

Article

PRC-Light YOLO: An Efficient Lightweight Model for Fabric Defect Detection

Baobao Liu ^{1,2,*}, Heying Wang ¹, Zifan Cao ¹, Yu Wang ¹, Lu Tao ¹, Jingjing Yang ¹ and Kaibing Zhang ¹

¹ School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China; whey820@163.com (H.W.); zifan_96@163.com (Z.C.); w201572972@163.com (Y.W.); yangj012019@163.com (J.Y.); zhangkaibing@xpu.edu.cn (K.Z.)

² Shaanxi Key Laboratory of Clothing Intelligence, School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China

* Correspondence: liubaobao1222@163.com

Abstract: Defect detection holds significant importance in improving the overall quality of fabric manufacturing. To improve the effectiveness and accuracy of fabric defect detection, we propose the PRC-Light YOLO model for fabric defect detection and establish a detection system. Firstly, we have improved YOLOv7 by integrating new convolution operators into the Extended-Efficient Layer Aggregation Network for optimized feature extraction, reducing computations while capturing spatial features effectively. Secondly, to enhance the performance of the feature fusion network, we use Receptive Field Block as the feature pyramid of YOLOv7 and introduce Content-Aware ReAssembly of Features as upsampling operators for PRC-Light YOLO. By generating real-time adaptive convolution kernels, this module extends the receptive field, thereby gathering vital information from contexts with richer content. To further optimize the efficiency of model training, we apply the HardSwish activation function. Additionally, the bounding box loss function adopts the Wise-IOU v3, which incorporates a dynamic non-monotonic focusing mechanism that mitigates adverse gradients from low-quality instances. Finally, in order to enhance the PRC-Light YOLO model's generalization ability, we apply data augmentation techniques to the fabric dataset. In comparison to the YOLOv7 model, multiple experiments indicate that our proposed fabric defect detection model exhibits a decrease of 18.03% in model parameters and 20.53% in computational load. At the same time, it has a notable 7.6% improvement in *mAP*.

Keywords: fabric defect detection; YOLOv7; lightweight network; HardSwish; Wise-IOU v3



Citation: Liu, B.; Wang, H.; Cao, Z.; Wang, Y.; Tao, L.; Yang, J.; Zhang, K. PRC-Light YOLO: An Efficient Lightweight Model for Fabric Defect Detection. *Appl. Sci.* **2024**, *14*, 938. <https://doi.org/10.3390/app14020938>

Academic Editors: Yue Wu, Kai Qin, Qiguang Miao and Mingyang Zhang

Received: 14 December 2023

Revised: 13 January 2024

Accepted: 15 January 2024

Published: 22 January 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

The detection of fabric defects plays a significant role in the industrial sector. In the process of fabric manufacturing, the defects on the surface of fabric are inevitable. Therefore, scholars are focused on the effective identification and precise localization of these fabric defects [1].

There are two main categories of methods for detecting fabric defects: traditional methods and deep learning methods. The traditional methods include the texture structure method, histogram statistics, spectral method, modeling method and adaptive dictionary learning method for fabric defect detection. Li et al. put forward an approach that relies on saliency region and similarity localization detection to achieve accurate contour segmentation of periodic textured fabric images and address the challenge of defect detection [2]. This detection speed is not dominant, and the detection effect for fabric images with no obvious defects or complex texture fabric images is generally not satisfactory. Xiang et al. proposed a yarn-dyed fabric defect detection algorithm based on Fourier convolution and convolutional self-encoder that avoids the conventional method of introducing noise in the training phase and employs a random masking method to generate image pairs for training instead [3]. In the process, some spatial domain information may be lost, making it difficult

for the decoder to fully recover the original input when inverting the frequency domain data back to the spatial domain. Wu et al. developed a method of cotton fabric defect detection based on K-Singular Value Decomposition dictionary learning [4], in which the dictionary is generally better suited to handle sparse textures. Kanwal et al. utilized the K-means algorithm to develop a bag-of-words model and proposed a significance-based bag-of-words model for fabric defect detection [5]. The bag-of-words model may ignore the spatial information and contextual relationships of features, leading to the loss of local information and structural information in images. Although the detection speed and performance of the methods based on traditional image processing are better than those of manual detection, the methods of generalization ability are inadequate. In the past two decades, deep learning has become a trend for fabric defect detection, and the techniques have achieved significant advancements in this field [6]. Methods for detection based on deep learning can be classified further into two primary categories: single-stage and two-stage methods. The single-stage methods mainly include Solid State Drive (SSD) [7] and You Only Look Once (YOLO) [8,9] series, and the two-stage algorithms comprise R-CNN [10], Fast R-CNN [11], Faster R-CNN [12] and so on. Zhao et al. utilized ROI Align and ResNet50 network to replace the region of interest pooling layer and VGG16 features extraction network in Faster R-CNN [13]. In addition, the softmax classifier is used to identify the image and obtain the prediction results, but the detection speed needs to be improved. Zhang et al. introduced the channel attention mechanism to highlight defect features and suppress background noise features in MobileNetV2-SSDLite and redefined the loss function by Focal Loss [14]. However, this model still has a lot of parameters and is difficult to deploy. Based on YOLOv5s, Zhou et al. used K-Means++ clustering to acquire the anchor frames, followed by combining the CARAFE upsampling operator. In addition, they added a small target detection layer to the baseline model, which results in an increase in the amount of model parameters [15]. Wu et al. embedded Classification-Aware Regression Loss (CARL) into YOLOF. Although this method establishes a relationship between classification and target localization [16], CARL exhibits sensitivity to noise and outliers within the data, potentially resulting in the model's tendency to overfit to these anomalies. Lin et al. proposed a weighted bidirectional feature network by substituting the Backbone module within the original YOLOv5 structure with the Swin Transformer [17]. Moreover, they employed the generalized focal loss to decrease the missed detection rate and improve the learning of positive sample instances. However, the Swin Transformer increases the parameter number and calculation amount of the original network, and there is no guarantee that the model will achieve the same detection results on either mobile or embedded terminals. Guo et al. embedded the Atrous Spatial Pyramid Pooling module and squeeze-and-excitation channel attention module into YOLO. They used convolution kernels with varying expansion rates to pool feature maps and achieved improved defect detection accuracy by self-learning the weights of each feature channel [18], but the background of the dataset is too simple, which can lead to idealized experimental results. Di et al. developed a deconvolution-based adaptive feature fusion network and introduced a context receptive field block into the Backbone of the YOLOv5 model; in addition, they used exponential distance IOU as the model bounding box loss function [19]. The detection precision of the model is improved by adaptively weighting the bounding box gradient, but this method has 3.8M more parameters than the baseline YOLOv5 and a lower *mAP* value than YOLOv7.

The methods mentioned above can achieve higher defect detection accuracy or the quantity of computation, but often at the cost of adding more parameters to the original model. In addition, the background of the dataset used in some literature is not complicated enough, so they have high *mAP* values. In this paper, we made some data images of fabric defects with complicated background and researched the structure of YOLOv7 [20]. Our main contributions are as follows:

- We propose Partial Convolution (PConv) to replace the 3×3 Conv of Extended-Efficient Layer Aggregation Networks (E-ELAN) in the Backbone module. This replacement aims to decrease network model calculation and parameters.
- In the Neck module, we incorporate a Receptive Field Block (RFB) [21] and integrate the Content-Aware ReAssembly of FEatures (CARAFE) lightweight upsampling operator.
- We utilize the HardSwish activation function to decrease the computational burden and memory access of the network. The Wise-IOU v3 with dynamic non-monotonic focusing mechanism is applied as the bounding box loss function of the model.
- Compared with YOLOv7 and other detection models, the experimental results validate that PRC-Light YOLO has the highest *mAP* and improves the performance of fabric defect detection.

The remaining sections of this article are organized as follows: In Section 2, we introduce the YOLOv7 model and its image processing process. Section 3 presents the PRC-Light YOLO model, which encompasses the construction of lightweight modules, feature fusion network and optimization of activation functions and loss functions. Section 4 provides a detailed account of the experimental results and analysis. In Section 5, we design a fabric defect detection system. Section 6 discusses further research based on the experimental results. Finally, a conclusion is provided.

2. YOLOv7 Model Structure

The YOLOv7 model is mainly composed of three parts: Backbone, Neck and Head. The model structure is shown in Figure 1.

