



Article

An Improved Lightweight Network for Real-Time Detection of Apple Leaf Diseases in Natural Scenes

Sha Liu ^{1,2,3,†}, Yongliang Qiao ^{4,†}, Jiawei Li ^{1,2,3}, Haotian Zhang ^{1,2,3}, Mingke Zhang ¹ and Meili Wang ^{1,2,3,*}

¹ College of Information Engineering, Northwest A&F University, Xianyang 712000, China

² Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture, Xianyang 712000, China

³ Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Service, Xianyang 712000, China

⁴ Australian Centre for Field Robotics (ACFR), Faculty of Engineering, The University of Sydney, Sydney, NSW 2006, Australia

* Correspondence: wml@nwsuaf.edu.cn

† These authors contributed equally to this work.

Abstract: Achieving rapid and accurate detection of apple leaf diseases in the natural environment is essential for the growth of apple plants and the development of the apple industry. In recent years, deep learning has been widely studied and applied to apple leaf disease detection. However, existing networks have too many parameters to be easily deployed or lack research on leaf diseases in complex backgrounds to effectively use in real agricultural environments. This study proposes a novel deep learning network, YOLOX-ASSANano, which is an improved lightweight real-time model for apple leaf disease detection based on YOLOX-Nano. We improved the YOLOX-Nano backbone using a designed asymmetric ShuffleBlock, a CSP-SA module, and blueprint-separable convolution (BSConv), which significantly enhance feature-extraction capability and boost detection performance. In addition, we construct a multi-scene apple leaf disease dataset (MSALDD) for experiments. The experimental results show that the YOLOX-ASSANano model with only 0.83 MB parameters achieves 91.08% mAP on MSALDD and 58.85% mAP on the public dataset PlantDoc with a speed of 122 FPS. This study indicates that the YOLOX-ASSANano provides a feasible solution for the real-time diagnosis of apple leaf diseases in natural scenes, and could be helpful for the detection of other plant diseases.

Keywords: smart agriculture; deep learning; disease detection; attention mechanism; lightweight



Citation: Liu, S.; Qiao, Y.; Li, J.; Zhang, H.; Zhang, M.; Wang, M. An Improved Lightweight Network for Real-Time Detection of Apple Leaf Diseases in Natural Scenes. *Agronomy* **2022**, *12*, 2363. <https://doi.org/10.3390/agronomy12102363>

Academic Editor: Andrea Sciarretta

Received: 31 July 2022

Accepted: 16 September 2022

Published: 30 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 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

Apples are one of the most productive fruits in the world due to their rich nutritional value. At present, the cultivation area and output of apples in China are leading in the world, and it is one of the most important economic crops and fruits in China [1]. The development of the apple industry not only attends to human nutritional needs, but also brings great benefits to local economies. However, due to environmental and germ factors, various diseases can appear on apple leaves, which seriously hinder apple growth and the development of the apple industry, eventually causing economic losses [2].

Traditionally, apple leaf disease identification mostly relies on farmers' experience, which can lead to misjudgment and blind drug application due to complex disease symptoms, not only failing to prevent and control but also causing environmental pollution [3]. The rapid development of artificial intelligence has provided new ideas for apple leaf disease detection. Researchers began to apply machine learning algorithms to plant disease diagnosis, such as support vector machines (SVM) and K-means clustering. However, these shallow-layer models are still relatively inefficient due to the complexity of image pre-processing and feature extraction. In recent years, smart sensor and deep-learning-based technologies have been widely used in modern agriculture [4]. Through feature learning,

convolutional neural networks (CNNs) can extract features automatically and make full use of the visual features for plant disease detection. Baranwal et al. [5] proposed a method for apple leaf disease detection with an accuracy of more than 90%, but the method has limitations as it requires the leaves to be placed in simple backgrounds. SARDOGAN et al. [6] constructed Incept-Faster R_CNN network by introducing Inception v2 structure in Faster R_CNN, and the detection accuracy of two kinds of apple leaf diseases is 84.50%. Although CNNs have demonstrated satisfactory results in apple leaf disease detection, much of the research focuses only on disease detection in experimental scenarios, which cannot effectively detect apple leaf diseases in natural scenes.

Bansal et al. [7] proposed a model which is an ensemble of pre-trained DenseNet121, EfficientNetB7, and EfficientNet NoisyStudent [8] to achieve classification of apple leaves, and the proposed model can identify leaves with multiple diseases with 90% accuracy. Mukherjee et al. [9] proposed a GoogLeNet-based method for plant disease identification, and transfer learning has been used to fine tune the pre-trained model. An accuracy of 85.04% has been achieved in the identification of three disease classes in apple plant leaves. Adeel et al. [10] proposed an automatic system for segmentation and recognition of grape leaf diseases, which can reach an average segmentation accuracy rate of 90% and classification accuracy above 92%. More recently, Fu et al. [11] provided a method for detecting kiwis by adding both 3×3 and 1×1 convolutions to YOLOv3-tiny for network optimization, and the final model has a average precision of 90.05% and data weight of 27 MB. These methods based on large convolutional neural networks are relatively accurate, but not sufficiently useful for practical disease control applications due to the large number of parameters and computational effort involved.

