, being at home, and resting.

To mimic real-world scenarios, where detectors typically encounter similar types of data, a letterbox process, similar to that of YOLOv5, is applied to each image in the data set. This process scales the image according to the original ratio and pads it with additional pixels along the shorter dimension, resulting in a square data region sized at 640x640. The corresponding ground truth boxes undergo the same size transformation. Additionally, the ratio of the object box's longest side to the image's side length is calculated and statistically analyzed. The size statistics of the ground truth box within the 640×640 processed image are presented in Table 2.

Observing the table, it becomes apparent that the majority of the data set used in this experiment concentrates within the small to medium size range. This alignment with typical scenarios encountered by the task is crucial. In order to efficiently utilize this data set, it is divided into a validation set and a test set, each consisting of 1,048 images, while the remaining images are dedicated to model training.

Evaluation Criteria

Similar to other object detection tasks, precision and recall are employed as performance metrics in this study to assess the model's effectiveness, and they are calculated using formulas (9) and (10). The calculation of these metrics involves three fundamental terms: true positive (TP), false positive (FP), and false negative (FN). Precision denotes the ratio of correctly detected targets to all positive predictions, encompassing both correct and incorrect predictions. On the other hand, recall represents the ratio of correctly detected instances among all positive instances. In the application scenario of the study, the optimization of these factors holds clear practical significance for improving model performance: Higher precision indicates fewer errors in the predicted results, while higher recall implies a reduced number of missed person objects during detection.

$$Precision = \frac{TP}{FP + TP} \tag{9}$$

$$Recall = \frac{TP}{FN + TP} \tag{10}$$

By visualizing precision and recall on a coordinate system, the precision-recall (P-R) curve can be generated. The area under this curve, referred to as AP, serves as a comprehensive evaluation of the model's object detection performance. In the specific experiment, which focuses solely on the detection of human targets, the mean mAP is equivalent to AP. The formula for calculating AP is displayed in formula (11).

$$AP = \int_0^1 P\left(R\right) dR \tag{11}$$

Table 2. The size statistics of ground truth box in 640×640 image after processing

Proportion of the longest edge of box (%)	0~20	20~40	40~60	60~80	80~100
Box quantity ratio (%)	35.3	26.7	22.1	11.5	4.4

Throughout the computational process, the selection of confidence thresholds and intersection over union (IOU) values exerts a significant influence on the outcome of the calculations. Moreover, given the practical deployment scenarios of the task, parameters such as the model size, as represented by the number of parameters, giga floating point operations per second (GFLOPs) as a measure of computational complexity, latency time, and FPS emerge as critical benchmarks. Notably, the number of parameters serves as an indicator of the model's size, while GFLOPs denote the count of floating-point operations required for a single forward pass of the model. Latency time, which refers to the duration needed for the model to make a prediction on a single input, alongside the corresponding FPS, provides a more intuitive reflection of the model's operational efficiency.

Comparison With State-of-the-Art Models

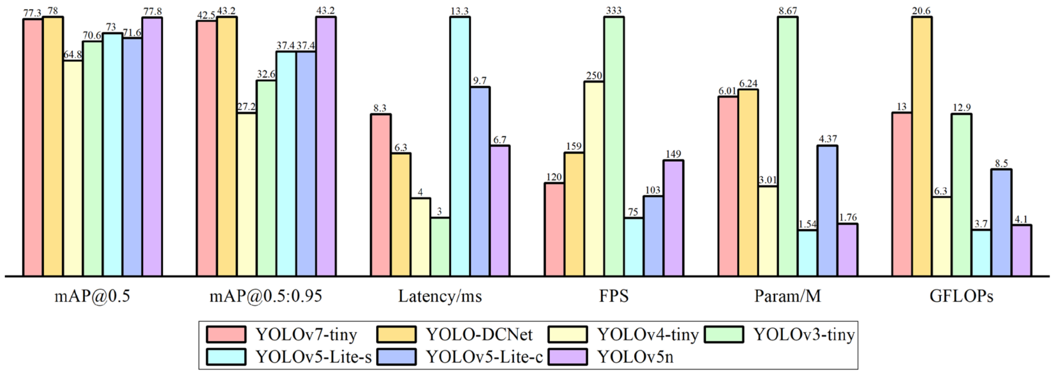
An intuitive approach to showcase the research accomplishments in this paper is to compare the results with outstanding models that have achieved impressive performance in this task at the present stage. Therefore, YOLO-DCNet, constructed based on the YOLO framework, was evaluated alongside one-stage detectors that are considered state-of-the-art in related fields, all under the same conditions. The results of these comparisons are depicted in Figure 10 and demonstrate several key evaluation metrics, including mAP, inference efficiency, model size, and GLOPs.