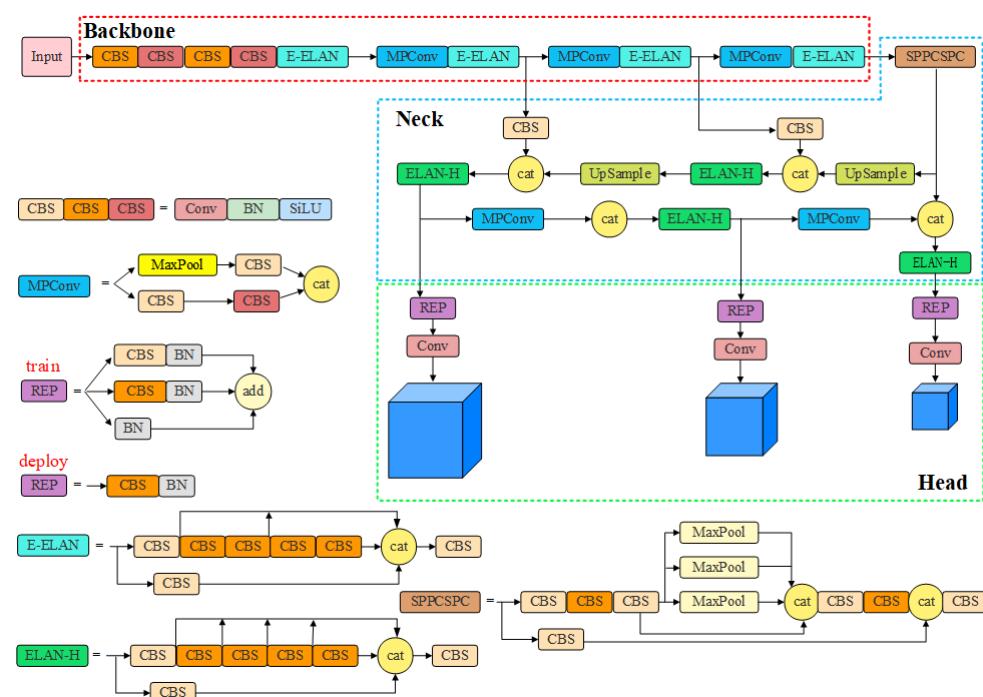


Figure 1. The network structure of YOLOv7.

2.1. Backbone

The Backbone part mainly comprises the CBS (Convolution, Batch normalization, SiLU) layer, E-ELAN module, and MPCConv (Max Pool Convolution) module [22]. Among them, different CBS colors signify varied convolution kernels. The MPCConv module comprises upper and lower branches. The upper branch shows the feature map via MaxPool operation, effectively reducing spatial dimensions by half. Subsequently, a CBS operation is applied to halve channel count. In the lower branch, the feature map is processed by the first CBS to reduce the number of channels, followed by another CBS to

further reduce channel count. Lastly, the upper and lower branches are merged by tensor splicing. The E-ELAN module contains two branches [23], and this module improves the ability of the network to learn additional features.

2.2. Neck

The Path Aggregation Feature Pyramid Network (PAFPN) structure is used in the Neck module of the YOLOv7 for feature fusion [24]. This involves transmitting and fusing higher-level feature information through an upsampling process, followed by feature map predictions obtained through a down-sampling fusion method. The final output comprises results from three feature layers. As image processing may result in significant image distortion, the SPPCSPC module incorporates three parallel MaxPool operations during a series of convolutional operations, effectively reducing the occurrence of image distortion.

2.3. Head

In the Head detection section, the REP module differs between the training and inference stages. During the training stage, it comprises three branches. While in the inference stage, the REP module undergoes reparameterization and consists of only a 3×3 Conv.

Thus, the fabric defect detection process based on YOLOv7 is outlined as follows. Firstly, preceding the input of an image into the Backbone, the model executes a series of operations, including adaptive image scaling, Mosaic data enhancement, adaptive anchor frame calculation and so on. Secondly, the Backbone module extracts features from the processed image. Thirdly, the Neck module conducts feature fusion processing on the extracted features from the Backbone, generating feature maps of three different sizes: large, medium and small. Finally, the Head layer detects the fused features and outputs the final results.

3. Fabric Defect Detection Based on PRC-Light YOLO

Although YOLOv7 has excellent detection accuracy and speed on datasets such as COCO and PASCAL VOC [20], its performance diminishes for fabric defect detection in fabric datasets. Thus, we study the theoretical foundation and network architecture of YOLOv7, subsequently proposing PRC-Light YOLO for a fabric defect detection model.

3.1. Lightweight Backbone Network

The reduction of floating-point operations (*FLOPs*) can alleviate the computational burden of neural networks, thereby shortening both forward and backward propagation times. This reduction significantly contributes to achieving lower latency. Simultaneously, lower *FLOPs* can enhance the parallelism of neural networks on hardware, increasing throughput and enabling faster processing of input data streams. Consequently, to achieve lightweight neural networks that meet the demands of low latency and high throughput, the focus of most researchers remains on minimizing the total number of *FLOPs*. This is because there exists the following relationship between neural network latency and *FLOPs*:

$$\text{Latency} = \frac{\text{FLOPs}}{\text{FLOPS}} \quad (1)$$

Floating-point operations per second (*FLOPS*) is a metric for computing speed efficiency. As indicated by Equation (1), reducing the number of *FLOPs* does not always guarantee a decrease in neural network latency. Hence, it is essential to concurrently reduce *FLOPs* and optimize *FLOPS* to truly achieve low latency.

Chen et al. demonstrated that frequent memory accesses by convolution operations reduced *FLOPS* values in neural networks [25]. ShuffleNet [26], MobileNet [27] and GhostNet [28], for example, embed Group Convolution (GConv) or Depthwise Convolution (DWConv) to extract spatial features. For an input $I \in R^{c \times h \times w}$, with c convolution kernels $W \in R^{k \times k}$, we obtain an output $O \in R^{c \times h \times w}$. Then, the *FLOPs* for regular convolution

is $h \times w \times k^2 \times c^2$, and the DWConv's *FLOPs* is $h \times w \times k^2 \times c$. Despite the lower *FLOPs*, the DWConv operation cannot be directly followed by the regular convolution operation, or it will lead to a significant degradation of model accuracy. To compensate for the drop in precision, the PointWise Convolution (PWConv) operation is typically employed after DWConv, with the channel count in DWConv being augmented to c . As a result, the Memory Access Cost (MAC) for DWConv is as follow:

$$h \times w \times 2c' + k^2 \times c' \approx h \times w \times 2c' \quad (2)$$

The MAC for regular convolution can be represented as follows:

$$h \times w \times 2c' + k^2 \times c \approx h \times w \times 2c \quad (3)$$

A comparison of Equations (2) and (3) suggests that the DWConv operation may elevate the frequency of memory accesses in the process of reducing *FLOPs*. Furthermore, since these models typically involve extra data processing operations such as splicing, disambiguation, and pooling, the inclusion of DWConv does not invariably curtail the neural network's latency.

Therefore, we adopt a novel partially convolution PConv in the Backbone, which reduces the *FLOPs* and MAC of neural networks. The principle of the PConv is illustrated in Figure 2.

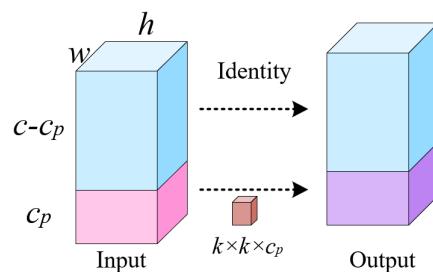


Figure 2. The principle of PConv.

As can be seen from Figure 2, PConv only performs the convolution operation on part of the input channels by using the regular conventional Conv for spatial feature extraction. Meanwhile, the remaining $c - c_p$ channels keep an identity mapping to Output. This approach allows selective processing while preserving certain channel information. The *FLOPs* of PConv is calculated to be $h \times w \times k^2 \times c_p^2$, and the MAC is $h \times w \times 2c_p + k^2 \times c_p^2 \approx h \times w \times 2c_p$, where c_p denotes the number of channels in PConv that perform regular convolution. The c_p is usually taken to be $1/4c$. Therefore, the *FLOPs* of PConv are only $1/16$ of those of the regular Conv, resulting in reduced memory access. This paper replaces all 3×3 convolution operations within E-ELAN of the YOLOv7 Backbone network with PConv. This substitution allowed for a reduction in the frequency of memory accesses by the neural network. This substitution allowed for the maintenance of higher *FLOPS* with fewer *FLOPs*, consequently reducing neural network latency and making the network more effective at extracting spatial features.

3.2. Improved Feature Fusion Network

3.2.1. RFB Feature Pyramid

The Feature Pyramid SPPCSP is made up of two components: Single-Point Perfection (SPP) [29] and Complexity-Sensitive Perfection (CSP) [30]; as a result, it has a larger number of model parameters and computations, which leads to slower inference speed of the YOLOv7 model. Additionally, in the sampling process, SPPCSP might cause a reduction in the size of input feature maps. This could potentially lead to a degree of feature information loss, which has an impact on the detection of small objects. Consequently, we integrate the RFB feature pyramid into the Neck of the YOLOv7 model, which is able to learn deeper

features from the lightweight CNN model, making the detection model faster and more accurate with less calculation.

The RFB is a multi-branch convolution [31] block inspired by Inception [32] in network construction. Its internal structures are divided into two parts: multi-branch convolution employing various convolution kernels and dilated convolutions featuring variable rates. The dilated convolution operation's primary function is to expand the receptive field so that it can capture crucial information in regions with more context. The structure of the RFB is illustrated in Figure 3.