In addition, natural scenes with strong-light, shading, and low-light conditions may affect the detection accuracy. Multiple small and complex spots may appear on the same leaf, which often has various shapes and colors, increasing the difficulty of detection and identification. In order to achieve the practicality of detection models, it is important to ensure that the models are efficient and relatively lightweight. Additionally, for achieving lightweight models, there are already many efforts in the research domain of model compression and compact network design. The commonly used techniques include replacing a large portion of 3×3 filters with smaller 1×1 filters [12]; using depthwise-separable convolution to reduce operations [13]; utilizing feature reuse and channel shuffling to achieve increased efficiency [14]. In addition, more and more attention mechanisms are being applied to filter out the most critical information for the current task from a large number of messages. There are mainly two types of attention mechanisms most commonly used in computer vision: channel attention and spatial attention. Both of these strengthen the original features by aggregating the same feature from all the positions with different aggregation strategies, transformations, and strengthening functions [15–18]. Based on these observations, some studies utilizing GCNet [19], convolutional block attention module (CBAM) [20], and shuffle attention (SA) [21] have integrated both spatial attention and channel attention into one module; these studies achieved significant improvement. SA is also an extremely lightweight plug-and-play block that can offer significant advantages in practical use.

Based on the above research, this study proposes a deep learning method based on YOLOX-Nano [22] to implement the disease detection of apple leaves, which offers improvements to address some of the problems in the detection process. By applying feature reuse in the bottleneck block of YOLOX-Nano, an asymmetric ShuffleBlock approach is designed to achieve a better trade-off between model performance and lightweight nature. The SA attention is introduced into the cross stage partial (CSP) [23] module to allow the network to focus on essential features while suppressing unnecessary ones. Additionally, BSConv [24] is used to replace the depthwise-separable convolution (DSC) [25] in YOLOX-Nano, bringing performance gains while significantly improving the efficiency of the network, and CIOU [26] loss is used for faster convergence and more accurate localization of diseased leaves. Finally, a lightweight network YOLOX-ASSANano is proposed in this

study to achieve real-time accurate detection of diseased apple leaves in natural scenes. Additionally, experiments on images captured in natural scenes with complex backgrounds demonstrated the proposed approach has more practicality in real orchard farming. The main contributions and innovations of this work are summarized as follows:

- A multi-scene apple leaf disease dataset (MSALDD) is established. The MSALDD can meet the needs of apple disease detection, which remains a challenge in complex backgrounds and under different capture conditions.
- The proposed method improves the feature learning capability of the network by using the designed asymmetric ShuffleBlock, reducing the interference of other factors in disease feature extraction using the proposed CSP-SA module. Meanwhile, BSConv and CIoU loss are used to further improve the performance.
- Experimental results show that the proposed YOLOX-ASSANano achieved a mAP of 91.08% on MSALDD with 0.83 MB parameters. The proposed model achieves a good trade-off between accuracy and parameters to efficiently detect three common diseases of apple leaves in natural scenes.

The rest of this paper is organized as follows. Section 2 describes the details of the dataset and the improvements based on the YOLOX-Nano network. Section 3 introduces the implementation details and the experimental validation of the proposed method. Finally, Section 4 presents conclusions and further work.

2. Materials and Methods

2.1. Dataset

2.1.1. Data Acquisition

Three common apple leaf diseases were selected for this study: leaf rust, leaf scab, and leaf blotch. These diseases were chosen for their frequent occurrence and the damage they cause to apple trees, which brings huge losses to the apple industry. We collected images of diseased apple leaves in natural scenarios from Baishui Apple Orchard in Weinan, Shaanxi Province, China. Apple 11 and OPPO Find X3 mobile phones were used to capture apple leaf images with resolutions of 4032×3024 and 4096×3072 pixels, respectively. The phones were held 25–50 cm away from the canopy and the image-acquisition time was from 8:00 a.m. to 19:00 p.m., which allowed the dataset to contain images of different light intensities and leaf sizes, ensuring the diversity of data. By collating the captured images and public datasets [27], a total of 3209 images of multi-scene apple leaf disease were obtained; this dataset was named MSALDD, and 80% of the MSALDD was used for training and the other 20% was used for testing to perform the experiment. Table 1 shows the distribution of labels for each category in the training and testing sets. A total of 6268 diseased apple leaves were labelled, including 2415 leaves with rust disease, 1864 leaves with scab disease, and 1989 leaves with blotch disease.

Table 1. Label distribution of MSALDD.

	Leaf Rust	Leaf Scab	Leaf Blotch	Sum
Train	1751	1412	1549	4712
Test	664	452	440	1556

Figure 1 illustrates the example images of three apple leaf diseases. The first row shows the images with simple backgrounds, while the images in the second row have multiple leaves and relatively complex backgrounds. Additionally, it can be visually seen that the three disease spots on apple leaves are similar and diverse: all diseases are nearly whorled and unevenly distributed, but differ in color and texture characteristics. Leaf rust appears as shiny orange-red spots in the early stages and then expands to form rounded orange-yellow spots with red margins. Leaf scab starts on the front side of the leaf, the spots first as yellowish circles, then gradually turns brown and finally black with a layer of black mold. Leaf blotch starts as reddish-brown round or surrounded spots with clear

margins, later turning grey with small black spots scattered in the center, and scorching of diseased leaves occurs in severe cases. The differences among these disease spots contribute to the detection and recognition of various apple disease leaves. Additionally, the collected dataset has the following three characteristics: (1) Some of the images in the MSALDD dataset have relatively simple backgrounds, and most of the images contain complex backgrounds that can interfere with diseased leaf detection, which ensures the practicality of the model. (2) Healthy leaves and different species of diseased leaves may appear in the same image at the same time. (3) All images in the dataset were manually annotated with expert guidance.

