


## Article

# A Lightweight Method for Peanut Kernel Quality Detection Based on SEA-YOLOv5

Zhixia Liu <sup>†</sup>, Chunyu Wang <sup>†</sup>, Xilin Zhong, Genhua Shi, He Zhang, Dexu Yang and Jing Wang <sup>\*</sup> 

College of Engineering, Shenyang Agricultural University, Shenyang 110866, China; 1994500009@syau.edu.cn (Z.L.); 15047631752@vip.163.com (C.W.); zxl13842231703@163.com (X.Z.); 16604069266@163.com (G.S.); 13842054723@163.com (H.Z.); yangdexu@syau.edu.cn (D.Y.)

\* Correspondence: wj77@syau.edu.cn

<sup>†</sup> These authors contributed equally to this work.

**Abstract:** Peanuts are susceptible to defects such as insect damage, breakage, germinant, and mildew, leading to varying qualities of peanuts. The disparity in peanut kernel quality results in significant differences in their prices and economic value. Conducting real-time, accurate, and non-destructive quality inspections of peanut kernels can effectively increase the utilization rate and commercial value of peanuts. Manual inspections are inefficient and subjective, while photoelectric sorting is costly and less precise. Therefore, this study proposes a peanut kernel quality detection algorithm based on an enhanced YOLO v5 model. Compared to other models, this model is practical, highly accurate, lightweight, and easy to integrate. Initially, YOLO v5s was chosen as the foundational training model through comparison. Subsequently, the original backbone network was replaced with a lightweight ShuffleNet v2 network to improve the model's ability to differentiate features among various types of peanut kernels and reduce the parameters. The ECA (Efficient Channel Attention) mechanism was introduced into the C3 module to enhance feature extraction capabilities, thereby improving average accuracy. The Ciou loss function was replaced with the alpha-IoU loss function to boost detection accuracy. The experimental results indicated that the improved model, SEA-YOLOv5, achieved an accuracy of 98.8% with a parameter count of 0.47 M and an average detection time of 11.2 ms per image. When compared to other detection models, there was an improvement in accuracy, demonstrating the effectiveness of the proposed peanut kernel quality detection model. Furthermore, this model is suitable for deployment on resource-limited embedded devices such as mobile terminals, enabling real-time and precise detection of peanut kernel quality.

**Keywords:** peanut kernel; quality testing; YOLOv5; lightweight; attention mechanism



**Citation:** Liu, Z.; Wang, C.; Zhong, X.; Shi, G.; Zhang, H.; Yang, D.; Wang, J. A Lightweight Method for Peanut Kernel Quality Detection Based on SEA-YOLOv5. *Agriculture* **2024**, *14*, 2273. <https://doi.org/10.3390/agriculture14122273>

Academic Editors: Gijs A. Kleter and Perla A. Gómez

Received: 27 September 2024  
Revised: 10 December 2024  
Accepted: 10 December 2024  
Published: 11 December 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

As a pivotal oilseed and economic crop, peanuts occupy a prominent position in Chinese agriculture, with their export volume standing out among numerous agricultural products in the country. They are one of the few agricultural products whose exports exceed imports [1]. During storage and transportation, peanuts are highly susceptible to breakage, insect infestation, damage, and mold growth due to factors such as humidity, temperature, and other uncontrollable elements. Ensuring the appearance quality of peanut kernels is crucial, as it not only affects their edible value and market price, but also directly relates to consumer health and safety. Moldy peanuts produce a carcinogen that poses a serious threat to human health. Therefore, strictly controlling the quality of peanuts is a crucial step in ensuring food safety and maintaining public health [2–4]. Manual sorting is highly subjective, inefficient, and costly; photoelectric sorting requires high technical proficiency from operators, has poor adaptability to environmental factors, low accuracy, and is costly. Solving these problems requires a more efficient, accurate, and economical detection method [5].

With the rapid development of artificial intelligence technology in recent years, target detection based on deep learning has become the focus of scientific research and attracted much attention. Target detection models based on deep learning are mainly divided into two main categories: the first category includes candidate region-based two-stage target detection models, and the mainstream models are R-CNN [6], Faster R-CNN [7], Mask R-CNN [8], and so on; the other category includes regression-based one-stage target detection models, which omits the candidate region generation step, and can directly extract features, and the mainstream models are SSD [9], YOLO [10–15], etc. These models have been widely used in the field of agriculture, such as pest and disease identification and detection of plant leaves, maturity detection of various types of fruits, fruit grading quality detection, seed grading detection, and weed detection. Gai et al. [16] proposed a deep learning algorithm based on the improved YOLO v4 deep learning algorithm for cherry fruit detection, with an average accuracy value of 0.15% improvement over the original YOLO v4 model. Rai et al. [17] built a model based on the improved YOLO v7-tiny architecture for weed detection using aerial images and video crops, demonstrating a 0.6% accuracy improvement and five-times faster speed. Wang et al. [18] built an improved Faster R-CNN model as an engineering tool for tomato ripening detection with an average detection accuracy of 96.14%. Shi et al. [19] proposed a lightweight model OMC-YOLO based on improved YOLOv8n for automatic detection and grading of oyster mushrooms with an average accuracy value of 94.98% and a 26% reduction in the number of parameters and computation. Bhupendra et al. [20] proposed a method for classification of pre-processed rice grains based on a CNN model with an overall classification accuracy of 98.37%. You et al. [21] developed an apple target detection model for fruit thinning period with an average accuracy of 95.2%. Xiong et al. [22] proposed a lightweight YOLO v5-Lite model for detecting papaya ripeness in natural environments with an average detection accuracy of 92.4% and a model memory footprint of only 11.3 MB.

Currently, many scholars are engaged in research pertaining to peanut quality testing. Yang et al. [23] proposed a convolutional neural network (CNN) for peanut pod quality detection. The proposed algorithm is based on an improved version of ResNet, achieving an accuracy of 98.1% with a parameter size of only 32.63 M. Wu et al. [24] employed hyperspectral imaging in conjunction with a hybrid learning model (SEL) and an optical gradient boosting algorithm to achieve non-destructive detection of peanut mold, achieving an accuracy of 99%. Yang et al. [25] obtained peanut pod images of 12 varieties through a scanner. Concurrently, the enhanced deep CNN, VGG16, was employed for the recognition and classification of the peanut dataset, with an average accuracy of 96.7%. Yang et al. [26] employed a multi-target recognition method with a multispectral system and an enhanced version of Faster R-CNN for the detection of peanut defects, achieving an average accuracy of 99.97%. Zhang et al. [27] proposed a peanut kernel quality defect recognition system based on machine vision and an adaptive CNN. The system achieved an average recognition rate of 99.7%.

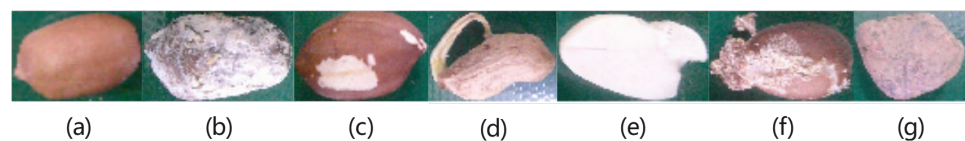
The analysis demonstrates that deep learning detection algorithms have been effectively utilized in the field of peanut quality detection. However, there are limitations, including a limited number of detection categories, a high number and volume of parameters, and a lack of applicability to embedded systems. To address the aforementioned issues, this study presents a detection algorithm based on enhanced YOLOv5 (SEA-YOLOv5), which aims to achieve low-cost, high-precision, lightweight, and rapid peanut kernel quality detection. The research presented here enhances the performance of the model in the peanut kernel quality detection task, effectively improving the accuracy and efficiency of peanut kernel quality detection. This ensures the quality and safety of agricultural products and provides a technical reference for other similar tasks. The scientific community can further optimize and extend the model to meet target detection needs in more complex scenarios, which will promote the further development of agricultural intelligent technology, improve the efficiency of agricultural production, and enhance quality and safety levels.

This will facilitate the joint exploration of innovative solutions to promote the synergistic development of related fields.