Upon examination of the results, it can be deduced that YOLO-DCNet outperformed other models in terms of mAP@0.5 and mAP@0.5:0.95. Specifically, YOLO-DCNet achieved mAP@0.5 and mAP@0.5:0.95 scores of 78% and 43.2% respectively, on this data set, demonstrating significant enhancements compared to YOLOv4-tiny. While maintaining superiority in precision, YOLO-DCNet also exhibited reduced latency time and increased the number of frames processed per second compared to YOLOv7-tiny, yielding optimizations of 2 ms and 39 FPS respectively. Furthermore, when considering model size, YOLO-DCNet proved to be smaller than YOLOv3-tiny. Considering the specific application of this model, particularly in scenarios with limited computing power, YOLO-DCNet, with its advantages of higher precision and inference efficiency, stands as a primary candidate for implementation.

Ablation Experiments

To empirically validate the impact of the various computation units mentioned earlier on the model's training, inference, and precision performance, a series of ablation experiments were conducted, encompassing different combinations of optimization schemes as presented in Table 3. The DC-R scheme involved replacing the CBL block and maxpool layer with re-parameterization convolution in the backbone, compared to YOLOv7-tiny. Conversely, the DC-D scheme only replaced the ELAN module with the proposed DMSR module. Meanwhile, the DC-RD scheme incorporated both

Figure 10. Comparison of different object detection algorithms on the test set



modifications from the previous schemes and further simplified some CBL blocks in the head section. Moreover, the DC-RD-FPN scheme incorporated the modifications from DC-RD and introduced the CLM module to replace the main computation module in FPN. Similarly, the DC-RD-FPN-RC scheme, in addition to the modifications of DC-RD-FPN, replaced the CBL blocks used for connections in FPN with the RepConv blocks.

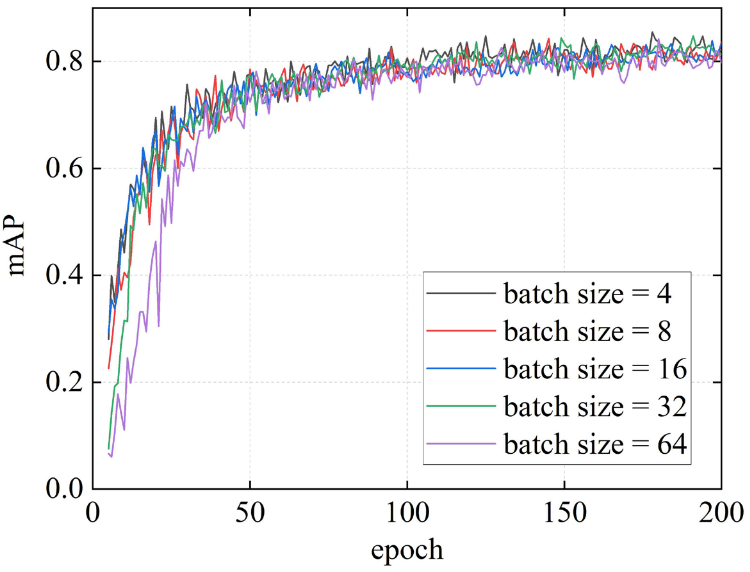
Before comparing the different schemes, the DC-RD-FPN-RC scheme, which encompasses all the improvement methods, was trained using various batch sizes on the data set. The resulting mAP-epoch curve for different schemes is illustrated in Figure 11. Analyzing the curve, it is evident that regardless of the number of images used in each batch, the trend of mAP demonstrates an initial rapid increase, followed by a gradual slowdown around the halfway point of the total epochs, eventually reaching a nearly stable state. Among these scenarios, the batch size of 16 achieved a relatively high mAP value, surpassing 83%. Comprehensive analysis reveals that the chosen batch size in the experiment strikes a reasonable balance between precision and training efficiency.