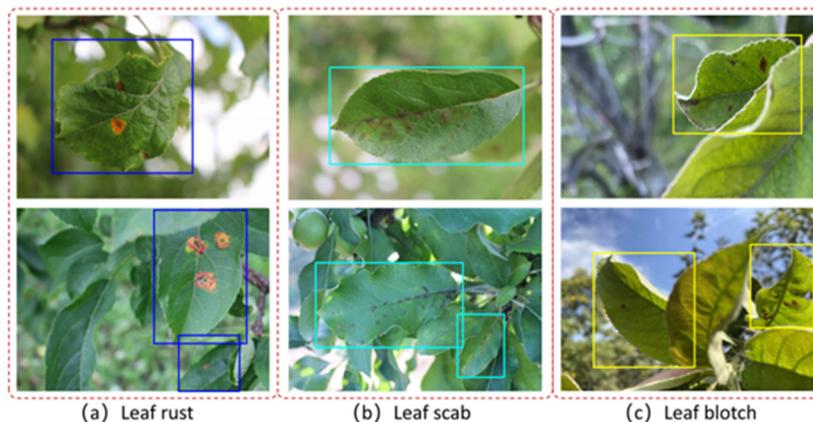


Figure 1. Examples of three different apple leaf diseases.

2.1.2. Data Processing

Image annotation is a crucial step in building the dataset and is used to mark out the location and category of diseased leaves. Using the “LabelImg” annotation tool [28], areas of diseased leaves in the image can be accurately marked. In deep learning, the effect of class imbalance and small number of samples on recognition performance is detrimental [29]. In this study, online data-enhancement methods were used to compensate for the insufficient number of samples. The specific data-enhancement methods included affine transformations, HSV domain filtering, and mosaic [30], which can reduce the effect of different locations and illumination levels on the detection results, and thus improve the model’s detection capability in complex backgrounds.

2.2. Methods of Apple Diseased Leaf Detection

2.2.1. Network Architecture

YOLO is a high-precision target-detection method, which can directly predict the location and category of the target for the entire image based on a single convolutional network. Recently, Ge and Liu et al. [22] presented some experienced updates to YOLO series, which forms a high-performance anchor-free detector called YOLOX. Additionally, for mobile devices, they adopt depthwise convolution to construct the YOLOX-Nano, which is a lightweight and efficient network. Figure 2 describes the overall learning framework for apple diseased leaf detection, where we can see that YOLOX-Nano consists of three main parts: backbone, neck, and head.

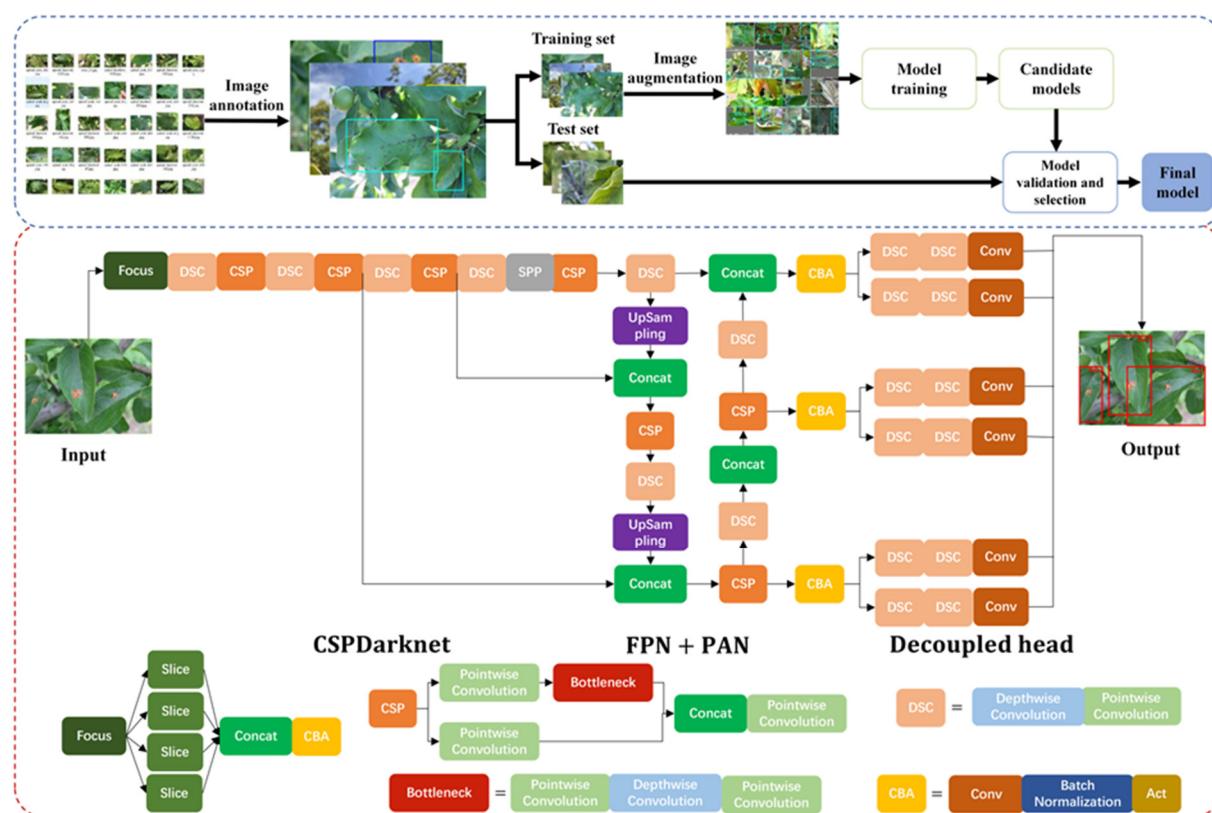


Figure 2. The overall framework for diseased apple leaf detection.