## 2. Materials and Methods

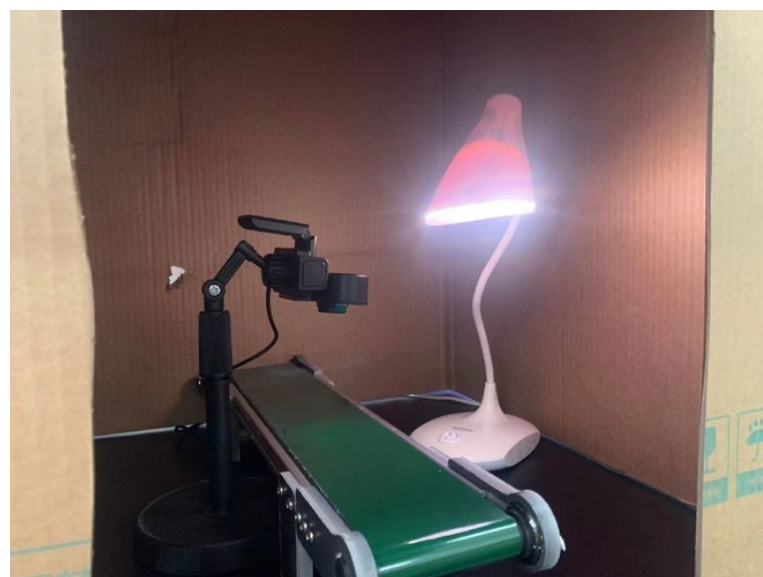
### 2.1. Production and Construction of Peanut Kernel Dataset

The peanut variety examined in this experiment was Qinghua 308 (medium grain type). A total of 300 peanuts were randomly selected for hulling from a large number of peanut samples, and about 100 full and undamaged peanut kernels were selected as the complete sample group. The remaining peanut kernels were equally divided into 5 parts. The samples were divided into a germination sample group, an insect erosion sample group, a breakage sample group, a broken sample group and a mildew sample group. All sample groups were created to ensure the same environment and the same variables. Finally, a soil/stone sample group was added, bringing the total to 7 sample groups. Representative samples of the various categories are illustrated in Figure 1.



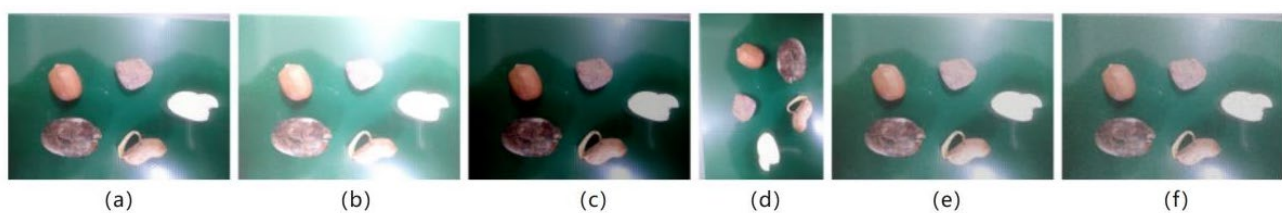
**Figure 1.** Comparison of samples in different categories: (a) complete; (b) mildew (c) breakage (d) germinant (e) broken (f) insect erosion; and (g) soil/stone.

To ensure consistency in the data acquisition process, this experiment was conducted using a custom-built peanut kernel data-acquisition test bench, as illustrated in Figure 2. The test bench comprises a dark box, a CHUBU DC311L camera, a light source, and a conveyor belt. The camera is positioned at a height of 150 mm, while the light source is situated at a height of 220 mm. In the conveyor belt, a rectangular box with dimensions of 80 mm by 80 mm was used to draw 4–5 peanut kernels or stones, which were randomly selected from the seven sample groups. The kernels or stones were placed in the rectangular box in a random combination and random spacing, and the conveyor belt was driven at a rate of 40 mm/s. The peanut kernels were rolled and conveyed forward by a multi-cone roller and conveyor belt, with the camera operating at a rate of 2 frames per second. This process yielded 2323 images, which were then screened to remove those with blurred content. Ultimately, 2000 images were selected for further analysis.



**Figure 2.** Peanut kernel data-acquisition test bed.

In this experiment, the LabelImg tool was employed to annotate the original images with rectangles; to designate the complete peanut kernel labels as complete, breakage, broken, mildew, germinant, insect erosion, and soil/stone; and to generate XML (Extensible Markup Language)-type markup files. The labeled data were randomly divided into a training set and a validation set according to an 8:2 ratio. To simulate the various luminance and motion blurring effects observed in the actual production process, as well as other factors that can affect the efficacy of model training and generalization performance, resulting in overfitting or underfitting, it is necessary to enhance the robustness and accuracy of the model. This can be achieved through the incorporation of a pretraining phase. The data set was augmented in five ways: introduction of salt-and-pepper noise, brightness adjustment, exposure adjustment, Gaussian noise, and random rotation. This resulted in an additional 10,000 images being included in the data set. The impact of this augmentation is illustrated in Figure 3.



**Figure 3.** Examples of peanut kernel images: (a) original image; (b) increased exposure; (c) changed brightness; (d) flip; (e) Gaussian noise; (f) salt-and-pepper noise.

## 2.2. Peanut Kernel Detection Model

### 2.2.1. YOLOv5 Model

YOLOv5 is an advanced single-stage target detection algorithm. As shown in Figure 4, at the input end, the mosaic data enhancement method is used to scale, crop, arrange, and then splice four images randomly, which greatly enriches the dataset and enhances the generalization of the model. The accuracy and robustness of the model are improved by calculating the optimal anchor frame parameters with an adaptive anchor frame. The backbone network is primarily employed for the extraction of features in instance segmentation. This is achieved through the use of convolutional modules (CBS, C3), CSP1\_X and SP2\_X modules, and spatial pyramid pooling. The SPPF module generated by the SPP [28] structure changes the three parallel maximum pooling layers in SPP into serial connections to reduce the computational cost of multi-feature fusion. The neck network is structured in a manner that is analogous to that of a feature pyramid network, with the addition of a path aggregation network. FPN conveys semantic information from top to bottom, and PAN integrates positional information from bottom to top. Thus, more abundant feature information can be obtained. The prediction terminal predicts three different size feature maps, adopts the CIoU function as the boundary-frame loss function, obtains the best prediction frame through non-maximum suppression, and the final output comprises the category and confidence of the predicted target, together with its location information. Compared with the previous generations of the YOLO series, YOLOv5 pays more attention to the detection of small targets, and has higher accuracy. While maintaining higher precision, YOLOv5 has smaller weight files and faster training and reasoning speed. According to different network depths and widths, YOLOv5 is available in four versions, namely, YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. The width and depth of these models are different, so that they can adapt to different application requirements.

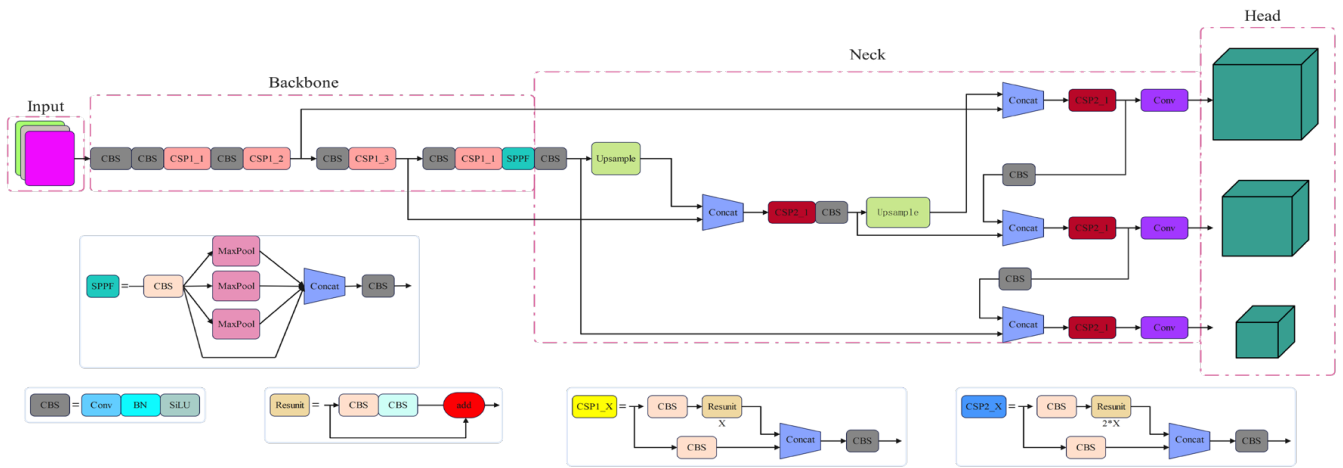


Figure 4. YOLOv5 model structure.