Additionally, it is essential to analyze the stability of different combinations of computation units during training. In order to accomplish this objective, we generated the recall-epoch curves, depicted in Figure 12, to visualize the training progress of each model corresponding to different schemes.

Table 3. Different improvement schemes

Scheme	BackBone-RC	DMSR	New-FPN	CLM	FPN-RC
DC-R	√				
DC-D		√			
DC-RD	√	√			
DC-RD-FPN	√	√	√		
DC-RD-FPN-C	√	√	√	√	
DC-RD-FPN-RC	√	√	√	√	√

Figure 11. The MAP-epoch curves of scheme DC-RD-FPN-RC with different batch sizes

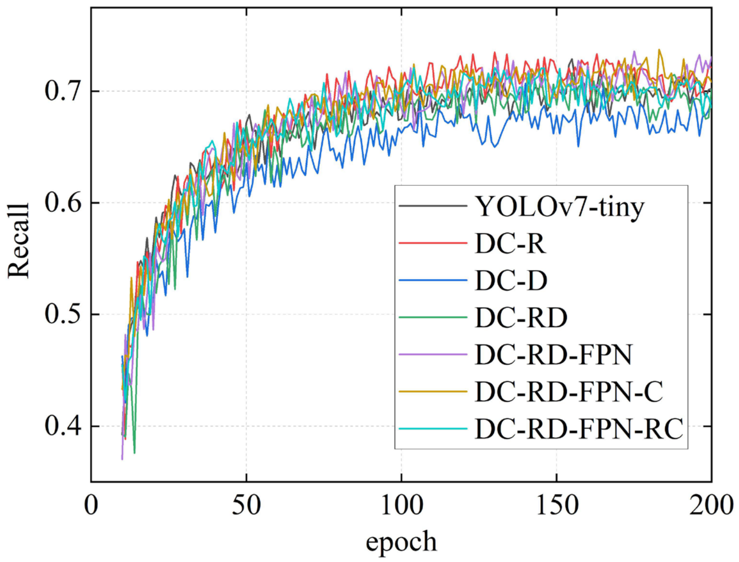


The trend observed in the recall values for each scheme is quite similar: a rapid increase followed by a leveling off after 100 epochs. This indicates that the learning performance of each scheme's model, when compared to YOLOv7-tiny, has not been significantly negatively impacted. Among the schemes, the DC-RD-FPN-RC scheme achieves a recall of 72.1% during this process, indicating its ability to minimize missed detections during practical task execution. From the above observations, we can infer that the DMSR, CLM, and other modules utilized in the ablation experiments effectively control the model's learning performance.

The experimental results of various optimization schemes for this task are shown in Figure 13, which visually presents the performance of different schemes across multiple evaluation criteria. From these results, we draw the following conclusions: Although the DC-R scheme, which directly uses re-parameterized convolution in the backbone, shows a certain improvement in precision, it introduces more parameters and computational complexity, resulting in a 0.5 ms increase in inference time. On the other hand, the DC-G scheme, which introduces the DMSR module in the backbone, exhibits the opposite performance in all aspects, with a slight decrease in mAP@0.5. This indicates the need for further improvement in feature compression and fusion between computational modules in the backbone. The DC-RD scheme achieves a balance between the aforementioned schemes, increasing FPS by 12, but there is still room for improvement in terms of precision. Furthermore, the DC-RD-FPN scheme removes some CBL blocks in FPN, introducing more computational complexity to enhance mAP performance by adjusting the channel numbers. Through the introduction of the CLM module in the DC-RD-FPN-C scheme, effective multi-scale feature aggregation is achieved with reduced computational costs. Compared to the original YOLOv7-tiny, this model achieves a 0.7% improvement in mAP@0.5, a 0.5% improvement in mAP@0.5:0.95, and optimizes latency time and FPS by approximately 31.7%. Building upon this, the DC-RD-FPN-RC scheme, which involves the RepConv block in FPN, further increases mAP@0.5 by 0.2%, achieving the highest value of 78% among all schemes.