The backbone network for extracting image features, which consists of a focus module, a spatial pyramid pooling (SPP) block [31], a CSP module, and a DSC block. The focus module performs a slicing operation on the image and achieves down-sampling without information loss. The CSPNet [23] is used to solve the problem of excessive inference cost of neural networks caused by the repetition of gradient information in optimization. In the SPP block of this network, local features and global features are obtained by pooling operations at different scales, which can significantly increase the receptive fields and help the network to learn object features more comprehensively. The neck consists of feature pyramid networks (FPN) [32] structure and path aggregation networks (PAN) [33] structures, which are used to achieve fusion of feature maps at different scales for prediction at multiple scales. FPN will up-sample the top-down feature maps and add them to the bottom-up adjacent layers to obtain integrated feature maps. The PAN will reinforce the bottom-up paths, making it easier for low-level information to propagate to the top layers. Decoupled head for classification and regression tasks which allow for higher performance and faster detection rates. Moreover, YOLOX-Nano uses intersection-over-union (IoU) [34] loss for bounding box regression, and SiLU as the activation function of the network. In order to make YOLOX-Nano more suitable for apple leaf disease detection in natural scenarios and balance accuracy and speed. We implement feature reuse in the bottleneck block and design an asymmetric ShuffleBlock to keep the model lightweight. Using the attention mechanisms in CSP module to make the network focus on important features for disease detection. Replace the DSC module with BSConv, and use complete intersection-over-union (CIoU) loss to further optimize the network.

2.2.2. Asymmetric Shuffleblock

In order to achieve the practicality of the model, it is necessary to improve feature extraction and ensure the efficiency of the network. Additionally, there has been a lot of research about improving model performance and ensuring lightweight. The MobileNet [25,35,36]

series and ShuffleNet [14] are remarkably successful lightweight CNN models based on depthwise-separable convolution and intelligent design blocks including MobileNet Unit and ShuffleNet Units. ShuffleNet v2 [23] removes group convolution in ShuffleNet v1 for practical efficiency and implements feature reuse to reduce computational complexity but can effectively. Yang and Shen et al. [21] argued that it is beneficial to increase capacity by using cheaper intrinsic features or even direct feature reuse; the authors rethought the functional characteristics of two pointwise convolutions in the inverted residuals and proposed a novel asymmetrical bottleneck.

Based on the previous findings, we implement feature reuse in the bottleneck block and design an asymmetric ShuffleBlock, in which the input feature maps, X , are firstly split into two parts through a channel split, $X = [X_1, X_2]$, with each part being half of the input feature channel. Between X_1 and X_2 , the former will be converted with an asymmetrical bottleneck with no change in internal dimensions, and the latter will directly go through the block and concat with the former. The structure of bottleneck block in YOLOX-Nano and asymmetric ShuffleBlock is illustrated in Figure 3. Mathematically, the asymmetrical ShuffleBlock can be expressed by

$$Y = \text{Concat}(\text{PW}(\text{DW}(\text{Concat}(2q \cdot X_1, F_{p-q}(X_1)))), X_2) \quad (1)$$

where $X_1 \in R^{h \times w \times \frac{c}{2}}$ denotes the former part input tensor of X , while h , w , and c denote the height, width, and channel dimension, respectively. $F_{p-q} \in R^{h \times w \times (p-q) \times \frac{c}{2}}$ is the output of the first pointwise convolution (PW), which enriches the information flow by feature reuse; q controls the asymmetry rate which is shown in [21] that had the best performance when taken as 1; p denotes the expansion factor, and we set p to 2 in our experiments to achieve a good trade-off between accuracy and efficiency. Additionally, a PW for learning feature correlations from different channels after a depthwise convolution (DW). Finally, the channel concatenation and shuffle operation to achieve feature fusion and information communication of the two branches. The design of reusing the input features does not impair the expressiveness of the convolution blocks, but can effectively reduce computational complexity. Additionally, the experimental results show that the asymmetric ShuffleBlock is not only efficient, but also accurate.

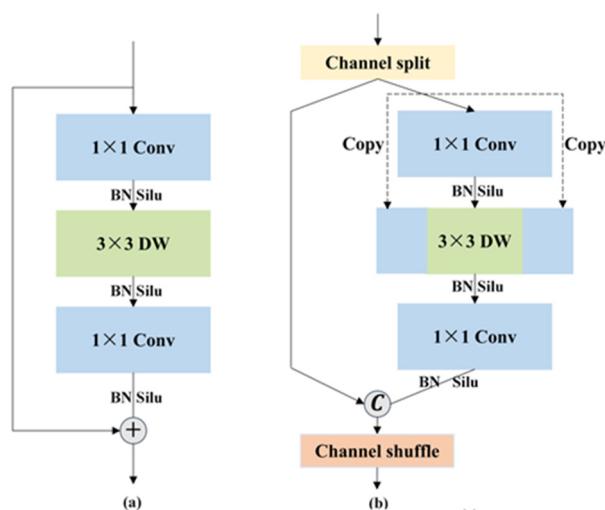


Figure 3. Structure of (a) bottleneck block and (b) asymmetric ShuffleBlock.

2.2.3. Cross Stage Partial Module with Shuffle Attention

Considering that the images are captured in natural scenes with complex backgrounds, in order to have better detection results, the attention mechanisms can be used to make the network focus on the diseased leaves rather than other factors. Squeeze and excitation (SE) [37] was used to model channel-wise relationships using two FC layers. ECA-Net [38]

adopted a 1D convolution filter to generate channel weights and significantly reduced the model complexity of SE. CBAM [20], GCNet [19], and spatial group-wise enhancement (SGE) [39] combined the spatial attention and channel attention serially. Although fusing them together may achieve better performance than their individual implementations, it will inevitably increase the computational overhead. Zhang and Yang et al. [40] proposed an efficient shuffle attention (SA) module to address this issue, which adopts shuffle units to combine two types of attention mechanisms effectively.