When training the peanut kernel dataset, the original YOLOv5 algorithm has the problems of insufficient training accuracy and low recognition rate. The models of different YOLOv5 series versions were tested on the peanut kernel data set, and the mean average precision (mAP), parameters, floating-point operations per second (FLOPs), frames per second (FPS), and size of model were evaluated to select the most appropriate YOLOv5 version. Table 1 shows the test results. Among the four versions, the average accuracy of YOLOv5s is slightly lower than that of the other three, but the parameters, size of model, and FLOPs are the lowest and the FPS is the highest. Considering the model complexity and detection accuracy, size of model is crucial for embedded devices. Compared with YOLOv5x, which has the best detection effect, the floating-point number, parameters, and size of model are greatly reduced and the FPS is greatly increased when the average accuracy of YOLOv5s is only reduced by 1.1 percentage points. Therefore, the YOLOv5s model is finally selected as the basic model for subsequent improvement in this study.

Table 1. Comparison of different YOLOv5 versions.

Model	Depth Multiple	Width Multiple	Parameters (M)	Size of Model (MB)	Precision (%)	Recall (%)	mAP0.5 (%)	FLOPs (G)	FPS (s – 1)
YOLOv5s	0.33	0.5	7.07	13.7	96.1	96.5	0.984	16.3	158.73
YOLOv5m	0.67	0.75	21.07	40.8	97.7	97.8	0.988	50.6	80.45
YOLOv5l	1.00	1.00	46.65	89.3	97.9	97.9	0.992	114.6	70.32
YOLOv5x	1.33	1.25	87.27	166	99.4	99.7	0.995	217.9	59.63

2.2.2. ShuffleNetv2: Lightweight Backbone Network

YOLOv5s uses CSPDarknet53 as the backbone network for image feature extraction, including CBL, FOCUS, CSP, and SPP modules. The backbone network includes multiple deep convolutional modules, which enhances the feature-extraction capability of the model. Nevertheless, this approach also results in a larger size of model, increased model complexity, and challenges in terminal device deployment. To improve the lightweight structure of the YOLOv5s network for detecting peanut kernel appearance quality, ShuffleNetv2 is introduced in Backbone to strengthen the model’s effective extraction ability of similar features of multiple categories of peanut kernel. ShuffleNetv2 is a lightweight network proposed by MegVII (Beijing, China) in July 2018, which achieves a balance of speed and accuracy. Different from ShuffleNetv1, ShuffleNetv2 was designed as an efficient network with both speed and accuracy in mind under the G1–G4 criteria, as far as was possible [29,30].

The ShuffleNetV2 building block consists of a base unit and a downsampling unit (Figure 5). The fundamental unit presents the channel splitting operation, which divides the input feature graph into two branches along the channel dimension. The number of channels is represented as  $C'$  and  $C-C'$ . The left branch executes an equal mapping, whereas the right branch performs three continuous convolutions in one-step intervals, resulting in an output channel that is identical to the input channel. Two  $1 \times 1$  convolutions are standard convolution, and one  $3 \times 3$  convolution is deep convolution. As a key component in lightweight networks [31–33], deep convolution, in conjunction with the subsequent  $1 \times 1$  convolution, constitutes deep separable convolution. This approach markedly diminishes the parameters and computational costs associated with the convolutional layer, while enhancing the feature interaction process. The outputs of the two branches are then concatenated and subsequently shuffled to enhance the process of information fusion between the channels. Compared with the base unit, the left branch of the downsampling unit is successively increased by  $3 \times 3$  depth convolution and  $1 \times 1$  standard convolution, and the step length of the depth convolution in the two branches is 2. Channel segmentation is omitted, the number and width of network channels are directly increased, and the feature-extraction capability of the network is enhanced.

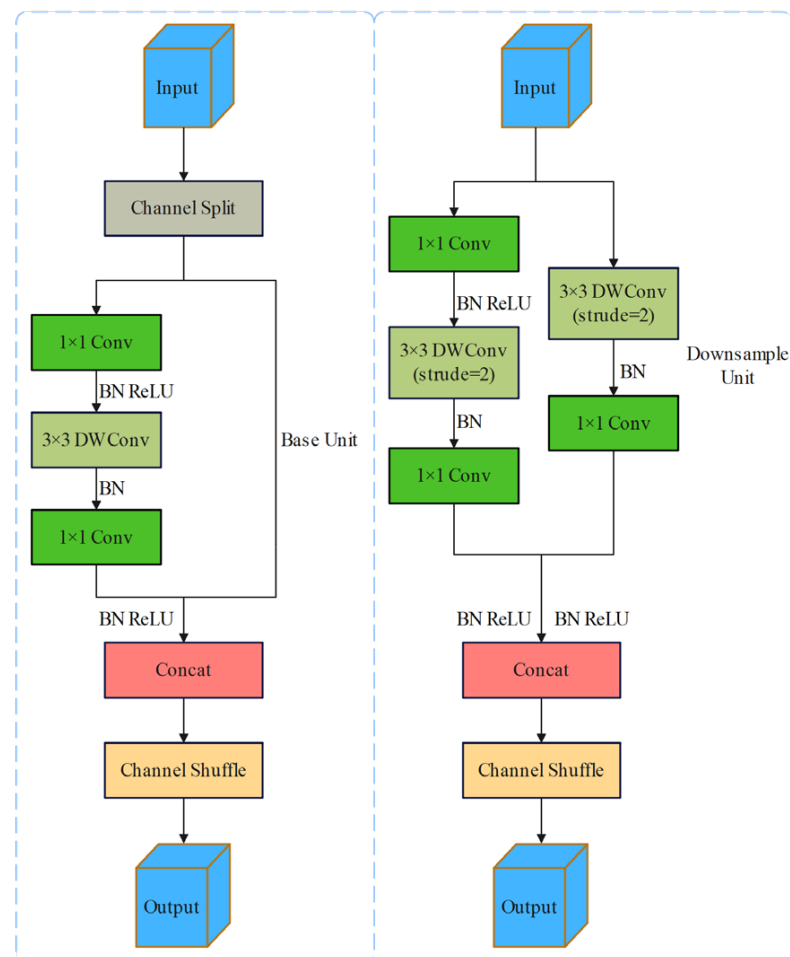


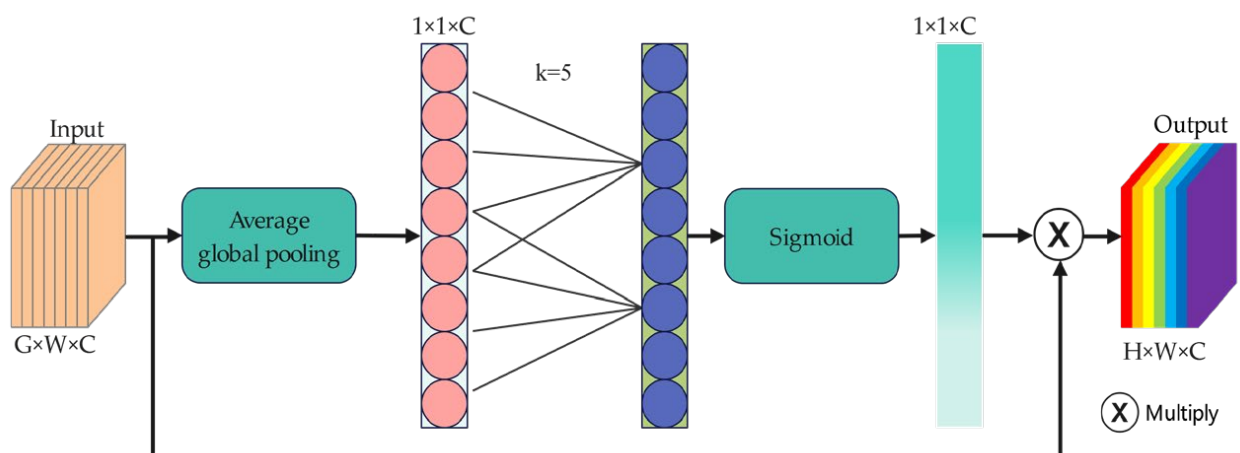
Figure 5. ShuffleNetV2 building block.

### 2.2.3. ECA Mechanism

The alteration of the illumination and the environmental impact of the testing apparatus will impair the network's capacity to extract pivotal data, which will consequently impact the model's performance. At the same time, the ShuffleNetv2 module will reduce the number of parameters, but also slightly reduce the model performance. To solve the

above problems and improve the attention paid by the model to peanut kernels, the ECA mechanism was added to the model network structure.

The ECA mechanism [34] module (Figure 6) removes the full connection layer in the squeeze and excitation (SE) [35] module. After the global average-pooling operation, features are learned through a one-dimensional convolution. The ECA mechanism proposes a local cross-channel interaction strategy without dimensionality reduction, which effectively avoids the impact of dimensionality reduction on channel attention learning. The ECA module uses a fast 1D convolution kernel of size  $k_1$ , and uses an adaptive method to determine  $k_1$ , where the coverage of the cross-channel interaction (the size of the 1D convolution kernel  $k_1$ ) is proportional to the channel dimension, thereby avoiding manual adjustment of the  $k_1$  value. Proper interaction across channels can significantly reduce the complexity of the model, while maintaining performance. Through the adjustment of a few parameters, an obvious effect gain can be obtained, and the effect on the speed of network processing is small.