Considering the context of edge computing platforms, such as personal smartphones, which usually have limited computational capabilities and where the reliability of carried algorithms is crucial to users, it is logical to select the DC-RD-FPN-RC scheme as the final YOLO-DCNet. This

Figure 12. Recall-epoch curves of different improvement schemes



scheme demonstrates the best precision performance while significantly improving inference speed compared to YOLOv7-tiny.

Furthermore, in this study, feature visualization was performed to compare the DMSR module and CLM module in YOLO-DCNet with the ELAN module present in the backbone and head of YOLOv7-tiny, as illustrated in Figure 14. The visualization results of the main computational units' output feature maps at the same positions in the backbone and feature pyramid of the two models are represented by B1~B4 and P1~P3, respectively. In this context, B1~B4 correspond to the topological order of the backbone network, while P1~P3 can be referenced to the corresponding positions of FPN output in Figure 9. Observing the feature maps generated at different positions in the backbone, it becomes evident that the DMSR module efficiently separates target features (represented by bright colors) from the background (represented by dark colors) compared to its counterpart which aggregates more nontarget information. Similarly, in the feature pyramid structure, the CLM module effectively distinguishes features from the background, and the feature map in the highest fusion level, P3, contains minimal redundant information from the environment. Through this comparative analysis, the effectiveness of each trained computational unit proposed in this paper for processing personnel targets is proven, demonstrating the rationale behind the model used in YOLO-DCNet.

Based on the experimental results presented, YOLO-DCNet outperforms the existing YOLOv7-tiny model in terms of mAP at IoU thresholds of 0.5 and 0.5:0.95, with improvements of 0.7% and 0.5%, respectively. Furthermore, despite a minimal increase in the number of parameters, YOLO-DCNet exhibits significant enhancements in inference speed, achieving a latency optimization of 2 ms and a frame rate boost of 39 FPS. The visualization experiments also effectively illustrate the

Figure 13. Ablation results of different schemes on the test set

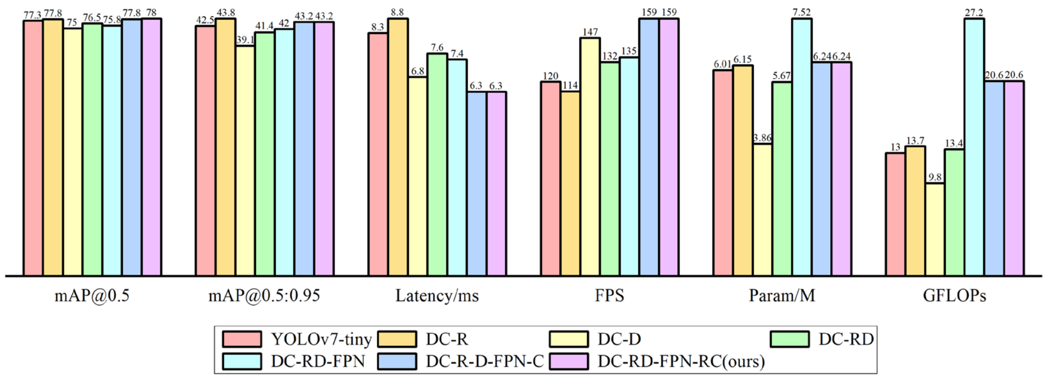
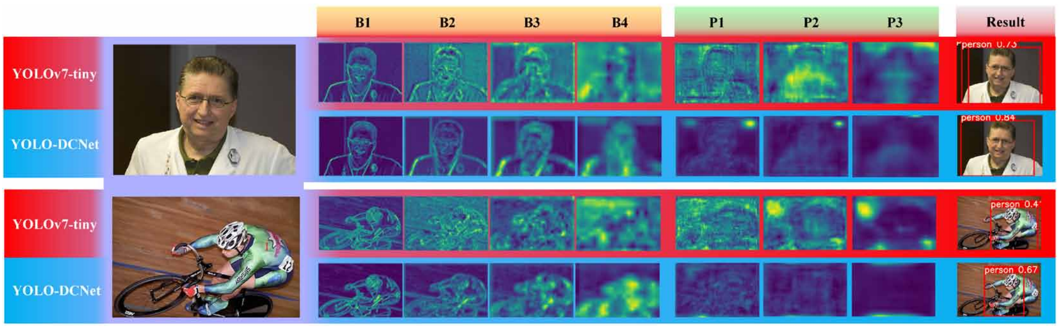