In this study, the CSP module was improved by introducing SA to enable the network to learn what and where to emphasize or suppress and to ensure the efficiency of the model, which we name CSP-SA. The structure of CSP-SA is shown in Figure 4. Firstly, the input feature map is obtained in two parts by using 1×1 convolution, and one branch ($S_1 \in R^{h \times w \times \frac{c}{2}}$) adopts channel attention to exploit the inter-relationships of the channels, while the other branch with bottleneck block ($S_2 \in R^{h \times w \times \frac{c}{2}}$) uses spatial attention to obtain the spatial relationships of the features. For channel attention, the global averaging pooling (GAP) is first used to generate channel-wise statistics, which can be obtained by shrinking S_1 through its spatial dimensions, $H \times W$, as shown in Equation (3). Then, a simple gating mechanism with sigmoid activation is adopted to fully capture channel-wise dependencies. Finally, the attention map is multiplied to the input feature map for adaptive feature refinement, thus enabling the neural networks to focus on important parts. In short, the final output of channel attention can be computed by

$$S'_1 = \sigma(F_c(GAP(S_1))) \cdot S_1 = \sigma(W_1 \cdot s + b_1) \cdot S_1 \quad (2)$$

$$s = GAP(S_1) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W S_1(i, j) \quad (3)$$

where $s \in R^{1 \times 1 \times \frac{c}{2}}$ denotes the channel-wise statistics, $W_1 \in R^{1 \times 1 \times \frac{c}{2}}$, $b_1 \in R^{1 \times 1 \times \frac{c}{2}}$ are parameters used to scale and shift, s , respectively, and the sigmoid function is denoted by σ .

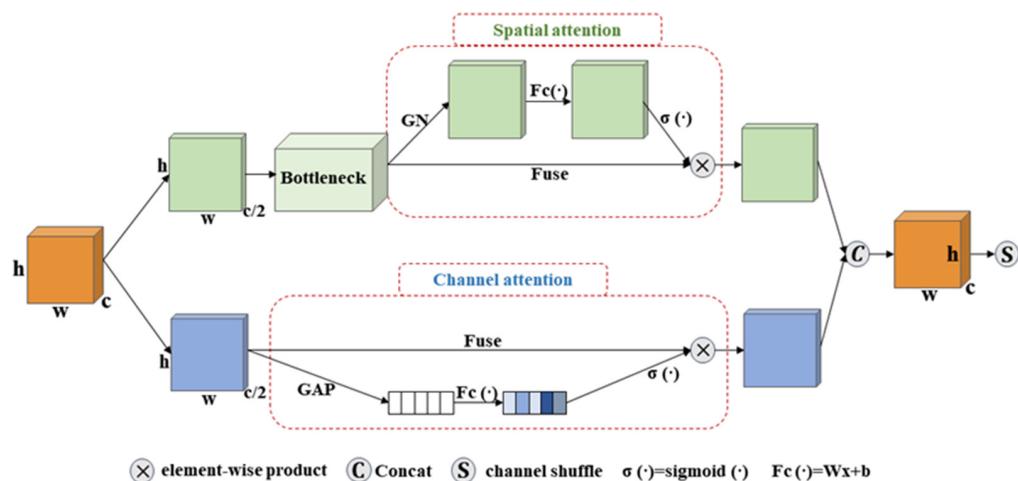


Figure 4. Structure of CSP-SA module.

Differently from channel attention focusing on ‘what’ an informative part is, spatial attention focuses on ‘where’ is meaningful, with a given input image. For spatial attention, group norm (GN) [41] is used over S_2 to obtain space-wise statistics. $F_c(\cdot)$ is adopted to enhance the representation; a compact feature is created to enable guidance for precise and adaptive selection; then, the attention map is multiplied with the input feature map to perform adaptive learning of features. The final output of spatial attention is obtained by

$$S'_2 = \sigma(F_c(GN(S_2))) \cdot S_2 = \sigma(W_2 \cdot GN(S_2) + b_2) \cdot S_2 \quad (4)$$

where refers to the sigmoid function, and W_2 and b_2 are parameters with shape $R^{1 \times 1 \times \frac{c}{2}}$. Two branches, S'_1 and S'_2 , are then concatenated and aggregated; finally, the channel shuffle operation [42] is adopted to make the information flow across branches along the channel dimension. The introduction of the SA allows the CSP module to capture the pixel-level pairwise relationship and channel dependency, which allows the network to focus on more relevant regions of diseased leaves. Meanwhile, the SA is an extremely lightweight structure, allowing considerable performance improvement while keeping the overheads small.

2.2.4. Blueprint-Separable Convolution

Lightweight models are more practical with higher training speed and less computational resources. Deeply separable convolution (DSC) [25] is used to build models in YOLOX-Nano to achieve model lightweight. The first layer of DSC is a depthwise convolution (DW) which performs lightweight filtering by applying a single convolutional filter per input channel. The second layer is a pointwise convolution (PW), which is responsible for building new features through computing linear combinations of the input channels. To further improve the model performance, we replace DSC with blueprint-separable convolution (BSConv) [24] as an efficient building block for our backbone network. The explanation of DSC and BSConv is shown in Figure 5. BSConv focuses on intra-kernel correlations while DSC in fact enforces cross-kernel correlations [24]; it is shown in [43] that the former dominates and has larger potential for an efficient separation for regular convolution. It becomes even more apparent given that natural images are inherently correlated along the depth axis, which propagates through all layers. In BSConv, the feature maps of the previous convolution can be fully utilized by the depthwise convolution through the preceding pointwise distribution. In contrast, each kernel of the first depthwise convolution of DSC can only benefit from a single feature map, leading to limited expressiveness. Therefore, we introduced BSConv as a more efficient separation of regular convolutions, and it is experimentally shown that it achieves better detection performance compared with DSC. Our backbone network structure with an input image size of 416×416 is shown in Table 2.