**Figure 6.** ECA mechanism structure diagram.

#### 2.2.4. Alpha-IoU Loss Function

The loss function of bounding box regression plays an important role in target detection, which can significantly improve the performance of the model. In object detection tasks, intersection over union (IoU) is widely used to measure the overlap between the prediction box and the real label box. Its structure is shown in Figure 7.

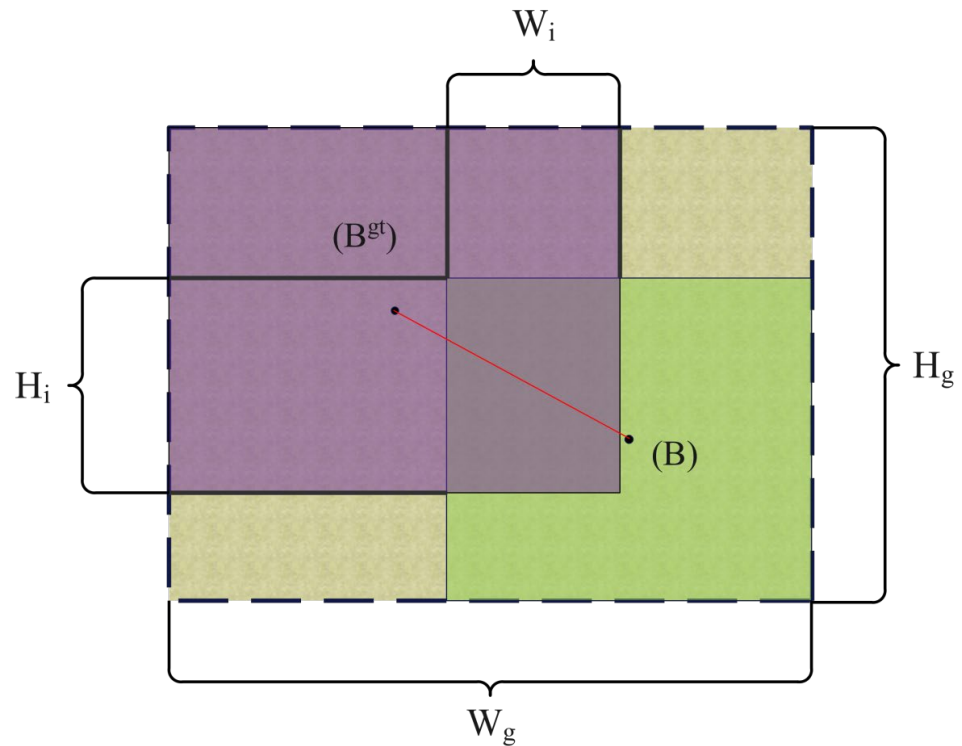
Supposing that the predicted detection box  $\vec{B} = [x, y, w, h]$  and the real detection box  $\vec{B}_{gt} = [x_{gt}, y_{gt}, w_{gt}, h_{gt}]$ , where  $W_g$  and  $H_g$  represent the width and height of the minimum bounding box, respectively, the function of IoU loss is defined as

$$\text{Loss}_{\text{IoU}} = 1 - \text{IoU} = 1 - \frac{W_i H_i}{S_u} \quad (1)$$

where  $S_u$  refers to the union of the prediction box and the real box. When the IoU has no intersection between the prediction box and the real box, the backpropagation gradient of  $\text{Loss}_{\text{IoU}}$  will disappear. To solve this problem, a penalty term is added to the subsequent loss function.

YOLOv5s uses CIoU as the bounding box-regression loss function, while GIoU is a method proposed by Rezatofighi et al. to optimize bounding boxes [36]. CIoU loss function is designed based on IoU, which considers the position, scale, aspect ratio and other factors between the predicted box and the real box. CIoU minimizes the distance between the predicted box and the real box, and backpropagates it as a loss value. When the prediction frame and the real frame exhibit a high degree of overlap (very large IoU) or a low degree of overlap (very small IoU), resulting in a denominator of CIoU approaching 0 or a numerator

approaching the denominator, it will lead to an inability to reflect the precise position of the prediction frame and the real frame, which will affect the stability of the model and the convergence speed.



**Figure 7.** IoU loss-function prediction box and real box. Note: the green part of the figure is the prediction box, the purple part is the real box;  $W_g$  and  $H_g$  represent the width and height of the minimum bounding box, respectively;  $W_i$  and  $H_i$  represent the width and height of the overlap between the real box and the prediction box, respectively;  $B$  and  $B^{gt}$  represent the center points of the prediction and real boxes, respectively.

To address the issues associated with the calculation of the loss in CIoU, alpha-IoU [37] is employed in lieu of conventional IoU. Alpha-IoU incorporates a greater number of penalty terms than IoU and employs the power exponential form to represent the loss function, thereby facilitating adaptability to varying degrees of bounding box-regression precision [38]. Additionally, alpha-IoU incorporates an adjustable hyperparameter, alpha, which can be calibrated to align with varying training requirements. Empirical evidence demonstrates that the alpha-IoU loss function outperforms all existing IoUs, exhibiting enhanced regression precision and resilience.

Specifically, the alpha-IoU and CIoU loss functions have the following general forms:

$$Loss_{\alpha-IoU} = 1 - \left(\frac{WiHi}{S_u}\right)^\alpha + \tau^\alpha(B, B^{gt}) \tag{2}$$

$$Loss_{CIoU} = 1 - \frac{WiHi}{S_u} + \frac{\rho^2(B, B^{gt})}{W_g^2 + H_g^2} + \beta v \tag{3}$$

Combining Equations (2) and (3), the alpha-CIoU loss function can be summarized as follows:

$$Loss_{\alpha-CIoU} = 1 - \left(\frac{WiHi}{S_u}\right)^\alpha + \frac{\rho^{2\alpha}(B, B^{gt})}{(W_g^2 + H_g^2)^\alpha} + (\beta v)^\alpha \tag{4}$$

where  $B$  and  $B^{gt}$  represent the center point of the prediction box and the real box, respectively;  $\tau$  represents the penalty term;  $\rho$  is the Euclidean distance; and  $\alpha \in N(N = 1, 2, 3, \dots)$ .



$\beta$  represents the balance proportion coefficient, and  $v$  is the parameter considering the proportion consistency of the two frames:

$$\beta = \frac{v}{(1 - \text{IoU}) + v} \tag{5}$$

$$v = \frac{4}{\pi^2} \left( \arctan \frac{w}{h} - \arctan \frac{w^{gt}}{h^{gt}} \right)^2 \tag{6}$$

where  $w$ ,  $w^{gt}$ ,  $h$  and  $h^{gt}$  represent the width and length of the two boxes, respectively. In the specific training process, alpha-IoU shows robustness for different models and datasets. In most cases, if the value of alpha-IoU is 3, excellent performance can be maintained in the training.

### 2.2.5. Network Structure of SEA-YOLOv5

In this study, the architecture of the YOLOv5 version was established as the basis for model improvement. The goal was to address issues related to accuracy, size of model, and detection speed to develop a more appropriate model for detecting peanut kernel-category targets at the primary processing stage. The backbone network in YOLOv5 comprises multiple deep convolutional modules, which serve to enhance the model’s feature extraction capability while simultaneously increasing the model’s size, complexity, and the difficulty of terminal device deployment. Accordingly, lightweight ShuffleNetV2 building blocks are employed to supplant the backbone network, thereby reducing the model’s overall complexity. We also introduced the convolutional attention module ECA in the backbone network layer to compensate for the loss of detection accuracy caused by lightweight feature-extraction networks. Finally, the alpha-IoU loss function is used as a measure of the prediction box-regression loss term to improve inference accuracy. This adjustment is made to improve the input-specific data distribution and ultimately enhance feature detection at different image scales. The comprehensive, augmented network configuration is illustrated in Figure 8. CBRM represents a convolutional module comprising convolution, batch normalization, rectifying linear unit (ReLU), and maximum pooling. SN\_Block\_X represents a network structure in which a subsampled unit is connected in series with X repeating base units from the ShuffleNetV2 building block.