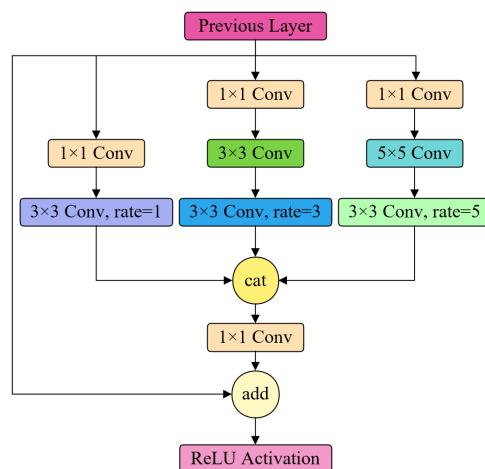


Figure 3. The structure of the RFB.

In the RFB, we utilize branches with varying dilation factors to capture multiscale information and different ranges of dependencies. Despite these branches having distinct receptive fields, they share the same weights, which help reduce the number of parameters, thereby lowering the risk of overfitting and maximizing the utilization of information from each sample. Finally, the RFB concatenates the computation results from different branches, preventing the model from encountering gradient explosion and vanishing gradient issues during training, ultimately achieving the goal of fusing different features.

3.2.2. CARAFE Upsampling Operator

The upsampling method in the YOLOv7 feature fusion network is the nearest neighbor interpolation method [33]. The principle of this method is to identify the nearest neighbor pixel in the target image based on each pixel's position in the original image and subsequently replace the value of that pixel in the original image with the value of the nearest neighbor pixel. However, the image quality after scaling by this method is poor, and obvious jagged edges may appear. To address this issue, we introduce the CARAFE upsampling operator. This innovation significantly bolsters the neural network's semantic feature extraction capacity during the upsampling process within object detection tasks. Importantly, this improvement is made without significantly increasing the amount of computations and parameters. The CARAFE upsampling process is depicted in Figure 4.

The CARAFE architecture is made of two primary modules: the upsampling kernel prediction module and feature reassembly module. These components work together to enable effective upsampling and feature reassembly within the network. In the upsampling kernel prediction module, assuming the upsampling ratio is σ , the process begins with channel compression through a 1×1 convolution operation on the input feature map of size $H \times W \times C$, resulting in a feature map of size $H \times W \times C_m$. Subsequently, encoding the feature map using a $k_{enc} \times k_{enc}$ convolutional kernel generates the recombinant kernel $k_{up} \times k_{up}$, obtaining a feature map with $\sigma^2 \times k_{up}^2$ channels. The channels are then spatially unfolded and subjected to softmax normalization, producing a feature map of size $\sigma H \times \sigma W \times K_{up}^2$, where each feature value corresponds to an upsampling kernel. In the feature recombination module, through the mapping relationship, the region $k_{up} \times k_{up}$

centered in the input feature map is taken out, and then the dot product operation is performed with the upsampling kernel at the same position in $\sigma H \times \sigma W \times K_{up}^2$ to obtain the output value.

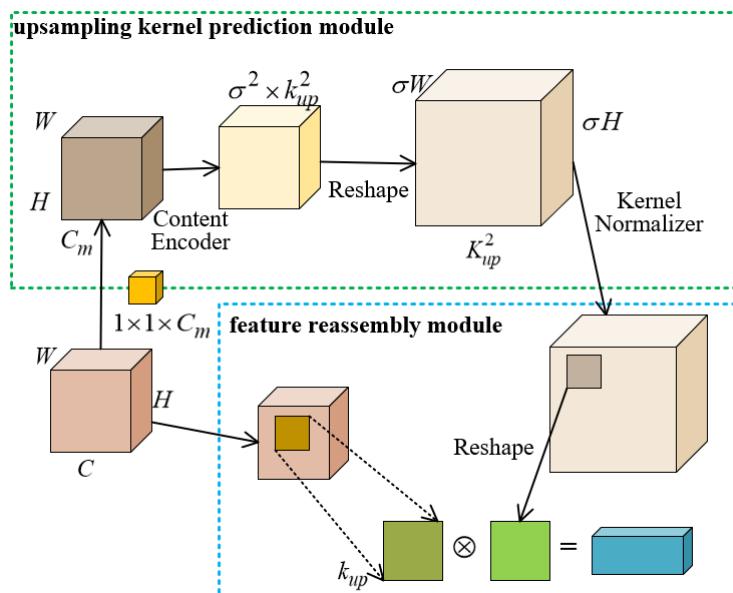


Figure 4. Upsampling process on CARAFE.

The CARAFE achieves instance-specific content-aware processing by generating real-time adaptive convolution kernels and can aggregate contextual information over a larger receptive field. Therefore, we introduce the CARAFE upsampling operator to replace two upsampling operations in the Neck structure of the YOLOv7 model. According to the trade-off relationship between k_{enc} and k_{up} , to maintain the performance and efficiency, k_{up} takes the value of 7 and k_{enc} takes the value of 5 in the experiments.

3.3. HardSwish Activation Function

Activation functions are significant in neural networks, introducing non-linearity and enabling the networks to learn and represent complex functions, thereby enhancing the network's representational capacity [34]. In the absence of activation functions, deeper neural networks would be limited to handling linearly differentiable problems [35]. Commonly used activation functions are Sigmoid, ReLU, Swish, Mish, GELU and so on. The YOLOv7 model employs the SiLU activation function, which is expressed as follows:

$$SiLU(x) = x \cdot \text{sigmoid}(x) = \frac{x}{1 + e^{-x}} \quad (4)$$

The computation process of the Sigmoid function involves exponentiation, which undoubtedly increases the computational complexity of the SiLU activation function. However, the Sigmoid activation function can be approximated using a piecewise linear function called HardSigmoid [36], which significantly reduces the computational cost. The HardSigmoid activation function and HardSwish activation function expressions are as follows:

$$\text{HardSigmoid}(x) = \begin{cases} 0, & x \leq -3 \\ 1, & x \geq 3 \\ (x + 3)/6, & \text{otherwise} \end{cases} \quad (5)$$

$$\text{HardSwish}(x) = \begin{cases} 0, & x \leq -3 \\ (x^2 + 3x)/6, & -3 < x < 3 \\ x, & x \geq 3 \end{cases} \quad (6)$$

The HardSwish and SiLU activation function output curves are shown in Figure 5. HardSwish and SiLU are differentiable at all points and have no upper bounds, enabling them to avoid overfitting while making the model more generalizable. Furthermore, the segmented function reduces the number of memory accesses, which significantly reduces the latency cost.

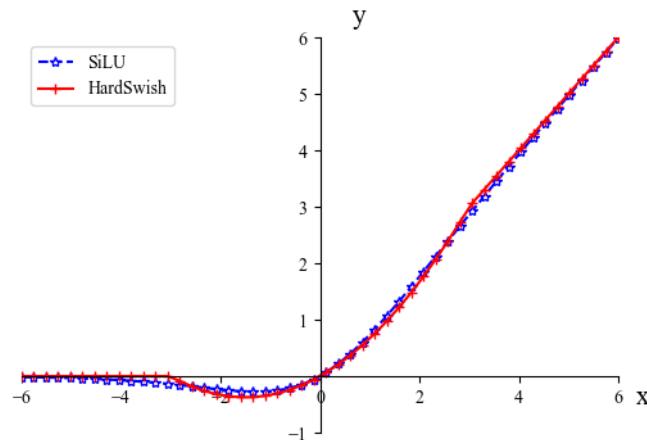


Figure 5. Activation function output curves.

3.4. Wise-IOU v3 Bounding Box Loss

The loss function of YOLOv7 comprises three parts: Bounding Box Loss (BoxLoss), Classification Loss (ClsLoss) and Object Confidence Loss (ObjLoss). Among them, the BoxLoss employs the Complete Intersection Over Union (CIOU) loss [37], and the ObjLoss and ClsLoss are calculated using the binary cross-entropy loss function. The YOLOv7 total loss expression is shown in Equation (7).

$$\text{Loss} = \text{Loss}_{\text{Obj}} + \text{Loss}_{\text{cls}} + \text{Loss}_{\text{Box}} \quad (7)$$

The CIOU loss in YOLOv7 considers three geometric factors: the overlap area between bounding boxes, the difference in aspect ratios and the distance between their center points. Given a target frame and a prediction frame, the CIOU loss is calculated as:

$$L_{\text{CIOU}} = 1 - \text{IOU} + \frac{\sqrt{(i_x - j_x)^2 + (i_y - j_y)^2}}{c^2} + \alpha v \quad (8)$$

$$\alpha = \frac{v}{(1 - \text{IOU}) + v} \quad (9)$$

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h})^2 \quad (10)$$

The gradient of v with respect to w and h is calculated as follows:

$$\begin{cases} \frac{\partial v}{\partial w} = \frac{8}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h}) * \frac{h}{w^2 + h^2} \\ \frac{\partial v}{\partial h} = \frac{8}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h}) * \frac{w}{w^2 + h^2} \end{cases} \quad (11)$$

$(i_x, i_y), (j_x, j_y)$ are the center point coordinates of the real frame and the prediction frame, respectively, v measures the similarities in length ratios, α represents the balancing parameter and the diagonal length of the smallest outer rectangle containing the true and predicted boxes is denoted by c . Analysis of Equation (10) shows that when the aspect ratios of the predicted and real frame are equal, the penalty term related to the aspect ratio becomes 0. Thus, the aspect ratio penalty term loses its effect, which reduces the optimization performance of the model for similarity. From Equation (11), the relationship between $\frac{\partial v}{\partial w}$ and $\frac{\partial v}{\partial h}$ is inversely proportional, and during training, the simultaneous adjustment of both width and height dimensions in the prediction frame is not feasible. This constraint

hinders the prediction frame from effectively approximating the dimensions of the real frame [38].