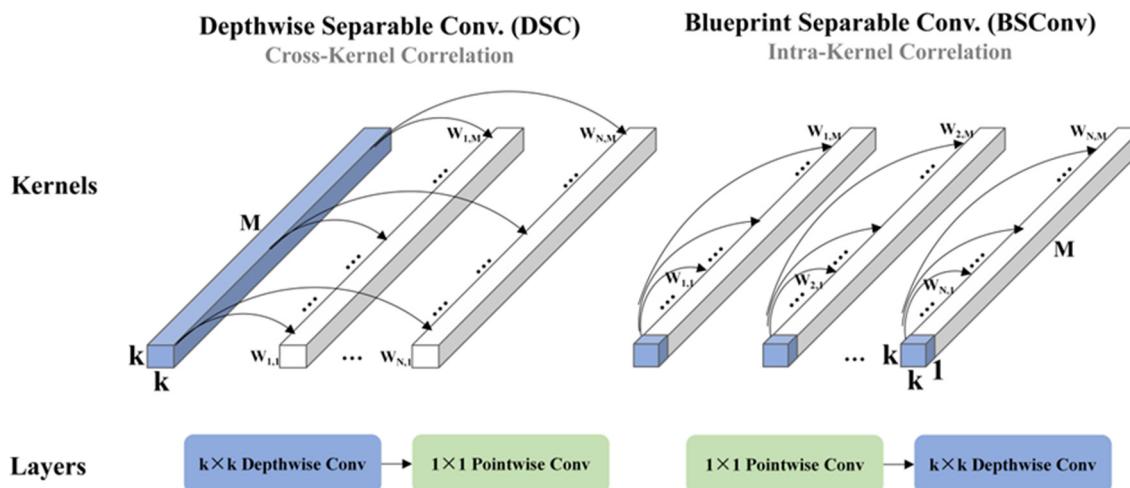


Figure 5. Interpretation for DSC and BSConv.

Table 2. The backbone network structure of YOLOX-ASSANano.

Name	Output Size	Attention	Concat
Focus	208 × 208 × 16		
DSC	104 × 104 × 32		
Conv	104 × 104 × 16		
Asymmetric ShuffleBlock	104 × 104 × 16	spatial	
Conv	104 × 104 × 16	channel	
Conv	104 × 104 × 32		(−1, −2)
DSC	52 × 52 × 64		
Conv	52 × 52 × 32		
Asymmetric ShuffleBlock	52 × 52 × 32	spatial	
Conv	52 × 52 × 32	channel	
Conv	52 × 52 × 64		(−1, −2)
DSC	26 × 26 × 128		
Conv	26 × 26 × 64		
Asymmetric ShuffleBlock	26 × 26 × 64	spatial	
Conv	26 × 26 × 64	channel	
Conv	26 × 26 × 128		(−1, −2)
DSC	13 × 13 × 256		
SPP	13 × 13 × 256		
Conv	13 × 13 × 128		
Asymmetric ShuffleBlock	13 × 13 × 128	spatial	
Conv	13 × 13 × 128	channel	
Conv	13 × 13 × 256		(−1, −2)

2.2.5. Complete Intersection-Over-Union Loss

To obtain faster convergence and more accurate localization of diseased leaves, we use CIoU [26] loss for bounding box regression. The IoU [34] loss is used in YOLOX-Nano to calculate the difference of the ground truth and prediction bounding boxes, which encodes the widths, heights, and locations of two bounding boxes into the region property, and then calculates a normalized measure that focuses on their areas [44]. This property makes IoU loss invariant to the scale of the problem under consideration, but it does not reflect whether two shapes are in the vicinity of each other or are very far from each other. For accurate and efficient localization, Zheng et al. proposed the CIoU loss, which takes into account information on the scale of the centroid, overlap, and aspect ratio of the boundary and can better describe the regression of rectangular boxes. Additionally, we achieved significant performance gains with CIoU loss. The IoU loss and CIoU loss can be calculated by Equations (5) and (6). As shown in Figure 6, IoU represents the ratio of the intersection and union of the predicted bounding box and the ground truth bounding box; d and c are the distance between the centers of the two bounding boxes and the diagonal distance of their union, respectively. w^{gt} and h^{gt} are the width and height of the ground truth bounding box, respectively, while w and h represent those of the predicted bounding box, respectively. Additionally, v measures the consistency of the aspect ratio and α is a positive trade-off parameter.

$$\text{loss}_{\text{IoU}} = 1 - \text{IoU} \quad (5)$$

$$\text{loss}_{\text{CIoU}} = 1 - \text{IoU} + \frac{d^2}{c^2} + \alpha v \quad (6)$$

$$\text{IoU} = \frac{|B_{\text{Groundtruth}} \cap B_{\text{Prediction}}|}{|B_{\text{Groundtruth}} \cup B_{\text{Prediction}}|} \quad (7)$$

$$\alpha = \frac{v}{(1 - \text{IoU}) + v} \quad v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (8)$$

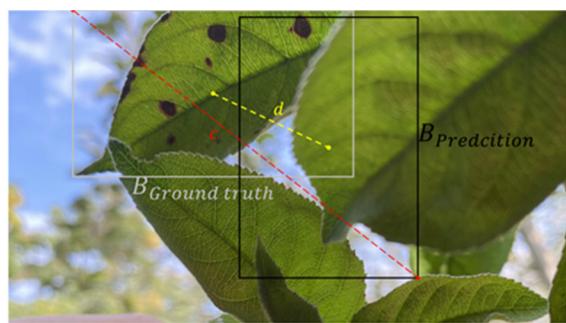


Figure 6. Illustration of CIoU loss for box regression.