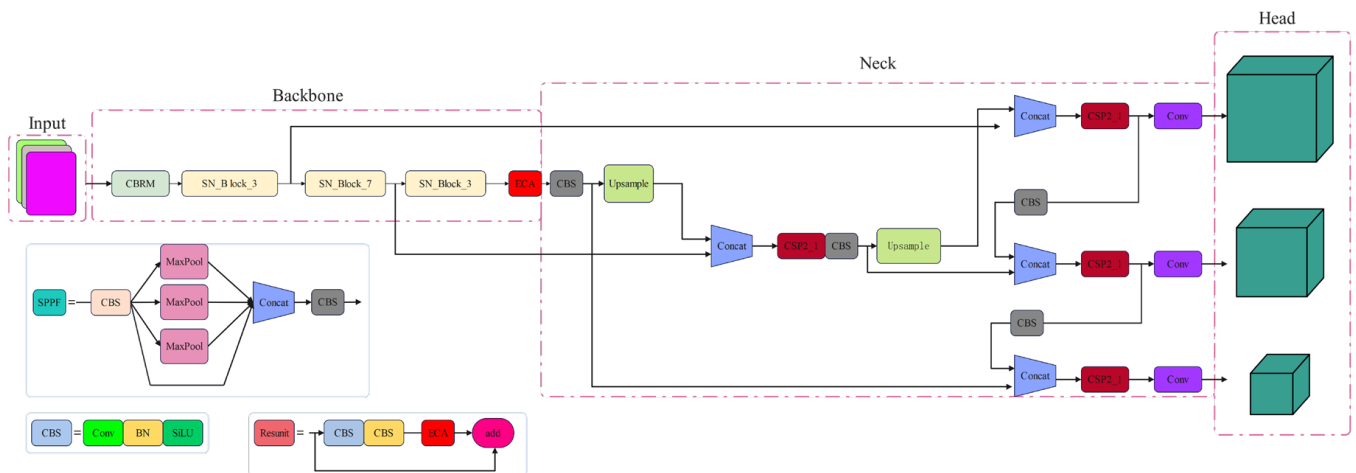


Figure 8. Network structure of SEA-YOLOv5.

## 3. Results

### 3.1. Test Environment and Parameter Setting

The computer test environment adopted in this study was as follows: the operating system was Windows10 (64 bit), the processor model was the 12th generation Intel Core i7-12700F, the working frequency was 2.10 GHz, the graphics card model was NVIDIA

GeForce RTX3060Ti, the graphics memory was 8 GB, and the programming language was Python3.8. The deep learning framework was PyTorch1.9.0, and a CUDA 11.1 parallel computing framework was adopted; the CUDNN version was 8.05. Starting from the initial learning rate of 0.01, the cosine annealing technique was used for subsequent reduction. The parameter optimization of the neural network adopted the stochastic gradient descent (SGD) algorithm, the momentum parameter was set to 0.937, the weight attenuation factor was set to 0.0005, the batch size used in each iteration training was set to 16, and the epoch of training was set to 200, so as to avoid over-fitting and reducing the training time. The subsequent comparison tests remained unchanged, and the input image was adjusted to  $640 \times 640$  pixels.

### 3.2. Evaluation Index

The classification and recognition model can be evaluated with accuracy (A), precision (P), recall (R), confusion matrix, etc. Target detection network-evaluation indicators also include PR (precision–recall) curve, average precision (AP), and mAP, etc. The A, P, and R rate are calculated as follows:

$$A = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (7)$$

$$P = \frac{T_P}{T_N + F_P} \quad (8)$$

$$R = \frac{T_P}{T_P + T_N} \quad (9)$$

where  $T_P$  (true positive) means that the prediction result of the positive sample is positive;  $T_n$  (true negative) indicates that the negative sample result is predicted to be negative;  $F_P$  (false positive) indicates that the prediction result of negative samples is positive; and  $F_n$  (false negative) indicates that the prediction result of the positive sample is negative. Accuracy represents the proportion of correctly identified samples in all samples, precision represents the proportion of positive samples in all positive samples in the prediction result, and recall rate represents the proportion of correctly identified positive samples in all positive samples in the test set.

The PR curve represents the relationship between precision and recall rate. In the PR curve, recall is the horizontal coordinate and precision is the vertical coordinate. The AP value is the area enclosed by the PR curve and coordinate axis, which is used to measure the detection performance of a single target in the model. The better the performance of the classifier, the higher the AP value. In multi-class target detection tasks, mAP is used for evaluation. mAP is the average AP of each detection class, which is used to measure the performance of the whole model. The formula is as follows:

$$AP = \int_0^1 P \cdot (R) dR \quad (10)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (11)$$

where N represents the number of detection categories, and the value in this study is 7.

### 3.3. Performance Comparison of Different Backbone Networks

To demonstrate the advantages of using the ShuffleNetV2 module over other improved backbone networks, several comparative tests were carried out. As shown by the test results in Table 2, the reduction in the number of convolutions resulting from the lightweight nature of the backbone network leads to a corresponding reduction in model depth. This, in turn, has the effect of slightly reducing the feature extraction capability of the network, which is manifested in a reduction in mAP0.5 for all segmentation models with lightweight

backbones in comparison to the original YOLOv5s model. Nevertheless, the mAP0.5 decline was no greater than 4 percentage points. In the lightweight approach, the ShuffleNet2 building block is employed to enhance the backbone network, thereby achieving the optimal lightweight performance. When the precision, recall, and mAP0.5 exhibit minimal discrepancy in comparison to other lightweight methodologies, the number of parameters is diminished, exhibiting a reduction of 93.7% in comparison to the initial segmentation model. Additionally, the size of the model is reduced to merely 9.7% of the original segmentation model, thereby fulfilling the requisite criteria for lightweight implementation.

**Table 2.** Performance comparison of different backbone networks.

	YOLOv5s	YOLOv5- Shuffle Netv2	YOLOv5- Mobile Netv3	YOLO- Ghost Net
Parameters/M	7.07	0.45	3.55	5.1
FLOPs/G	16.3	6.1	6.3	10.8
Size of model/MB	14.4	1.4	7.4	10.5
FPS/(s – 1)	158.73	169.49	140.85	149.25
Precision/%	96.1	95.4	89.5	95.3
Recall/%	96.5	95.2	92.3	95.2
mAP0.5/%	98.4	97.6	94.3	97.5

### 3.4. Performance Comparison of Attention Mechanisms

#### 3.4.1. Performance Comparison of Various Attention Mechanisms

To evaluate multiple attention mechanisms, the backbone network of YOLOv5s was used as the baseline model, with different attention mechanisms introduced into the model separately, to prove their effectiveness by comparing their respective evaluation metrics. The ECA mechanism achieves efficient computation of channel attention, while reducing the complexity of the model by avoiding fully connected layers and utilizing adaptive 1D convolution. ECA can adaptively adjust the size of the convolution kernel, thereby enabling the efficient capture of cross-channel dependencies, while reducing redundant computation. The results demonstrate that the mAP0.5 and FPS of the models with distinct attention mechanisms are improved. The model with an ECA module exhibits the most pronounced increase in mAP0.5, with a 0.7 percentage point enhancement relative to the original model. The modified model of YOLOv5s achieves an mAP0.5 of 99.1%, exhibiting the most significant improvement in this metric. Additionally, the FPS value demonstrates the highest increase, reaching 6.8% higher than that of the original model. The results of the comparison test are presented in Table 3.

**Table 3.** Comparison of performance of different attention mechanisms.

	YOLOv5s	YOLOv5- ECA	YOLOv5-SE	YOLOv5-CA	YOLOv5- CBAM
Parameters/M	7.07	7.22	7.29	7.28	7.29
FLOPs/G	16.3	16.3	16.8	16.7	16.9
Size of model/MB	14.4	14.7	14.9	14.8	14.9
FPS/(s – 1)	158.73	169.49	135.14	142.85	158.73
Precision/%	96.1	97.1	96.6	96.7	94.8
Recall/%	96.5	98.3	96.9	96.9	97.1
mAP0.5/%	98.4	99.1	98.7	98.8	98.6

### 3.4.2. Comparison of Performance at Different Locations of the Attention Mechanism

In order to investigate the effect of adding ECA mechanism to different network modules in YOLOv5 on the model performance, this paper introduces the ECA mechanism into the Backbone and Neck networks of YOLOv5, respectively, and compares the performance metrics of the models in the two cases. The key metrics for the comparison include Parameters, FLOPs, Size of model, FPS, Precision, Recall, and mAP@0.5. The experimental results are shown in Table 4:

**Table 4.** Comparison of performance of different locations.

	Parameters/M	FLOPs/G	Size of Model/MB	FPS/(s – 1)	Precision/%	Recall/%	mAP0.5/%
YOLOv5s	7.07	16.3	14.4	158.73	96.1	96.5	98.4
YOLOv5s-Backbone	7.22	16.3	14.7	169.49	97.1	98.3	99.1
YOLOv5s-Neck	7.37	16.8	14.9	155.56	97.5	98.9	99.3

From the results, it can be seen that the performance improvement brought by the ECA mechanism is greater than the computational overhead it brings; the increase in computation by adding the ECA mechanism in the backbone network is slightly lower than in the neck network, and adding ECA in the backbone network is more effective than in the neck network. Since the model is intended to be used in embedded devices, it is finally chosen to be added in the backbone network.