Figure 14. The feature visualization of part of the network layer in YOLOv7-tiny and YOLO-DCNet



superior feature extraction capabilities of YOLO-DCNet's innovative structure. Consequently, the potential application domains of YOLO-DCNet can be broadened to include edge computing devices. For example, when operating at an equivalent image processing rate, YOLO-DCNet can provide a performance buffer for additional functionalities such as object counting, tracking, and traffic analysis, while requiring fewer computational resources. From a hardware design standpoint, this could also ameliorate the thermal conditions of processors, thereby further optimizing the hardware configuration and layout. Considering these attributes, YOLO-DCNet is demonstrated to be an algorithm well-suited for deployment on edge devices.

CONCLUSION

This research employs a lightweight module-based single-stage design approach and optimization strategy to propose an algorithm for detecting personnel at the edge. The proposed model exhibits excellent precision performance while significantly enhancing inference efficiency. The proposed detector enhances the applicability of personnel detection within edge computing, particularly on platforms that necessitate solutions for power consumption, thermal management, and reliability issues. For instance, within smartphones or embedded development boards, it aids in human localization, thereby furnishing information for higher-level functionalities such as photography enhancement and portrait beautification. Furthermore, its potential applications are extendable across various domains, including serving as a frontend for vehicular flow counting in intelligent traffic systems and detecting dirt as part of the functionality of smart vacuum robots.

While this study has made valuable advancements, there remains room for further optimization: The employment of publicly available data sets has enhanced the reliability of the results, while the diversity of data types utilized could be further expanded; a robust object detection algorithm has been proposed, yet the validation process was primarily conducted on the graphics processing units of desktop platforms; although performance comparisons were made with various algorithms demonstrating state-of-the-art (SOTA) capabilities, the sample size of algorithms could be further increased for a more comprehensive analysis.

In our forthcoming research endeavor, we aim to meticulously extend the ambit of testing to encompass a more expansive array of data sets, deliberately emphasizing the augmentation of category diversity and the enlargement of target sizes, with the ultimate objective of corroborating the steadfastness of our research results. Concurrently, we plan to undertake a rigorous benchmarking process against the vanguard of contemporary models, a step that promises to shed light on the relative merits of our proposed model with enhanced precision. Furthermore, we intend to initiate deployment trials on mobile smartphones and robots endowed with integrated edge computing faculties, rigorously evaluating the model's effectiveness in a variegated assortment of authentic operational settings, thereby substantiating its robustness with indubitable evidence. Ultimately, we will refine this algorithm, leveraging the insights derived from our comprehensive research, to enhance its functionality specifically for edge platforms with considerable limitations. This optimization is aimed at guaranteeing that, even under such constraints, the algorithm achieves the highest caliber of performance attainable.

REFERENCES

Alsmirat, M. A., Al-alem, F. A., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. B. (2019). Impact of digital fingerprint image quality on the fingerprint recognition accuracy. *Multimedia Tools and Applications*, *78*(3), 3649–3688. doi:10.1007/s11042-017-5537-5

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J. I., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. doi:10.1186/s40537-021-00444-8 PMID:33816053

Behera, T. K., Bakshi, S., Sa, P. K., Nappi, M., Castiglione, A., Vijayakumar, P., & Gupta, B. B. (2023). The NITRdrone dataset to address the challenges for road extraction from aerial images. *Journal of Signal Processing Systems for Signal, Image, and Video Technology*, *95*(2-3), 197–209. doi:10.1007/s11265-022-01777-0

Biglari, A., & Tang, W. (2023). A review of embedded machine learning based on hardware, application, and sensing scheme. *Sensors (Basel)*, *23*(4), 2131. doi:10.3390/s23042131 PMID:36850729

Cao, M., Fu, H., Zhu, J., & Cai, C. (2022). Lightweight tea bud recognition network integrating GhostNet and YOLOv5. *Mathematical Biosciences and Engineering*, *19*(12), 12897–12914. doi:10.3934/mbe.2022602 PMID:36654027