The CIOU employs a monotonic focus mechanism, which is predicated on the assumption that high-quality examples exist within the training data. Its primary objective is to enhance the fitting capability of bounding box loss. Nevertheless, when the target detection training dataset includes low-quality examples, indiscriminately emphasizing the regression of bounding boxes for these low-quality instances evidently leads to a deterioration in the model's detection performance [39]. Therefore, we employ Wise-IOU v3 as the bounding box loss, which incorporates a non-monotonic dynamic focusing mechanism. This mechanism relies on the outlier degree rather than IOU to assess the quality of anchor boxes, which diminishes the detrimental gradient caused by low-quality examples, mitigates the competitiveness of high-quality anchor boxes and improves the overall performance of the detector. The equations are as follows:

$$R_{WIOU} = \exp\left(\frac{(i_x - i_y)^2 + (j_x - j_y)^2}{(W^2 + H^2)^*}\right) \quad (12)$$

$$L_{w1} = R_{WIOU} * L_{IOU} \quad (13)$$

$$r = \frac{\beta}{3 \times 1.9^{\beta-3}} \quad (14)$$

$$\beta = \frac{L_{IOU}^*}{\overline{L}_{IOU}} \in [0, +\infty) \quad (15)$$

$$L_{WIOU} = r * L_{w1} \quad (16)$$

W and H represent the width and height, respectively, of the smallest outer rectangle of the real frame and the prediction frame and $(W^2 + H^2)^*$ denotes the separation of W and H from the Computational Graph to prevent R_{WIOU} from making gradients that impede convergence. L_{IOU}^* is the monotonic focusing factor of L_{w1} . \overline{L}_{IOU} denotes the sliding mean with momentum m , and its dynamic updates contribute to maintaining an elevated overall β , effectively resolving the issue of slow convergence in the later stages of training. This enables Wise-IOU to flexibly adjust gradient gain allocation strategies during the training phase in order to best adapt to the current situation.

3.5. PRC-Light YOLO Model Structure

In summary, this paper proposes PRC-Light YOLO. The differences between PRC-Light YOLO and YOLOv7 are as follows:

- (1) We replace all 3×3 convolution operations of E-ELAN in the Backbone of YOLOv7 with PConv introduced in Section 3.1 to form the PE-ELAN module.
- (2) The RFB feature pyramid proposed in Section 3.2.1 replaces the SPPCSP module in the Neck of YOLOv7.
- (3) The CARAFE upsampling operators, as proposed in Section 3.2.2, replace the two upsampling operations in the Neck structure of the YOLOv7.
- (4) Due to the YOLOv7 model's utilization of the SiLU activation function, which is composed of exponentials, there is an increased computational cost. To address this issue, we bring in the HardSwish activation function, which is composed of piecewise functions, significantly reducing the computational burden of the model. Consequently, we replace the SiLU activation function in the CBS module with the HardSwish activation function. Convolution, Batch Normalization and HardSwish activation functions constitute the CBH module.
- (5) We have replaced the bounding box loss function of YOLOv7 with Wise-IOU v3. It helps mitigate adverse gradients stemming from low-quality instances, thereby enhancing the overall performance of the detector.

The PRC-Light YOLO structure is shown in Figure 6.

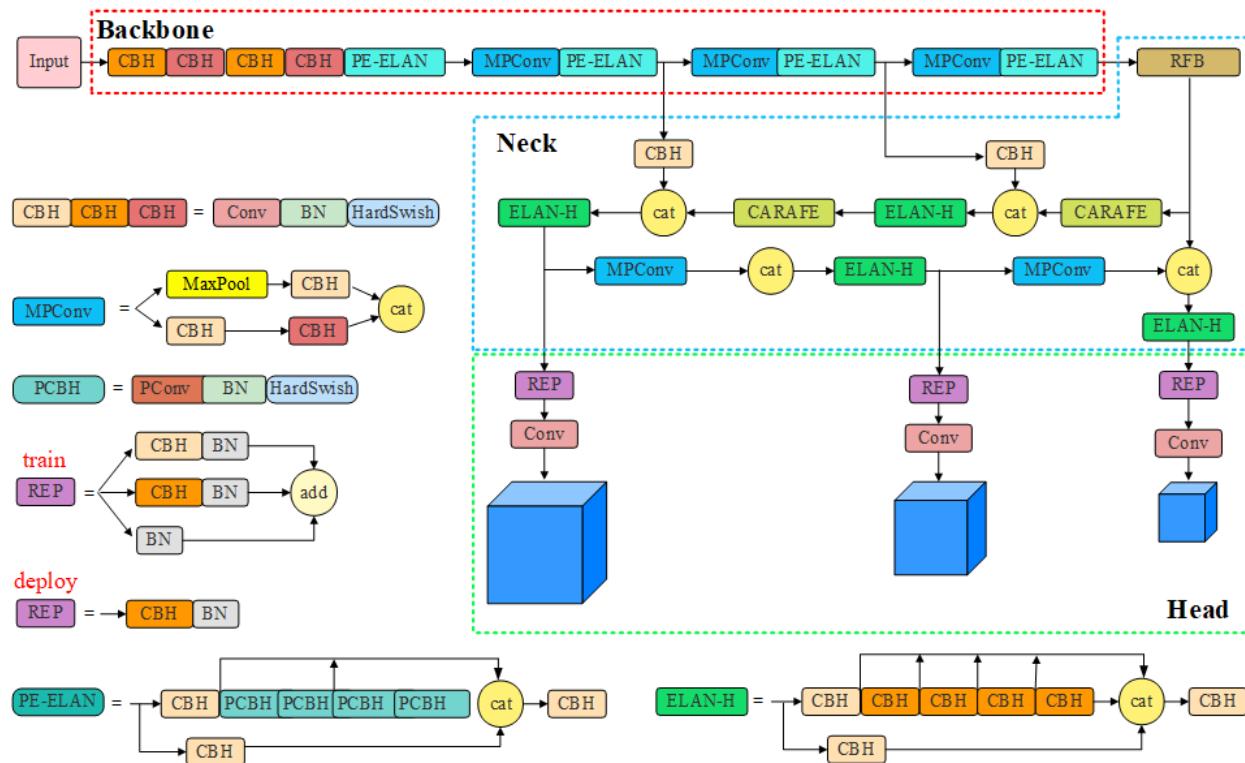


Figure 6. The PRC-Light YOLO structure.

4. Experiment

4.1. Fabric Defect Image Dataset

The dataset [40] was acquired from multiple resources, including apparel factories, the existing literature and the “Xuelang Manufacturing AI Challenge”, from which we select a total of 1061 images encompassing 4 categories: Warp hanged, Yard defects, Holes and Stains, named as dj, ds, pd and wz, respectively. Their characteristics are as follows:

(1) Warp hanged is small in size, typically suspended above the fabric without proper weaving, making it difficult to discern against a darker background.

(2) Yard defect refers to the occurrence of defects along the horizontal direction of the fabric within a specific length.

(3) Hole refers to a large number of broken weft holes on the cloth surface and the emergence of weft yarn, which is a serious defect.

(4) Stain manifests as an area on the fabric with a distinct color different from its surroundings, varying in size from large to small within the dataset.

In order to reduce the unnecessary computations of the network model, all images are cropped to 200 pixel \times 200 pixel to speed up the training. Then, we use MakeSense to label the defect types in the images and extract the txt annotation information files required for YOLOv7 model training. In order to enrich the fabric dataset, we randomly rotate the images, add noise, change the brightness, chromaticity and saturation techniques of the samples, perform data enhancement techniques on 1061 datasets, expand the dataset to 4244 datasets, enhance the PRC-Light YOLO model’s generalization capacity and then divide the training set, validation set and testing set according to the ratio of 8:1:1. Parts of the samples are shown in Figure 7.