### 3.5. Performance Comparison of Various Loss Functions

To assess the influence of disparate bounding box loss functions on the accuracy of network detection, the CIoU functions adopted by EIoU, SIoU, alpha-IoU, focal-EIoU, and YOLOv5s were compared to determine the most effective loss function, and the results are shown in Table 5.

**Table 5.** Performance comparison of different loss functions.

	YOLOv5	YOLO-EIoU	YOLO-Alpha-IoU	YOLO-SIoU	YOLO-Focal-EIoU
Parameters/M	7.07	7.07	7.07	7.07	7.07
FLOPs/G	16.3	16.3	16.3	16.3	16.3
Size of model/MB	14.4	14.4	14.4	14.4	14.4
FPS/(s – 1)	158.73	166.67	172.41	169.49	166.67
Precision/%	96.1	97.3	97.9	98.5	97.5
Recall/%	96.5	97.1	98.2	97.6	98.1
mAP0.5/%	98.4	98.6	99.1	98.9	98.8

The test results show that the alpha-IoU loss function model has the best detection performance. Compared to the original model, the accuracy improves by 1.8% and the recall rate increases by 1.7%. Each loss function of mAP0.5 and FPS has increased, but the alpha-IoU loss function is more prominent. Accordingly, the incorporation of the alpha-IoU loss function can markedly enhance the precision of the network's detection capabilities.

### 3.6. Ablation Experiments

With peanut kernel quality detection as the goal, to test the performance improvement of the original YOLOv5s model via the above three improvement strategies, different numbers of improvement factors were added separately, and 10 groups of networks were combined for the ablation test.

To verify the impact of each improvement on the network model, the feasibility and effectiveness of the ShuffleNetv2 lightweight network, the ECA mechanism, and the alpha-IoU loss function were analyzed in detail, and tests were carried out on the test set. The comparison results of the mAP values for each combined model are shown in Table 6.

**Table 6.** Results of performance evaluation indices in ablation test.

ShuffleNet v2 Lightweight Network	ECA Attention Module	Alpha- IOU Loss Function	mAP- 0.5/%	Parameters/ M	FLOPs/G	FPS /(s – 1)	Precision P/%	Recall R/%	Size of Model/MB
-	-	-	98.4	7.07	16.3	158.73	96.1	96.5	14.4
√	-	-	97.6	0.45	6.1	169.49	95.4	95.2	1.4
-	√	-	99.1	7.22	16.3	166.96	97.1	98.3	14.4
-	-	√	98.9	7.07	16.3	157.77	98.1	98.2	14.4
√	√	-	98.3	0.47	6.3	188.63	96.8	95.2	1.5
√	-	√	98.6	0.45	6.1	180.33	94.8	97.1	1.5
-	√	√	99.5	7.21	16.1	175.4	99.5	98.8	14.7
√	√	√	99.4	0.47	6.3	185.67	98.8	99.4	1.3

In this study, through a comprehensive ablation test, the ShuffleNet v2, ECA mechanism, and alpha-IoU loss function were innovated to optimize the YOLOv5s model and build a SEA-YOLOv5s network to detect the appearance quality of peanut kernels under the principle of both model lightweight and recognition accuracy. The network significantly reduces the parameters and computational requirements, reducing the number of parameters to 6.7% of the original model and the computation requirements to 38.7%, while achieving a significant increase in FPS of 117% compared to the original model. In terms of performance, the accuracy and average accuracy of SEA-YOLOv5s reach 98.8% and 99.4%, respectively, which are 2.7 and 1 percentage points higher than the original YOLOv5s, effectively realizing the dual improvement of the model's lightweight quality and recognition accuracy.

### 3.7. Comparison of Different Detection Models

To further verify the improved model performance, this study conducted a comparison test between the improved model SEA-YOLOv5 and commonly used models for target detection: Faster R-CNN, SSD, YOLOv5, YOLOv7, YOLOv7-tiny, and YOLOv8. The test results are shown in Table 7.

**Table 7.** Comparison of different detection models.

	VGG 16	Faster R-CNN	SSD	YOLOv5	YOLOv7	YOLOv7- Tiny	YOLOv8	SEA- YOLOv5
Parameters/M	55.65	47.24	23.57	7.07	9.15	6.02	11.13	0.47
FLOPs/G	158.26	134.6	147.56	16.3	26	13.1	28.5	6.3
Size of model/MB	668.26	567.3	94.6	14.4	19	11.7	21.4	1.3
FPS/(s – 1)	55.26	42.53	98.76	158.73	149.25	129.87	190.76	185.67
Precision/%	84.5	98.5	96.6	96.1	96.1	95.9	98.7	98.8
Recall/%	97.04	97.6	96.9	96.5	96.2	96.3	97.3	99.4
mAP0.5/%	89.63	98.9	98.7	98.4	98.3	97.9	98.9	99.4

The improved model is superior to the other detection models in terms of detection performance. SEA-YOLOv5 has only 0.47 M parameters and 6.3 G floating-point computation. Compared with other models, SEA-YOLOv5 has obvious advantages, indicating that the strategy adopted to address the model complexity plays an effective role. Compared with Faster R-CNN, SSD, YOLOv5, YOLOv7, YOLOv7-tiny, and YOLOv8, the detection accuracy P of the model is increased by 0.3%, 2.3%, 2.7%, 2.7%, 2.9%, and 0.1%, respectively, which proves that the detection performance is significantly improved. Compared with other models, the average accuracy value is also increased by 0.5%, 0.7%, 1%, 1.1%, 1.5%, and 0.5%, and the detection speed reaches 185.67 FPS, which can meet the real-time detection requirements. The final size of the lightweight model is 1.3 MB, which is only 9.02% of the original YOLOv5s model, and the number of parameters is about 6.68% of the original model, which is more suitable for embedded devices.

### 3.8. Tests Deployed on Raspberry Pi 4B

The Raspberry Pi 4B hardware configuration is shown in Table 8.

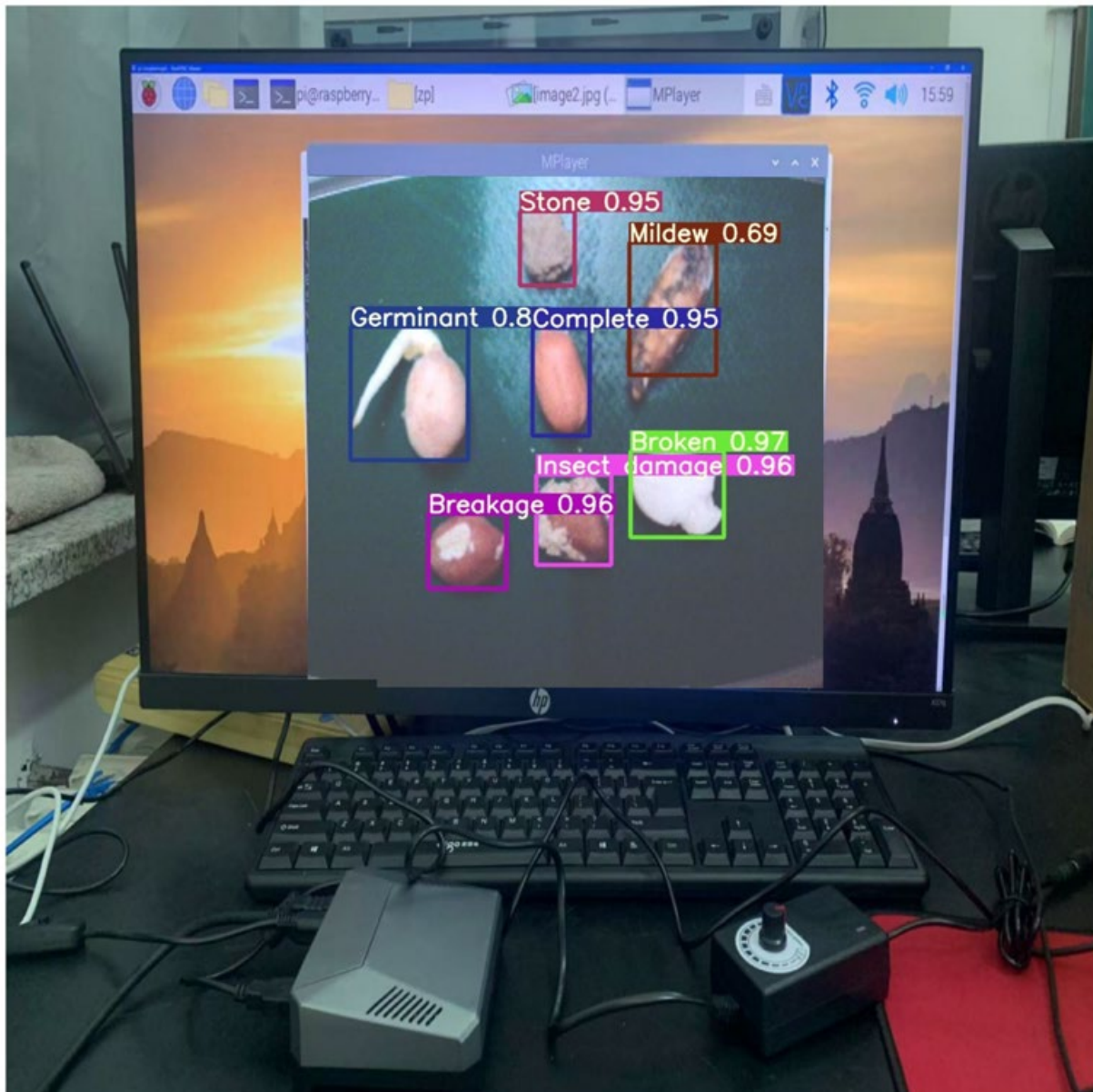
**Table 8.** Raspberry Pi 4B hardware configuration.