Chen, X., & Gong, Z. (2021). *YOLOv5-Lite* [Data Set] https://github.com/ppogg/YOLOv5-Lite

Chopra, M., Singh, S. K., Sharma, A., & Gill, S. S. (2022). A comparative study of generative adversarial networks for text-to-image synthesis. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–12. doi:10.4018/IJSSCI.300364

Chu, J., Zhao, X., Song, D., Li, W., Zhang, S., Li, X., & Liu, A. (2022). Improved semantic representation learning by multiple clustering for image-based 3d model retrieval. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–20. doi:10.4018/IJSWIS.297033

Deng, L., Li, H., Liu, H., & Gu, J. J. (2022). A lightweight YOLOv3 algorithm used for safety helmet detection. *Scientific Reports*, *12*(1), 10981. doi:10.1038/s41598-022-15272-w PMID:35768467

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., & Sun, J. (2021). RepVGG: Making VGG-style ConvNets great again. *2021 IEEE/CVF conference on computer vision and pattern recognition,* 13728-13737.

Feng, X. L., Gao, X. W., & Luo, L. (2021). X-SDD: A new benchmark for hot rolled steel strip surface defects detection. *Symmetry*, *13*(4), 706. doi:10.3390/sym13040706

Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., & Xu, C. (2020). GhostNet: More features from cheap operations. *2020 IEEE/CVF conference on computer vision and pattern recognition,* 1577-1586.

Han, K., Wang, Y., Xu, C., Guo, J., Xu, C., Wu, E., & Tian, Q. (2022). GhostNets on heterogeneous devices via cheap operations. *International Journal of Computer Vision*, *130*(4), 1050–1069. doi:10.1007/s11263-022-01575-y

Hunt, B., Ruiz, A. J., & Pogue, B. W. (2021). Smartphone-based imaging systems for medical applications: A critical review. *Journal of Biomedical Optics*, *26*(4). Advance online publication. doi:10.1117/1.JBO.26.4.040902 PMID:33860648

Islam, K. (2020). Person search: New paradigm of person re-identification: A survey and outlook of recent works. *Image and Vision Computing*, *101*, 101. doi:10.1016/j.imavis.2020.103970

Jain, A. K., Yadav, A., Kumar, M., García-Peñalvo, F. J., Chui, K. T., & Santaniello, D. (2022). A cloud-based model for driver drowsiness detection and prediction based on facial expressions and activities. *International Journal of Cloud Applications and Computing*, *12*(1), 1–17. doi:10.4018/IJCAC.312565

Ji, B., Wang, H. B., Zhang, M. X., Mao, B. R., & Li, X. J. (2022). An efficient lightweight network based on magnetic resonance images for predicting Alzheimer's disease. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–18. doi:10.4018/IJSWIS.313715

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A review of yolo algorithm developments. *8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021): Developing Global Digital Economy After Covid-19, 199,* 1066-1073.

Jocher, G. (2021). *YOLOv5* [Data Set]. https://github.com/ultralytics/yolov5

Kaur, J., & Singh, W. (2022). Tools, techniques, datasets and application areas for object detection in an image: A review. *Multimedia Tools and Applications*, *81*(27), 38297–38351. doi:10.1007/s11042-022-13153-y PMID:35493415

Kong, L. R., Wang, J. Z., & Zhao, P. (2022). YOLO-G: A lightweight network model for improving the performance of military targets detection. *IEEE Access : Practical Innovations, Open Solutions*, *10*, 55546–55564. doi:10.1109/ACCESS.2022.3177628

Krichen, M. (2021). Anomalies detection through smartphone sensors: A review. *IEEE Sensors Journal*, *21*(6), 7207–7217. doi:10.1109/JSEN.2021.3051931

Lee, Y., Hwang, J., Lee, S., Bae, Y., & Park, J. (2019). An energy and GPU-computation efficient backbone network for real-time object detection. *2019 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW 2019)*, 752-760.