2.3. Implementation Details

2.3.1. Experimental Setup

The experiments are performed on an 18.04.6-Ubuntu server with an Intel Xeon Platinum 8160T@2.1 GHz. It is accelerated by an NVIDIA TITAN RTX GPU that has 16 GB of RAM. All the deep learning models adopted in this study are implemented in the Pytorch deep learning framework. Additionally, the configuration parameters are listed in Table 3.

Table 3. Hardware and software environments.

Configuration Item	Value
CPU	Inter Xeon Plantium 8160T @2.1 GHz
GPU	NVIDIA TITAN RTX 24 G
Memory	128 G
Operating system	Ubuntu 18.04.6 LTS (64-bit)
Deep learning framework	Pytorch

2.3.2. Evaluation Metrics

In order to verify the performance of the model, indicators including the precision (P), recall (R), F₁-score, mean average precision (mAP), parameters, and frames per second (FPS) are adopted for the evaluation in this study. Parameters are used to measure the size of the model, and FPS is used to evaluate the real-time processing speed of the model. The F₁-score can be used to equalize the precision and recall. Here, the mAP is the mean value of the average precision when apple diseased leaves are detected; the higher the value, the better the detection result for diseased leaves. It is calculated based on intersection-over-union (IoU) loss; in this study, the target confidence threshold is taken as 0.25, the IoU is taken as 0.5 for test. The precision, recall, F₁-score, AP, and mAP can be calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\% \quad (9)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\% \quad (10)$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

$$\text{mAP} = \frac{\sum_{i=1}^C P(c)dR(c)}{C} \quad (12)$$

where TP, FP, and FN are the numbers of true positive cases, false positive cases and false negative cases, respectively. C is the number of types of apple leaf diseases, and C = 3 in this study.

3. Results and Discussion

3.1. Comparison Results on MSALDD

In order to verify the effectiveness of the proposed method for apple leaf diseases, five object-detection algorithms were compared in this study, including MobileNetv2-SSD, ShuffleNetv2-YOLOv5, YOLOv5s, YOLOX-Nano, and ours. The detection results on the MSALDD are shown in Table 4. The results show that the mAP of the proposed method is 91.08%, which is a 2.67% improvement compared with YOLOX-Nano, achieving the highest value among the five compared algorithms. Regarding the model size, our model is the lightest with only 0.83 MB of parameters. Additionally, the average inference time of the proposed model for a single image is about 8 ms, which achieves real-time detection. Taken together, the proposed method has the highest mAP, the lowest number of parameters and faster speed for detecting apple leaf diseases in natural environments compared with the four comparison algorithms.

Table 4. Detection results on MSALDD.

Models	Precision (%)	Recall (%)	mAP (%)	Parameters (MB)	FPS
MobileNetv2-SSD	64.47	57.38	64.04	11.78	62
ShuffleNetv2-YOLOv5	85.19	80.01	86.97	1.47	142
YOLOv5s	89.59	82.68	90.68	6.74	120
YOLOX-Nano	89.15	79.05	88.41	0.86	128
Ours	89.75	83.63	91.08	0.83	122

The average precision of the five algorithms for the detection of three apple leaf diseases of MSALDD is shown in Figure 7. The average precision of our model in detecting all three diseases in the natural environment was improved by 4.08%, 1.41%, and 2.52%, relative to YOLOX-Nano. Especially, the detection accuracies for leaf rust and leaf scab were the highest among the five compared algorithms, reaching 88.29% and 93.57%, respectively. The results showed that the proposed YOLOX-ASSANano is effective and feasible to achieve accurate and fast detection of apple diseased leaves.

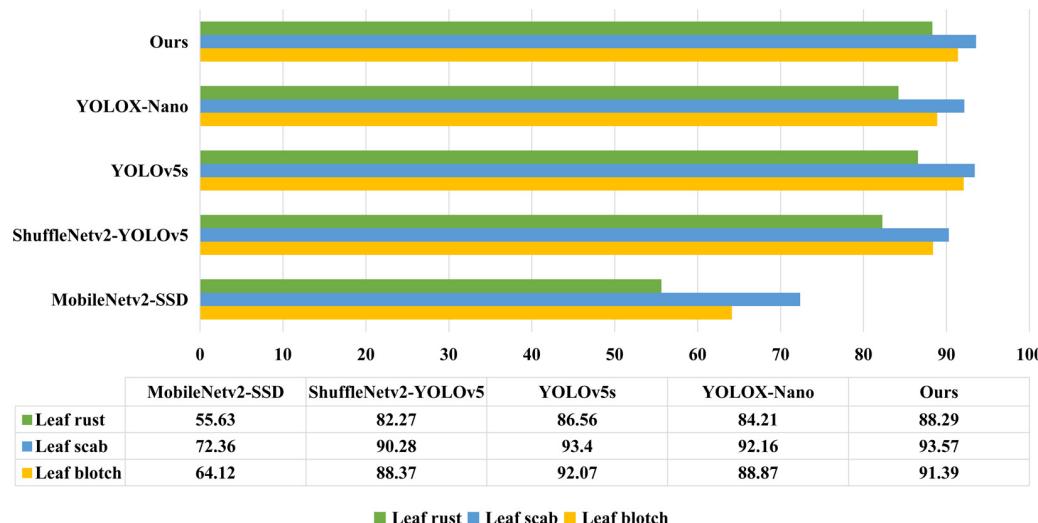


Figure 7. Average precision (AP) for the three disease categories of MSALDD.

3.2. Ablation Study on the Model's Performance

In order to verify the effectiveness of asymmetric ShuffleBlock, CAP-SA module, BSCConv, and CIoU loss function for the algorithm, ablation experiments were conducted on the MSALDD dataset under the same experimental setting as the comparison experiments.