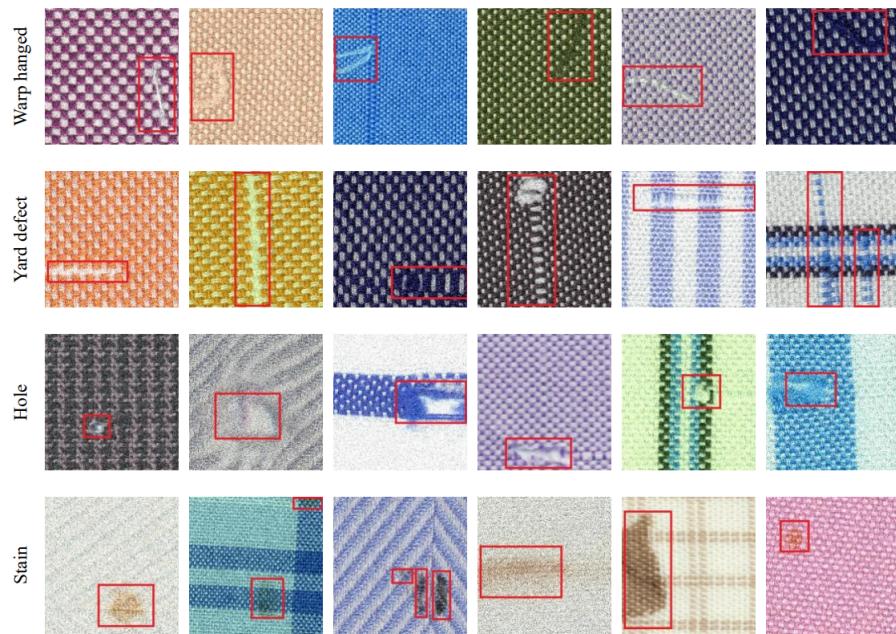


Figure 7. Defective fabric image samples.

4.2. Experimental Environment and Parameter Configuration

The experimental equipment parameters are shown in Table 1. The input image size is 320×320 , with a total of 150 epochs of training with batch size 16; the initial learning rate is set to 0.01.

Table 1. Experimental equipment parameters.

Name	Operating System	RAM	Graphics Card	CUDA	Python	Framework
Parameter	Windows X64	12G	NVIDIA Quadro P4000	11.3	3.8	PyTorch

4.3. Evaluation Metrics

To verify the improved model's performance, we utilize Precision (P), Recall (R), $F1$ -score and mean Average Precision (mAP) as the measures of detection accuracy; the formulas are calculated as follows:

$$P = \frac{TP}{TP + FP} \quad (17)$$

$$R = \frac{TP}{TP + FN} \quad (18)$$

$$F_1 - score = \frac{2 * P * R}{P + R} = \frac{2TP}{2TP + FP + FN} \quad (19)$$

$$mAP = \frac{\sum_{i=1}^N AP_i}{N} \quad (20)$$

TP represents true positives, representing the amount of positive samples accurately predicted to be positive in the positive samples. FP denotes false positives, indicating the count of negative samples that are predicted as positive samples. FN represents false negatives, depicting the count of positive samples that are wrongly predicted to be negative. Each category can draw PR curves based on Precision and Recall, every category corresponds to one PR curve, the area covered by which indicates the Average Precision (AP) of the category. mAP is the average value after summing up the APs of all categories.

4.4. Ablation Experiment

An ablation experiment is an important method in deep learning; by improving the modules of the YOLOv7 network structure and assessing the contribution of different improvement methods on its performance, we conducted seven sets of ablation experiments in all, where “✓” corresponds to the improvement method and the corresponding experimental order. The experimental results are listed in Table 2. Experiment 1 (exp1) is based on YOLOv7. Experiment 2 (exp2) replaces YOLOv7’s SiLU activation function with HardSwish. Experiment 3 (exp3) incorporates the Wise-IOU v3 into YOLOv7 as the bounding box loss function. Experiment 4 (exp4) substitutes YOLOv7’s SSPCSP pyramid with the RFB module. Experiment 5 (exp5) replaces the nearest neighbor interpolation method in YOLOv7 with the CARAFE operator. Experiment 6 (exp6) integrates HardSwish, Wise-IOU v3, RFB and CARAFE into YOLOv7. Experiment 7 (exp7) is based on PRC-Light YOLO, and it builds upon exp6 by replacing the 3×3 convolution operation in the Backbone’s E-ELAN with PConv.

Table 2. Results of ablation experiments.

Exp	HardSwish	Wise-IOU v3	RFB	CARAFE	PConv	Params/M	GFLOPs/G	mAP/%	Inference Time/ms
exp1	✗	✗	✗	✗	✗	37.21	105.2	75.4	35.08
exp2	✓	✗	✗	✗	✗	37.21	105.2	78.1	31.49
exp3	✗	✓	✗	✗	✗	37.21	105.2	79.1	32.08
exp4	✗	✗	✓	✗	✗	33.94	102.5	80.6	33.31
exp5	✗	✗	✗	✓	✗	37.87	106.5	79.8	34.65
exp6	✓	✓	✓	✓	✗	34.6	103.8	81.1	30.62
exp7	✓	✓	✓	✓	✓	30.5	83.6	83	28.46

From Table 2, it can be seen that the first set of experiments is based on YOLOv7, the model parameter number is 37.21 M, the Giga FLoating-Point Operations Per Second (*GFLOPs*) is 105.2 G and the *mAP* is 75.4%, followed by subsequent experiments, which are all based on the results of the first set of experiments as a benchmark for comparison. The *GFLOPs* mentioned above is used to measure the computational complexity of a model, and $1\text{GFLOPs} = 1 \times 10^9 \text{ FLOPs}$, so lower a *GFLOPs* reduces the latency of the neural network, thereby improving their computational speed. Experiments 2 and 3, without increasing the parameters, respectively, increased the *mAP* values by 2.7% and 3.7%, and in experiment 2, by introducing the HardSwish activation function, the inference time is lower than the other ones that do not use the HardSwish activation function, which verifies that the characteristic of the segmentation function greatly reduces the computational cost of the model. In Experiment 4, the introduction of the RFB feature pyramid reduces the parameters by 8.79% compared to the SPPCSPC module in YOLOv7 while simultaneously improving the *mAP* by 5.2 percentage points. Experiment 5 utilizes the CARAFE lightweight upsampling module in the feature fusion module, and the *mAP* value goes to 79.8% with only a slight increase in parameters and *GFLOPs*, which is a significant improvement. In order to verify the lightweight effect of the PConv, Experiment 6 integrates HardSwish, Wise-IOU v3, RFB and CARAFE into YOLOv7. By comparing Experiment 7 with Experiment 6, Our findings reveal a reduction in the network’s parameters by 11.85% and computations by 19.46%. Meanwhile, there is a 1.9% increase in the *mAP* value. In addition, the inference time of PRC-Light YOLO is 6.62 ms less than that of YOLOv7.

The variation curves of *mAP* values in seven groups of experiments are shown in Figure 8. Compared with YOLOv7, the number of parameters and *GFLOPs* of PRC-Light YOLO are reduced by 18.03% and 20.53% respectively, while the *mAP* value is improved by 7.6%. Experiments show that by improving the network structure of YOLOv7, the lightweight effect is significant; moreover, the efficacy of defect detection in fabrics using the enhanced model has significantly improved.

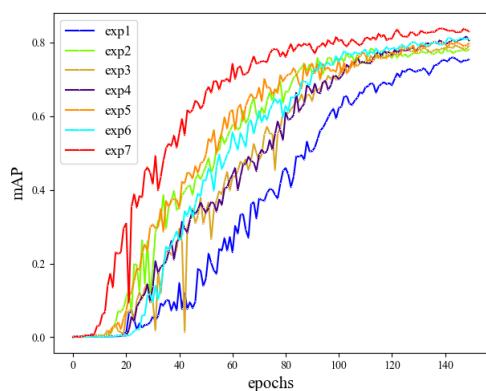
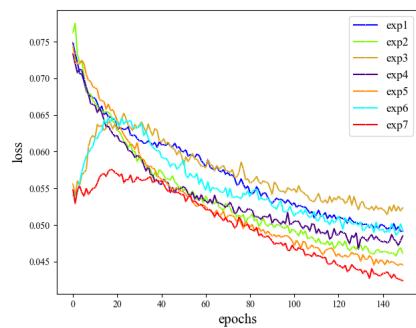
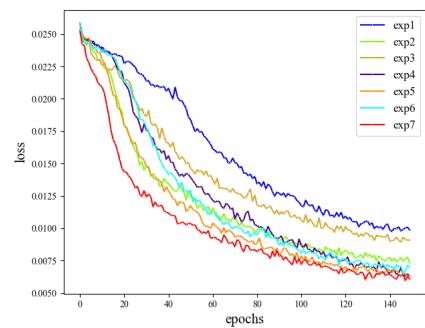


Figure 8. Change curve of mAP value.

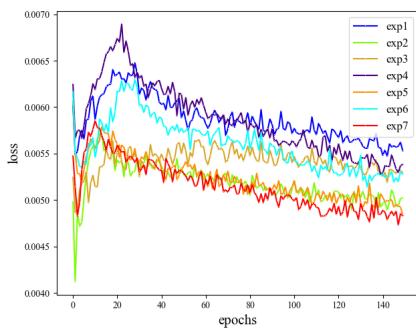
According to the loss curve chart above, it is evident that the PRC-Light YOLO model exhibits smaller loss values after convergence compared to other models. Specifically, experiments 3, 6 and 7 employ the Wise-IOU bounding box loss function, which dynamically adjusts the emphasis on different input regions over time. For anchor boxes with lower outliers, a smaller gradient boost is allocated, thereby directing the boundary regression towards normal-quality anchor boxes. This adjustment enhances the convergence speed of the network model. Figure 9 demonstrates that the PRC-Light YOLO model achieves lower loss values, thereby delivering superior performance in fabric defect detection.