Li, D., Deng, L., Gupta, B. B., Wang, H., & Choi, C. (2019). A novel CNN based security guaranteed image watermarking generation scenario for smart city applications. *Information Sciences*, *479*, 432–447. doi:10.1016/j.ins.2018.02.060

Li, X. Y., & Zhang, J. G. (2023). A semantic feature enhancement-based aerial image target detection method using dense RFB-FE. *International Journal on Semantic Web and Information Systems*, *19*(1), 1–18. doi:10.4018/IJSWIS.331083

Li, Y., Yuan, H., Wang, Y., & Xiao, C. (2022). GGT-YOLO: A novel object detection algorithm for drone-based maritime cruising. *Drones (Basel)*, *6*(11), 335. doi:10.3390/drones6110335

Liang, T., Bao, H., Pan, W., & Pan, F. (2022). ALODAD: An anchor-free lightweight object detector for autonomous driving. *IEEE Access : Practical Innovations, Open Solutions*, *10*, 40701–40714. doi:10.1109/ACCESS.2022.3166923

Ma, N., Zhang, X., Zheng, H., & Sun, J. (2018). ShuffleNet V2: Practical guidelines for efficient CNN architecture design. *Computer Vision*, *11218*, 122–138.

Mandle, A. K., Sahu, S. P., & Gupta, G. P. (2022). CNN-based deep learning technique for the brain tumor identification and classification in MRI images. *International Journal of Software Science and Computational Intelligence*, *14*(1), 1–20. doi:10.4018/IJSSCI.304438

Nhi, N. T. U., Le, T. M., & Van, T. T. (2022). A model of semantic-based image retrieval using C-tree and neighbor graph. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–23. doi:10.4018/IJSWIS.295551

Patrikar, D. R., & Parate, M. R. (2022). Anomaly detection using edge computing in video surveillance system. *International Journal of Multimedia Information Retrieval*, *11*(2), 85–110. doi:10.1007/s13735-022-00227-8 PMID:35368446

Qi, J. H., Zhang, J. D., & Meng, Q. Y. (2021). Auxiliary equipment detection in marine engine rooms based on deep learning model. *Journal of Marine Science and Engineering*, *9*(9), 1006. doi:10.3390/jmse9091006

Qian, W. G., Li, H., & Mu, H. P. (2022). Circular LBP prior-based enhanced GAN for image style transfer. *International Journal on Semantic Web and Information Systems*, *18*(2), 1–15. doi:10.4018/IJSWIS.315601

Qing, Y., Liu, W., Feng, L., & Gao, W. (2021). Improved YOLO network for free-angle remote sensing target detection. *Remote Sensing (Basel)*, *13*(11), 2171. doi:10.3390/rs13112171

Qiu, S., Li, Y., Zhao, H., Li, X., & Yuan, X. (2022). Foxtail millet ear detection method based on attention mechanism and improved YOLOv5. *Sensors (Basel)*, *22*(21), 8206. doi:10.3390/s22218206 PMID:36365902

Ren, S., He, K., Girshick, R. B., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. doi:10.1109/TPAMI.2016.2577031 PMID:27295650

Ruan, S., Li, S., Lu, C., & Gu, Q. (2023). A real-time negative obstacle detection method for autonomous trucks in open-pit mines. *Sustainability (Basel)*, *15*(1), 120. doi:10.3390/su15010120

Samothai, P., Sanguansat, P., Kheaksong, A., Srisomboon, K., & Lee, W. (2022). The evaluation of bone fracture detection of YOLO series. *2022 37th international technical conference on circuits/systems, computers and communications (ITC-CSCC 2022)*, 1054-1057.

Su, W., Li, L., Liu, F., He, M., & Liang, X. (2022). AI on the edge: A comprehensive review. *Artificial Intelligence Review*, *55*(8), 6125–6183. doi:10.1007/s10462-022-10141-4

Sun, L., Wang, P. S., Liu, P. Y., & Nie, Z. G. (2023). Image processing method of a visual communication system based on convolutional neural network. *International Journal on Semantic Web and Information Systems*, *19*(1), 1–19. doi:10.4018/IJSWIS.330022

Ultralytics. (2023). *YOLOv8* [Data Set]. https://github.com/ultralytics/ultralytics

Wan, Z., Tian, F., & Zhang, C. (2023). Sheep face recognition model based on deep learning and bilinear feature fusion. *Animals (Basel)*, *13*(12), 1957. doi:10.3390/ani13121957 PMID:37370467

Wang, H., Li, Z., Li, Y., Gupta, B. B., & Choi, C. (2020). Visual saliency guided complex image retrieval. *Pattern Recognition Letters*, *130*, 64–72. doi:10.1016/j.patrec.2018.08.010

Wang, H., Shang, S., Wang, D., He, X., Feng, K., & Zhu, H. (2022). Plant disease detection and classification method based on the optimized lightweight YOLOv5 model. *Agriculture-Basel, 12*(7).