Names	Specifications
CPU	ARM Cortex-A72 1.5 GHz (quad-core)
GPU	Broadcom VideoCore IV, OpenGL ES 2.0, 1080p 30 h.264/MPEG-4 AVC HD decoder
Memory	64 GB microSDXC

Raspberry Pi 4B environment build: using the Raspberry Pi OS operating system, configure the model runtime environment as Python 3.8, Pytorch 1.9.0, CUDA 11.1 and CUDNN 8.05. The Raspberry Pi 4B device is shown in Figure 9. The Raspberry Pi 4B test sample is shown in Figure 10.



**Figure 9.** Raspberry Pi 4B device.



**Figure 10.** Raspberry Pi 4B test sample.

To ascertain the efficacy of the SEA-YOLOv5 model in discerning the quality of peanut kernels, the improved model was put into the Raspberry Pi 4B in this experiment, and the actual test was carried out in the self-constructed small-scale simulation test bed. It was observed that the Raspberry Pi 4B possesses limited computational resources. It is essential to consider the computational load of the model, which is 6.3 GFLOPs for the proposed algorithm, a figure that is 10.0 lower than that of YOLOv5s. In addition, the number of parameters and the size of the model are greatly reduced from the original model, which makes the computation reduced and more suitable for embedded devices; the average number of recognized FPS is also better than other YOLO series models, and the average detection time per image is only 11.2 ms. Figure 11 shows the results of the actual detection in Raspberry Pi 4B, which shows the excellent performance of the improved peanut kernel detection model.

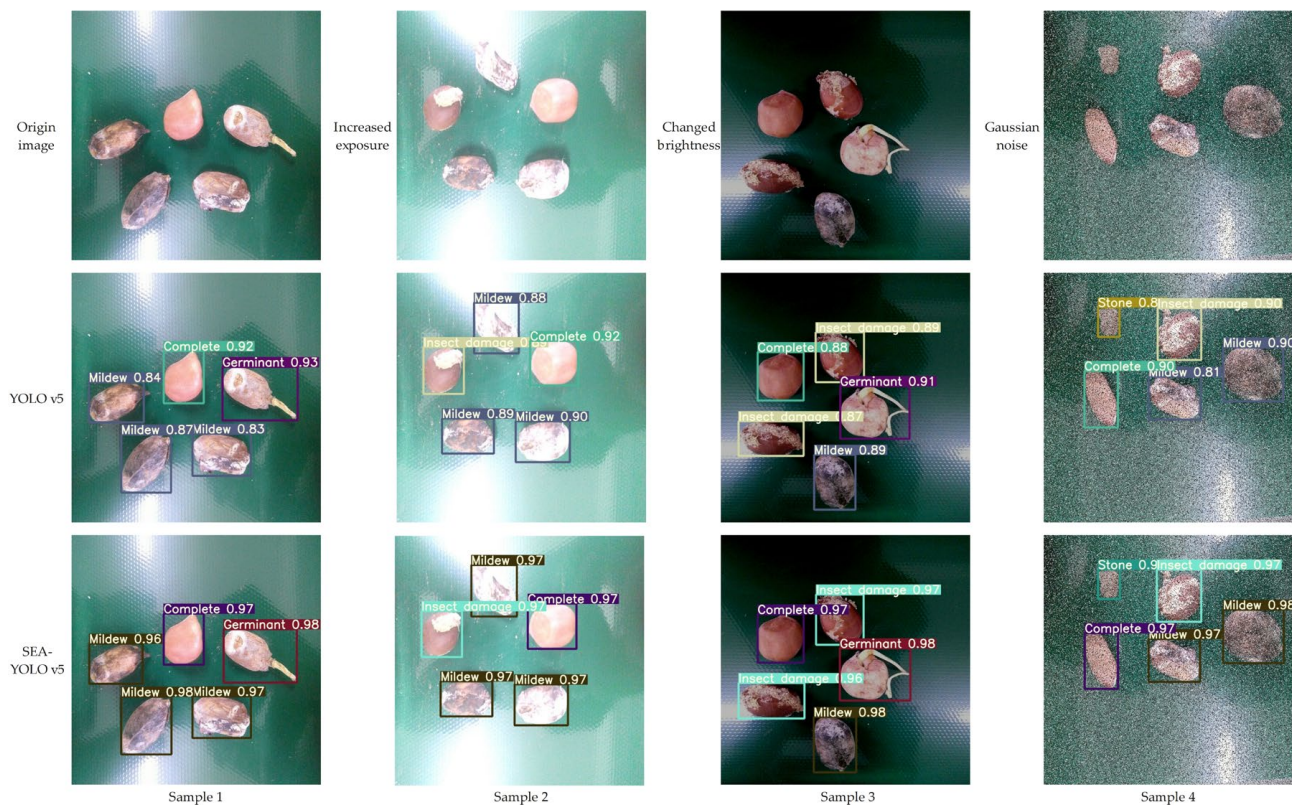


Figure 11. Comparison of the detection results of the model.

#### 4. Discussion

In the comprehensive examination of a peanut kernel appearance quality-detection model, the SEA-YOLOv5 model was developed, and its high efficiency and accuracy were validated. This successful demonstration of the model’s potential for wide application and excellent performance in this field provides compelling evidence of its value. This model integrates cutting-edge data processing and analysis technology and considers the core factors that may affect the appearance quality of peanut kernels. These factors include soil, stone, and other impurities introduced by field harvesting; breakage and damage caused by husking; insect erosion; mold; and germination caused by long-term storage. The experimental results demonstrate that the model is capable of accurately identifying and quantifying the influence of these complex factors on the appearance quality of peanut kernels, thereby providing substantial support for the realization of efficient and accurate quality assessment.

Despite the implementation of data augmentation techniques to mitigate the risk of overfitting, the model may still be susceptible to this issue if the training dataset lacks sufficient diversity or is of a limited size. To address this issue, it is essential to continuously collect a more diverse range of peanut kernel samples and conduct rigorous cross-validation and testing, to assess the model’s generalization capacity. Furthermore, in practical applications, peanut kernels may be presented in complex backgrounds, such as mixtures with other impurities, backgrounds with colors similar to peanut kernels, and so forth. These complex backgrounds may result in the model erroneously identifying background objects as peanut kernels or failing to detect the actual peanut kernels. To reduce the false detection rate, the background suppression capability of the model can be enhanced and more sophisticated post-processing algorithms can be employed to filter out the false detection results.

The current model is capable of detecting the presence of pests and diseases in peanut kernels. However, it lacks the capacity to assess the extent of infestation. To address this limitation, a larger dataset comprising peanut kernels exhibiting varying degrees



of pest and disease infestation will be collected in the future. This will facilitate the refinement of the algorithm to grade the severity of peanut kernel infestation. Furthermore, more rigorous threshold settings and post-processing algorithms will be employed to minimize the occurrence of false alarms. Additionally, data cleaning and pre-processing techniques will be utilized to mitigate the impact of noise on the detection outcomes. Concurrently, we will reinforce our collaboration with the agricultural sector, facilitate the expeditious transition of research outcomes, and endeavor to deploy this sophisticated quality assessment technology in the production of peanut kernels and other agricultural commodities with minimal delay. This will facilitate the enhancement of agricultural product quality and the assurance of food safety.