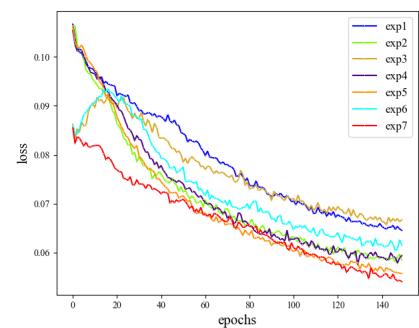
(a) Bounding box loss variation curve.



(b) Classification loss variation curve.



(c) Object confidence loss variation curve.



(d) The total loss function variation curve.

Figure 9. Loss Function. The loss function of the model consists of three parts, (a), (b) and (c), respectively, show the change curves of the BoxLoss, ClsLoss and ObjLoss for YOLOv7 and PRC-Light YOLO. (d) shows the total loss function variation curve.

According to Figure 10, it can be obtained that the mAP value of PRC-Light YOLO is increased from 75.4% at the beginning to 83%, and the experimental results verified

that the PRC-Light YOLO has improved the effect of defect detection. The AP values of threading, yard defects, stains and holes increased by 10.2%, 6.9%, 2.4% and 10.9%, respectively, with the most noticeable improvement in the detection of hanging warp and holes, and the AP values of the four categories of defects all are improved.

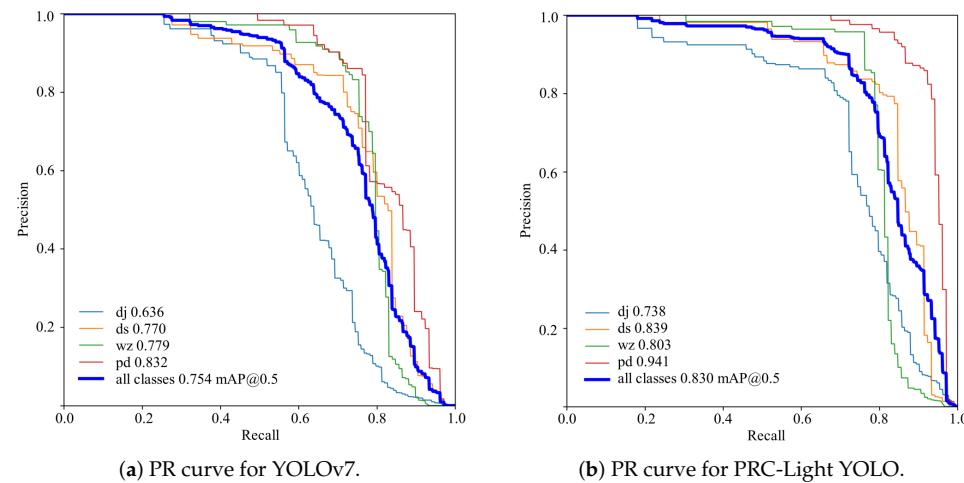


Figure 10. Precision–Recall. (a) and (b), respectively, show the PR curves of YOLOv7 and PRC-Light YOLO on the fabric dataset.

4.5. Comparison of Detection Effect

Figures 11 and 12 show the YOLOv7 and PRC-Light YOLO effects of four different types of defect detection, including the location of the defect by box selection, as well as above the box to display the type of defect and confidence. From the comparison in the figure, it can be observed that the PRC-Light YOLO detection model has improved the accuracy of fabric defect detection.

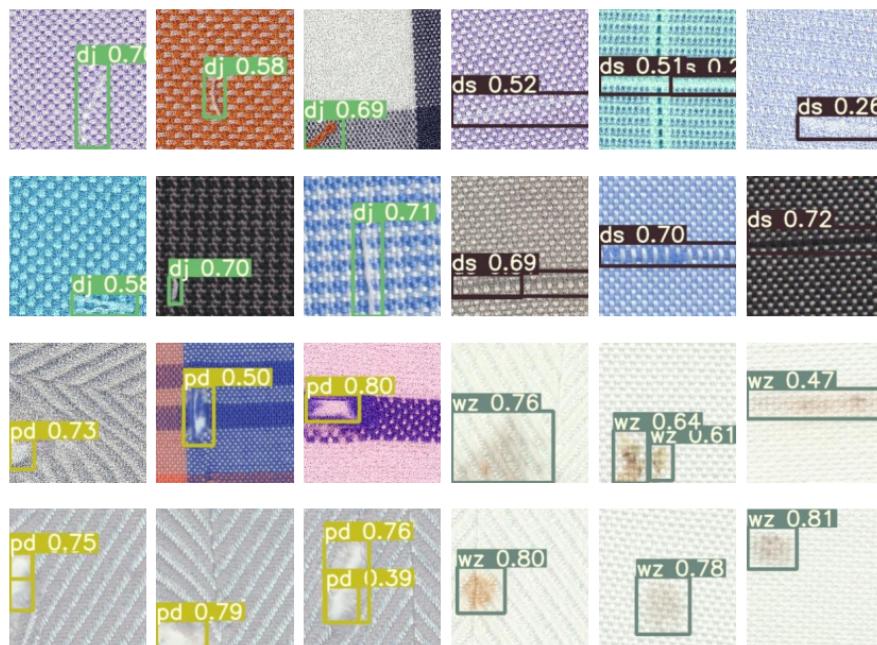


Figure 11. Detection effect of the YOLOv7.

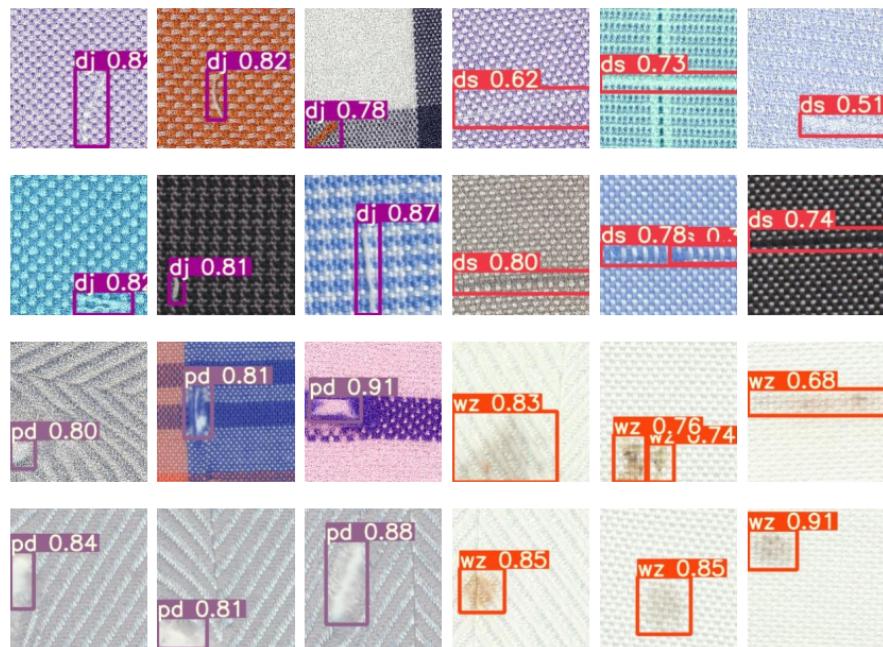


Figure 12. Detection effect of the PRC-Light YOLO.

4.6. Comparison Experiment

Through comparison experiments, the efficacy of various models can be assessed so that the better method can be selected. Therefore, we choose to conduct comparison experiments with Faster R-CNN, SSD, EfficientDet, CenterNet and YOLOv8x target detection models.

Table 3 shows a performance comparison of defect detection among various object detection models, with P , R , $F1$ -score, mAP and inference time selected as evaluation metrics.

Table 3. Defect detectability of different models.

Models	P	R	F1	Params/M	GFLOPs/G	mAP/%	Inference Time/ms
Faster R-CNN	0.784	0.491	0.6	136.98	370.3	0.696	177.35
SSD	0.832	0.529	0.647	105.2	87.41	0.731	143.58
EfficientDet	0.787	0.492	0.605	52.11	34.97	0.697	22.71
CenterNet	0.838	0.547	0.662	125	69.66	0.756	158.2
YOLOv7	0.826	0.697	0.76	37.21	105.2	0.754	32.08
YOLOv8x	0.754	0.754	0.754	68.23	258.5	0.784	40.39
PRC-Light YOLO	0.859	0.784	0.82	30.50	83.6	0.83	28.46

Table 4 shows the comparative results of different target detection models for the fabric defect dataset.

Table 4. Detection effects of different models.