Wang, X. F., Hao, X., & Wang, K. (2023). MC-YOLO-based lightweight detection method for nighttime vehicle images in a semantic web-based video surveillance system. *International Journal on Semantic Web and Information Systems*, *19*(1), 1–18. doi:10.4018/IJSWIS.330752

Wei, C., Li, H., Shi, J., Zhao, G., Feng, H., & Quan, L. (2022). Row anchor selection classification method for early-stage crop row-following. *Computers and Electronics in Agriculture*, *192*, 192. doi:10.1016/j.compag.2021.106577

Wu, Z. L., Wang, X., & Chen, C. (2021). Research on lightweight infrared pedestrian detection model algorithm for embedded platform. *Security and Communication Networks*, *2021*, 1–7. doi:10.1155/2021/1549772

Xiao, Y., Zhou, K., Guangzhen, C., Jia, L., Fang, Z., Yang, X., & Xia, Q. (2021). Deep learning for occluded and multi-scale pedestrian detection: A review. *IET Image Processing*, *15*(2), 286–301. doi:10.1049/ipr2.12042

Yang, L., Zhang, R., Li, L., & Xie, X. (2021). SimAM: A simple, parameter-free attention module for convolutional neural networks. *International Conference on Machine Learning, 139*.

Yang, Y., Zhou, Y., Din, N. U., Li, J., He, Y., & Zhang, L. (2023). An improved YOLOv5 model for detecting laser welding defects of lithium battery pole. *Applied Sciences-Basel, 13*(4).

Yao, Y., Wang, G. Z., & Fan, J. H. (2023). WT-YOLOX: An efficient detection algorithm for wind turbine blade damage based on YOLOX. *Energies*, *16*(9), 3776. doi:10.3390/en16093776

Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., & Hoi, S. C. (2022). Deep learning for person re-identification: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(6), 2872–2893. doi:10.1109/TPAMI.2021.3054775 PMID:33497329

Yu, C., Li, J., Li, X., Ren, X., & Gupta, B. B. (2018). Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram. *Multimedia Tools and Applications*, *77*(4), 4585–4608. doi:10.1007/s11042-017-4637-6

Zaidi, S. S., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M. N., & Lee, B. (2022). A survey of modern deep learning based object detection models. *Digital Signal Processing*, *126*, 126. doi:10.1016/j.dsp.2022.103514

Zhang, C. X., Kang, F., & Wang, Y. X. (2022). An improved apple object detection method based on lightweight YOLOv4 in complex backgrounds. *Remote Sensing (Basel)*, *14*(17), 4150. doi:10.3390/rs14174150

Zheng, Z., Zhou, J., Gan, J., Luo, S., & Gao, W. (2022). Fine-grained image classification based on cross-attention network. *International Journal on Semantic Web and Information Systems*, *18*(1), 1–12. doi:10.4018/IJSWIS.315747

Zhu, L., Lee, F., Cai, J., Yu, H., & Chen, Q. (2022). An improved feature pyramid network for object detection. *Neurocomputing*, *483*, 127–139. doi:10.1016/j.neucom.2022.02.016

Yiheng Wu, Algorithm software engineer. Education: Master's degree, graduated from China University of Mining and Technology in 2022. Involved in work/research areas: Computer Vision, Refrigeration Equipment Design and Control.

Jiaqiang Dong, Software Engineer, Associate Dean of the School of Information Technology, and Professor of Computer Science and Technology at Xichang University. Education: Master's degree, graduated from the University of Electronic Science and Technology of China in 2006. Involved in work/research areas: Computer Applications and Simulation.

Jianxin Chen, Thermal Design Engineer at CALB technology institute Co., Ltd. Education: Master's degree, graduated from China University of Mining and Technology in 2022. Involved in work/research areas: Thermal Design and Numerical Simulation.