## 5. Conclusions

This study presents an enhanced method for peanut kernel appearance quality detection based on the YOLO v5 algorithm. The model was trained using images of the Qinghua 308 variety peanut kernels, with the original dataset expanded to 10,000 images through five data augmentation techniques. After comparing different versions of YOLO v5, YOLO v5s was chosen as the base model, showing an initial accuracy of 96.1%. The model was then optimized through three key improvements: incorporating ShuffleNet v2 as the backbone network, which reduced the model volume by 90% and floating-point operations by 63%; integrating the ECA mechanism in the C3 module, which improved accuracy to 98.3%; and implementing the alpha-IoU loss function, which further enhanced detection accuracy by 1.8%. Ablation tests confirmed the effectiveness of these improvements, with the optimized model achieving 98.8% precision, while requiring only 0.47 MB in parameters. The enhanced SEA-YOLOv5 model demonstrated superior performance metrics compared to alternative deep learning algorithms, while maintaining a compact size of 1.4 MB, making it suitable for embedded device deployment.

While our improved method shares commonalities with existing studies, it makes unique contributions to peanut kernel appearance quality detection. Our approach comprehensively addresses the challenges encountered during detection, particularly in handling similar targets and small objects. The model's enhanced ability to accurately classify similar peanut kernels demonstrates improved generalization capabilities. This research not only provides an innovative technical solution for peanut kernel appearance quality detection, but also establishes a valuable reference for the development of specialized detection hardware. The combination of high accuracy, compact size, and robust performance makes a significant contribution to the practical application in the field of agriculture.

**Author Contributions:** Conceptualization, C.W.; formal analysis, J.W. and D.Y.; investigation, H.Z. and G.S.; data curation, J.W.; funding acquisition, Z.L.; methodology, C.W.; writing—original draft, C.W. and Z.L.; writing—review, C.W. and X.Z.; visualization, C.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** Liaoning Provincial Department of Education Project (LJKMZ20220997); Liaoning Provincial Department of Education Project (JYTMS20231304); National Key R&D Program “Intelligent Agricultural Machinery Equipment” Key Special Project (2016YFD0702102).

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The datasets used and/or analyzed during the current study are available from the corresponding authors upon reasonable request.

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

## References

1. Li, S.; Cao, Y. Dynamics and structural characteristics of world peanut and peanut oil production and trade. *World Agric.* **2018**, *11*, 113–119+253. [[CrossRef](#)]
2. Wang, Y.; Li, S.; Yin, H.; Liu, D.; Zheng, J.; Wang, H.; Wang, J.; Cao, C.; Jia, B.; Liu, J. Effects of different storage conditions on quality of peanut. *J. Food Saf. Qual. Insp.* **2021**, *12*, 4544–4551. [[CrossRef](#)]

3. Qi, Y.W. Research on Peanut Export in Shandong Province. Master's Thesis, Shandong University of Technology, Zibo, China, 2017.
4. Li, G.Q. Research on the Development of Export Trade of Peanut Products in Shandong. Master's Thesis, Shandong Agricultural University, Tai'an, China, 2010.
5. Patel, K.K.; Kar, A.; Jha, S.N.; Khan, M.A. Machine vision system: A tool for quality inspection of food and agricultural products. *J. Food Sci. Technol.* **2012**, *49*, 123–141. [[CrossRef](#)]
6. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 142–158. [[CrossRef](#)] [[PubMed](#)]
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards realtime object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)]
8. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask RCNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969. [[CrossRef](#)]
9. 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 European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37. [[CrossRef](#)]
10. 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, 26 June–1 July 2016; pp. 779–788. [[CrossRef](#)]
11. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake, UT, USA, 18–23 June 2018; pp. 1125–1131. [[CrossRef](#)]
12. Bochkovskiy, A.; Wang, C.; Hong, Y. Yolov4: Optimal speed and accuracy of object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020. [[CrossRef](#)]
13. Ultralytics. Yolov5[EB/OL]. 26 June 2020. Available online: <https://github.com/ultralytics/yolov5> (accessed on 25 December 2023).
14. Wang, C.W.; 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. [[CrossRef](#)]
15. Terven, J.; Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. [[CrossRef](#)]
16. Gai, R.; Chen, N.; Yuan, H. A detection algorithm for cherry fruits based on the improved YOLO-v4 model. *Neural Comput. Appl.* **2021**, *35*, 13895–13906. [[CrossRef](#)]
17. Rai, N.; Zhang, Y.; Villamil, M.; Howatt, K.; Ostlie, M.; Sun, X. Agricultural weed identification in images and videos by integrating optimized deep learning architecture on an edge computing technology. *Comput. Electron. Agric.* **2024**, *216*, 108442. [[CrossRef](#)]
18. Wang, Z.; Ling, Y.; Wang, X.; Meng, D.; Nie, L.; An, G.; Wang, X. An improved Faster R-CNN model for multi-object tomato maturity detection in complex scenarios. *Ecol. Inform.* **2022**, *72*, 101886. [[CrossRef](#)]
19. Shi, L.; Wei, Z.; You, H.; Wang, J.; Bai, Z.; Yu, H.; Ji, R.; Bi, C. OMC-YOLO: A Lightweight Grading Detection Method for Oyster Mushrooms. *Horticulturae* **2024**, *10*, 742. [[CrossRef](#)]
20. Bhupendra; Moses, K.; Miglani, A.; Kankar, P.K. Deep CNN-based damage classification of milled rice grains using a high-magnification image dataset. *Comput. Electron. Agric.* **2022**, *195*, 106811. [[CrossRef](#)]
21. Long, Y.; Yang, Z.; He, M. Apple target detection method based on improved YOLOv7 in fruity period. *Trans. Chin. Soc. Agric. Eng.* **2019**, *39*, 191–199. [[CrossRef](#)]
22. Xiong, J.; Han, Y.; Wang, X.; Li, Z.; Chen, H.; Huang, Q. Detection method of papaya ripeness in natural environment based on YOLO v5-Lite. *Trans. Chin. Soc. Agric. Mach.* **2019**, *54*, 243–252. [[CrossRef](#)]
23. Yang, L.; Wang, C.; Yu, J.; Xu, N.; Wang, D. Method of Peanut Pod Quality Detection Based on Improved ResNet. *Agriculture* **2023**, *13*, 1352. [[CrossRef](#)]
24. Wu, Q.; Xu, L.; Zou, Z.; Wang, J.; Zeng, Q.; Wang, Q.; Zhen, J.; Wang, Y.; Zhao, Y.; Zhou, M. Rapid nondestructive detection of peanut varieties and peanut mildew based on hyperspectral imaging and stacked machine learning models. *Front. Plant Sci.* **2022**, *13*, 1047479. [[CrossRef](#)] [[PubMed](#)]
25. Wang, Y.; Ding, Z.; Song, J.; Ge, Z.; Deng, Z.; Liu, Z.; Wang, J.; Bian, L.; Yang, C. Peanut Defect Identification Based on Multispectral Image and Deep Learning. *Agronomy* **2023**, *13*, 1158. [[CrossRef](#)]
26. Yang, H.; Ni, J.; Gao, J.; Han, Z.; Luan, T. A novel method for peanut variety identification and classification by Improved VGG16. *Sci. Rep.* **2021**, *11*, 15756. [[CrossRef](#)] [[PubMed](#)]
27. Zhang, S.; Zhang, Q.; Li, K. Detection of peanut kernel quality based on machine vision and adaptive convolution neural network. *Trans. Chin. Soc. Agric. Eng.* **2020**, *36*, 269–277. [[CrossRef](#)]
28. 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](#)]
29. 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. [[CrossRef](#)]

30. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856. [[CrossRef](#)]
31. Sandler, M.; Howard, A.; Zhu, M.L.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [[CrossRef](#)]
32. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861. [[CrossRef](#)]
33. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807. [[CrossRef](#)]
34. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539. [[CrossRef](#)]
35. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks proceedings. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141. [[CrossRef](#)]
36. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666. [[CrossRef](#)]
37. Yan, B.; Zhang, X.; Wang, D.; Lu, H.; Yang, X. Alpha-refine: Boosting tracking performance by precise bounding box estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 20–25 June 2021; pp. 5289–5298. [[CrossRef](#)]
38. Pei, Y.; Luo, H.; Zhang, S.; Li, J.; Xu, J. High-speed rail fastener detection Algorithm based on improved Faster R-CNN. *J. East China Jiaotong Univ.* **2023**, *40*, 75–81. [[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.