Categories	Faster R-CNN	SSD	EfficientDet	CenterNet	YOLOv7	YOLOv8x	PRC-Light YOLO
Warp hanged	0.738	0.756	0.731	0.802	0.636	0.732	0.738
Yard defects	0.582	0.596	0.599	0.652	0.77	0.768	0.839
Stain	0.692	0.759	0.686	0.765	0.779	0.786	0.803
Hotel	0.773	0.814	0.773	0.804	0.832	0.850	0.941

It can be seen that the two-stage detection model Faster R-CNN has the lowest mAP value compared to the other detection algorithms; in addition, the two-stage detection model has one more step of operation compared to the single-stage model: the candidate

regions are first generated, and then the regions are classified and identified, which also causes the slower training speed and inference time of Faster R-CNN. Compared with the target detection models of Faster R-CNN, SSD, EfficientDet, EfficientDet, YOLOv7 and YOLOv8x, the precision of PRC-Light YOLO is improved by 7.5%, 2.7%, 7.2%, 2.1%, 3.3% and 10.5%. The recall is increased by 29.3%, 25.5%, 29.2%, 23.7%, 8.7%, 3%; the *F1-score* is increased by 20%, 17.3%, 21.5%, 15.8%, 6%, 6.6%, and params is decreased by 77.7%, 71%, 41.5%, 75.6%, 18%, 55.3%. In terms of *GFLOPs*, PRC-Light YOLO is lower than Faster R-CNN, SSD, YOLOv7 and YOLOv8x but higher than EfficientDet and CenterNet, whereas in terms of *mAP*, PRC-Light YOLO outperforms EfficientDet and CenterNet by 13.3% and 7.4%. In terms of inference time, EfficientDet is faster than PRC-Light YOLO, but it is lower than PRC-Light YOLO in terms of P, R, F1 as well as *mAP* values, and PRC-Light YOLO is faster than all other models except EfficientDet. Comparing with other models, it can be found that PRC-Light YOLO is effective at detecting the Hole, and the *AP* value reaches 94.1%. In the case of Warp hanged, the detection effect of the PRC-Light YOLO model is not satisfactory, and the *AP* value is only higher than that of EfficientDet, YOLOv7 and YOLOv8x. Therefore, in the subsequent experiments, the PRC-Light YOLO model should be optimized by taking into full consideration the characteristics of the Warp hanged. By comparing the *AP* values of each model for detecting yard defects and stains, PRC-Light YOLO has at least 25.7% and 11.7% higher detection accuracy than the other models. Comprehensive indications reveal that our proposed PRC-Light YOLO has an outstanding detection performance on the fabric dataset, which validates the effectiveness of the improved method.

5. Fabric Defect Detection System

In order to further illustrate that the model has a certain value for engineering applications and provide a more intuitive depiction of the detection process and detection results, we use PyQt5 to design a fabric defect detection system based on PRC-Light YOLO. The UI interface of the system is shown in Figure 13.

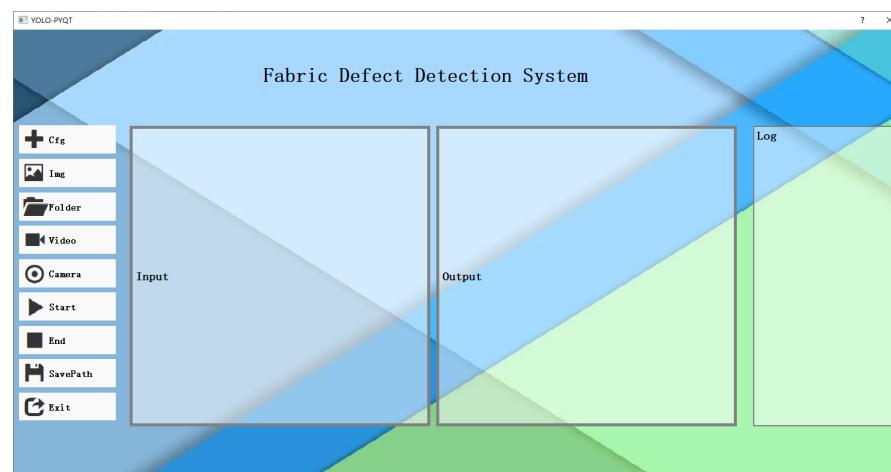


Figure 13. Detection system UI.

Clicking the “Cfg” button, select the necessary configuration file for the detection system. The file format should adhere to YAML format. The path to the chosen file will be displayed in the Statistical window. If the loading is successful, it will show “Load yaml success”. The application effect of the YOLO model is shown in Figure 14.

The YAML file contents include the path to the weight file, the category of the blemish and the confidence threshold and IOU threshold. The confidence threshold serves to filter the model’s output, displaying target boxes only when the detection probability exceeds this threshold. The IOU threshold primarily controls how the target detection algorithm handles overlapping targets. When two predicted bounding boxes intersect and both their

IOU values surpass the IOU threshold, the prediction box with lower confidence will be filtered out.

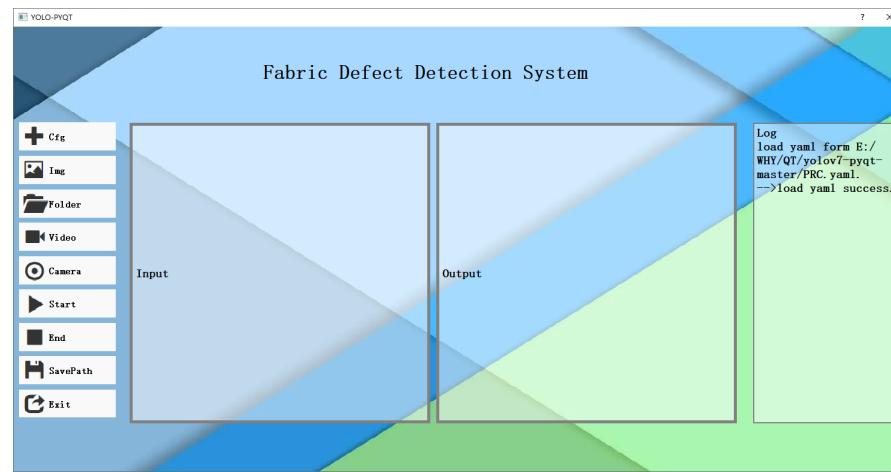


Figure 14. Application effect of the PRC-Light YOLO model.

After selecting the configuration file, you have the option to designate the path for saving the detection results. Additionally, the system is capable of processing images, videos and real-time data captured through a camera feed. The data path will be displayed in the Statistical window accordingly. By clicking the “Start Detection” button, the detection results will be presented to the right of the input image. The YOLO model detection effect is shown in Figure 15.



Figure 15. PRC-Light YOLO model checking effect.

In the Statistical window, you will also find information such as the time taken for detection, the path where the results are saved as well as the categories and quantities of defects detected. This information is essential for subsequent and timely manual inspection.

6. Discussion

The detection of defects in fabrics holds significant importance for ensuring fabric quality, reducing cost, enhancing production efficiency and preserving brand reputation. It is pivotal for the competitiveness and sustainable development of the fabric industry. This article presents a fabric defect detection model based on PRC-Light YOLO, which reduces the parameters of YOLOv7 while improving fabric defect detection accuracy. However, certain limitations still exist in some aspects.

(1) In the study of fabric defect detection, the size and variety of the dataset significantly impact the effectiveness of the model training. In practical industrial scenarios, fabric

backgrounds are often characterized by complexity and diversity and may also be affected by environmental factors such as light. However, the dataset used in the paper suffers from limitations such as a uniform background, limited quantity and insufficient diversity in defect types. Therefore, the exploration of methodologies for obtaining and constructing a substantial dataset of fabric defects becomes imperative. Furthermore, investigating fabric defects detection the background of intricate patterns and textures holds significant research significance so that the PRC-Light YOLO model can maintain robustness in real-world scenarios.

(2) In the research findings of this paper, it is evident that PRC-Light YOLO requires improvement in its capability to detect Warp hanged. Therefore, for the detection of minute defects, further investigation can be conducted into image preprocessing, network feature extraction capabilities or loss functions, with the aim of enhancing the precision of the network's detection capabilities.

(3) We have initially developed a textile defect detection system, which has interactivity and intuitiveness as its main goals. While this system introduces new perspectives for real-time fabric defect detection, it has not yet been integrated with hardware infrastructure. To achieve industrial applications, future efforts can involve further refining the detection platform and completing the setup of the system's hardware components. Additionally, we will collect user cases to evaluate the interactivity of the fabric defect detection system and continuously optimize it.

7. Conclusions

In response to the complex nature of defect types in fabrics, which vary in size, the computational intensity of detection models and the lower detection accuracy, this paper proposes the defect detection model PRC-Light YOLO. PRC-Light YOLO achieves network lightweight by replacing certain Conv layers with PConv layers in the Backbone. During the feature fusion stage, the RFB pyramid and the CARAFE upsampling operator are employed, and this approach enlarges the neural network's receptive field and enhances detection accuracy without significantly increasing the model's parameter count or computational load. PRC-Light YOLO adopts the HardSwish activation function, which exhibits piecewise characteristics that reduce computational costs and minimize memory access. Additionally, it utilizes Wise-IOU v3 as the bounding box loss function during model training, demonstrating robust performance in scenarios with varying dataset scales and highly imbalanced positive-to-negative sample ratios. Finally, we design a fabric defect detection system.

In order to improve the model's robustness and generalization ability, we extended the fabric dataset by applying diverse augmentation techniques, including random rotation, noise addition and adjustments to brightness, chromaticity and saturation. Ablation experimental results reveal that PRC-Light YOLO has a lower reduction in network model parameters by 18.03% and computations by 20.53% when compared with YOLOv7. Additionally, it showcases a 7.6% increase in *mAP*, enhancing the performance of detecting defects in fabrics. Furthermore, comparative experiments with other object detection models show that PRC-Light YOLO surpasses them in precision, recall, *F1-score* and *mAP* values. Although CenterNet outperforms PRC-Light YOLO by 6.4 percentage points in detecting Warp hanged, PRC-Light YOLO significantly outshines other models in detecting Yard defects, Stains and Holes with *AP* values consistently exceeding 80%. Notably impressive is the *AP* value of 94.1% achieved for detecting Holes.

To sum up, we propose the PRC-Light YOLO model, analyze the results of seven ablation experiments and seven comparative experiments, verify our model's excellent defect detection capability in fabrics and design a fabric defect detection system based on PRC-Light YOLO.

Author Contributions: Conceptualization, B.L. and H.W.; methodology, B.L. and H.W.; software, Z.C.; validation, H.W., Z.C. and Y.W.; investigation, H.W., L.T. and J.Y.; writing—original draft preparation, H.W.; writing—review and editing, H.W. and B.L.; supervision, B.L. and K.Z.; project administration and funding acquisition, B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61971339.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

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

Abbreviations

YOLO	You Only Look Once
IOU	Intersection Over Union
<i>mAP</i>	mean Average Precision
SSD	Solid State Drive
R-CNN	Region with CNN
ROI	Region of Interest
VGG	Visual Geometry Group
CARAFE	Content-Aware ReAssembly of FEatures
PConv	Partial Convolution
E-ELAN	Extended-Efficient Layer Aggregation Networks
RFB	Receptive Field Block
CBS	Convolution, Batch normalization, SiLU
MPConv	Max Pool Convolution
PAFPN	Path Aggregation Feature Pyramid Network
REP	Re-Parameterization
<i>FLOPs</i>	Floating-Point Operations
<i>FLOPS</i>	Floating-Point Operations Per Second
DWConv	Depthwise Convolution
GConv	Group Convolution
PWConv	PointWise Convolution
MAC	Memory Access Cost
ReLU	Rectified Linear
GELU	Gaussian Error Linear Unit
SiLU	Sigmoid Weighted Liner Unit
ObjLoss	Object Confidence Loss
ClLoss	Classification Loss
BoxLoss	Bounding Box Loss
CIOU	Complete Intersection Over Union
PE-ELAN	Partial convolution, Extended-Efficient Layer Aggregation Networks
CBH	Convolution, Batch normalization, HardSwish
<i>P</i>	Precision
<i>R</i>	Recall
<i>F1</i>	F1-score
<i>TP</i>	True Positive
<i>TN</i>	True Negative
<i>FP</i>	False Positive
<i>FN</i>	False Negative
<i>AP</i>	Average Precision
<i>GFLOPs</i>	Giga FLoating-Point Operations Per Second

References

- Zhang, H.; Qiao, G.; Lu, S.; Yao, L.; Chen, X. Attention-based Feature Fusion Generative Adversarial Network for yarn-dyed fabric defect detection. *Text. Res. J.* **2023**, *93*, 1178–1195. [[CrossRef](#)]
- Li, W.; Zhang, Z.; Wang, M.; Chen, H. Fabric Defect Detection Algorithm Based on Image Saliency Region and Similarity Location. *Electronics* **2023**, *12*, 1392. [[CrossRef](#)]
- Xiang, J.; Pan, R.; Gao, W. Yarn-dyed fabric defect detection based on an improved autoencoder with Fourier convolution. *Text. Res. J.* **2023**, *93*, 1153–1165. [[CrossRef](#)]
- Wu, J.; Li, P.; Zhang, H.; Su, Z. CARL-YOLOF: A well-efficient model for digital printing fabric defect detection. *J. Eng. Fibers Fabr.* **2022**, *17*, 15589250221135087. [[CrossRef](#)]
- Kanwal, M.; Riaz, M.M.; Ali, S.S.; Ghafoor, A. Saliency-based fabric defect detection via bag-of-words model. *Signal Image Video Process.* **2023**, *17*, 1687–1693. [[CrossRef](#)]
- Kahraman, Y.; Durmuşoğlu, A. Deep learning-based fabric defect detection: A review. *Text. Res. J.* **2023**, *93*, 1485–1503. [[CrossRef](#)]
- Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
- Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. 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; pp. 779–788.
- Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
- Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
- Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
- Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1916. [[CrossRef](#)]
- Zhao, J.; Zhou, S.; Zheng, Q.; Mei, S. Fabric defect detection based on transfer learning and improved Faster R-CNN. *J. Eng. Fibers Fabr.* **2022**, *17*, 15589250221086647.
- Zhang, J.; Jing, J.; Lu, P.; Song, S. Improved MobileNetV2-SSDLite for automatic fabric defect detection system based on cloud-edge computing. *Measurement* **2022**, *201*, 111665. [[CrossRef](#)]
- Zhou, S.; Zhao, J.; Shi, Y.S.; Wang, Y.F.; Mei, S.Q. Research on improving YOLOv5s algorithm for fabric defect detection. *Int. J. Cloth. Sci. Technol.* **2023**, *35*, 88–106. [[CrossRef](#)]
- Wu, Y.; Lou, L.; Wang, J. Cotton fabric defect detection based on K-SVD dictionary learning. *J. Nat. Fibers* **2022**, *19*, 10764–10779. [[CrossRef](#)]
- Lin, G.; Liu, K.; Xia, X.; Yan, R. An efficient and intelligent detection method for fabric defects based on improved YOLOv5. *Sensors* **2022**, *23*, 97. [[CrossRef](#)]
- Guo, Y.; Kang, X.; Li, J.; Yang, Y. Automatic Fabric Defect Detection Method Using AC-YOLOv5. *Electronics* **2023**, *12*, 2950. [[CrossRef](#)]
- Di, L.; Deng, S.; Liang, J.; Liu, H. Context receptive field and adaptive feature fusion for fabric defect detection. *Soft Comput.* **2023**, *27*, 13421–13434. [[CrossRef](#)]
- 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. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
- Liu, S.; Huang, D. Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 385–400.
- Baranwal, N.; Singh, K.N.; Singh, A.K. YOLO-based ROI selection for joint encryption and compression of medical images with reconstruction through super-resolution network. *Future Gener. Comput. Syst.* **2024**, *150*, 1–9.
- Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H. Designing network design strategies through gradient path analysis. *arXiv* **2022**, arXiv:2211.04800.
- Zhu, L.; Lee, F.; Cai, J.; Yu, H.; Chen, Q. An improved feature pyramid network for object detection. *Neurocomputing* **2022**, *483*, 127–139. [[CrossRef](#)]
- Chen, J.; Kao, S.H.; He, H.; Zhuo, W.; Wen, S.; Lee, C.H.; Chan, S.H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
- Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
- Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
- Tang, Y.; Han, K.; Guo, J.; Xu, C.; Xu, C.; Wang, Y. GhostNetv2: Enhance cheap operation with long-range attention. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 9969–9982.

29. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)]
30. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 13–19 June 2020; pp. 390–391.
31. Ding, P.; Qian, H.; Bao, J.; Zhou, Y.; Yan, S. L-YOLOv4: Lightweight YOLOv4 based on modified RFB-s and depthwise separable convolution for multi-target detection in complex scenes. *J. Real-Time Image Process.* **2023**, *20*, 71. [[CrossRef](#)]
32. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
33. Jiang, N.; Wang, L. Quantum image scaling using nearest neighbor interpolation. *Quantum Inf. Process.* **2015**, *14*, 1559–1571. [[CrossRef](#)]
34. Manzhos, S.; Ihara, M. Neural network with optimal neuron activation functions based on additive Gaussian process regression. *arXiv* **2023**, arXiv:2301.05567.
35. Polyzos, E.; Nikolaou, C.; Polyzos, D.; Van Hemelrijck, D.; Pyl, L. Direct modeling of the elastic properties of single 3D printed composite filaments using X-ray computed tomography images segmented by neural networks. *Addit. Manuf.* **2023**, *76*, 103786. [[CrossRef](#)]
36. Siddique, A.; Vai, M.I.; Pun, S.H. A low cost neuromorphic learning engine based on a high performance supervised SNN learning algorithm. *Sci. Rep.* **2023**, *13*, 6280. [[CrossRef](#)] [[PubMed](#)]
37. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
38. Sravya, N.; Lal, S.; Nalini, J.; Reddy, C.S.; Dell’Acqua, F. DPPNet: An efficient and robust deep learning network for land cover segmentation from high-resolution satellite images. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**, *7*, 128–139.
39. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-IoU: Bounding Box Regression Loss with Dynamic Focusing Mechanism. *arXiv* **2023**, arXiv:2301.10051.
40. Chen, M.; Yu, L.; Zhi, C.; Sun, R.; Zhu, S.; Gao, Z.; Ke, Z.; Zhu, M.; Zhang, Y. Improved faster R-CNN for fabric defect detection based on Gabor filter with Genetic Algorithm optimization. *Comput. Ind.* **2022**, *134*, 103551. [[CrossRef](#)]

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