Table 5 shows an evaluation of different expansion factor p of asymmetric ShuffleBlock on the MSALDD; it can be found that $p = 2$ demonstrates the best trade-off for disease detection in terms of accuracy and complexity. The influence of different loss functions for bounding box regression was also studied; the comparisons of the detection results are shown in Table 6. Experimentally, it was proved that using CIoU loss was most beneficial for model performance improvement. Using different loss functions for bounding box regression had little effect on the detection speed of our model, and all achieved real-time detection.

Table 5. Ablation study on expansion factor p .

	p	mAP (%)	Parameters (MB)
+Asymmetric ShuffleBlock	1	87.57	0.81
	2	89.06	0.83
	3	89.00	0.84
	4	89.04	0.86

Table 6. Ablation study on loss for regression.

	mAP (%)	FPS
IoU loss	90.47	136
CIoU loss	91.08	122
DIoU loss	90.86	130
GIoU loss	90.33	133

In Table 7, it can be seen that all the modules used in this work had a positive effect on the network. Firstly, compared with the bottleneck block used in YOLOX-Nano, the asymmetric ShuffleBlock achieved better results with fewer parameters. Additionally, according to Table 5, the asymmetric ShuffleBlock works best at $p = 2$. Secondly, we can see from the table that using the CSP-SA module can increase the mAP with only a small increase in parameters. Additionally, it clearly shows that BSConv can be used as a drop-in replacement of DSC, increasing the mAP of the model to 90.47% with essentially no increase in parameters or computational costs. Furthermore, as summarized in Table 6, the CIoU loss makes further performance improvements compared with other loss functions for bounding box regression. Combining Table 7 and the above analysis, the effectiveness of each component of the proposed algorithm is proved.

Table 7. Ablation experiments of YOLOX-ASSANano.

Asymmetric Shuffleblock	CSP-SA	BSConv	CIoU	mAP (%)	Parameters (MB)
✓				88.41	0.856
✓	✓			89.06	0.826
✓	✓	✓		89.84	0.827
✓	✓	✓	✓	90.47	0.829
✓	✓	✓	✓	91.08	0.829

A check mark indicates that the corresponding module is used.

3.3. Comparison Experiments on PlantDoc

To further verify the effectiveness and applicability of our algorithm, we conduct comparative experiments on the public plant leaf disease dataset PlantDoc [45] with the same experimental setting described above. The lack of availability of sufficiently largescale non-lab dataset remains a major challenge for enabling vision-based plant disease detection. Against this background, Singh and Jain et al. presented PlantDoc for visual plant disease detection, which contains images of plant diseases in natural scenes. Additionally, in total there are 2569 images with 30 different categories, 13 different plant species, and up to

17 categories of diseases in PlantDoc. It does not have large enough samples and most of the images are low resolution and noisy, which makes the detection accuracy relatively low. As can be seen from Table 8, the mAP of YOLOX-Nano on PlantDoc is 54.82% and the F_1 -score is 50.35%. The TL-SE-ResNeXt-101 model based on migration learning and residual networks proposed by Wang [46] has a mAP value of 47.37% when the input image size is 224×224 . Shill and Rahman [47] proposed an accurate plant disease detection model based on YOLOv4, which is named as M_YOLOv4 in this table, and the mAP and F_1 -scores of the model are 55.45% and 56.00%, respectively. The F_1 -score of YOLOX-ASSANano proposed in this work is 56.11%, and the mAP of our model is 3.41% higher than that of M_YOLOv4, reaching 58.86% as the best disease detection result, which indicates the greater usefulness of our model in real agricultural settings.

Table 8. Comparison results on PlantDoc.

Models	mAP (%)	F_1 (%)
YOLOX-Nano	54.82	50.35
TL-SE-ResNeXt-101	47.37	41.91
M_YOLOv4	55.45	56.00
Ours (YOLOX-ASSANano)	58.86	56.11

3.4. Visualization and Discussion

To further analyze the proposed detection model, Figure 8 shows the visualization of the detection results on the MSALDD dataset, and the first row shows the ground truth of the diseased leaf in each image. It can be seen that the proposed YOLOX-ASSANano outperformed other comparative algorithms in both bounding box prediction and recognition accuracy. In Figure 8A, the images in column, except from those for MobileNetv2-SSD, show that all models accurately detected diseased leaves due to the good illumination and obvious disease targets. For the diseased leaves that are partially obscured (Figure 8B—column) and images with off-angle and strong light interference (Figure 8C—column), the proposed YOLOX-ASSANano successfully detected diseased leaves, which performed better than MobileNetv2-SSD, ShuffleNetv2-YOLOv5, and YOLOv5s. Besides, when the disease spots are small (Figure 8D—column), only YOLOX-Nano and the proposed YOLOX-ASSANano are more accurate in locating diseased leaves, while others show missed detection. The images in the column of Figure 8E were taken in a dim environment; although some diseased leaves are difficult to see even by human eye, the proposed YOLOX-ASSANano could detect the diseased leaves accurately. In combination with the above analysis, the proposed YOLOX-ASSANano demonstrates good detection ability of apple disease leaves in various complex scenarios (e.g., different illuminations, small disease spots, and changing camera viewpoints).

The comparison of the detection results of the YOLOX-Nano and the proposed algorithm on the PlantDoc is shown in Figure 9. There is only one grape leaf in column A images with simple backgrounds, but YOLOX-Nano incorrectly identifies the leaf fractures of the grape leaf as strawberry leaves. For the images in column B, a late blight tomato leaf with a small target and only the reverse side is visible, which is detected by our model but not by YOLOX-Nano. The background in column C and column D is complex and the targets are not very clear, our model is able to identify the diseased leaves, while YOLOX-Nano shows false detection. Based on the above analysis, we can see that YOLOX-ASSANano performs better in detecting plant diseases in real agricultural environments.

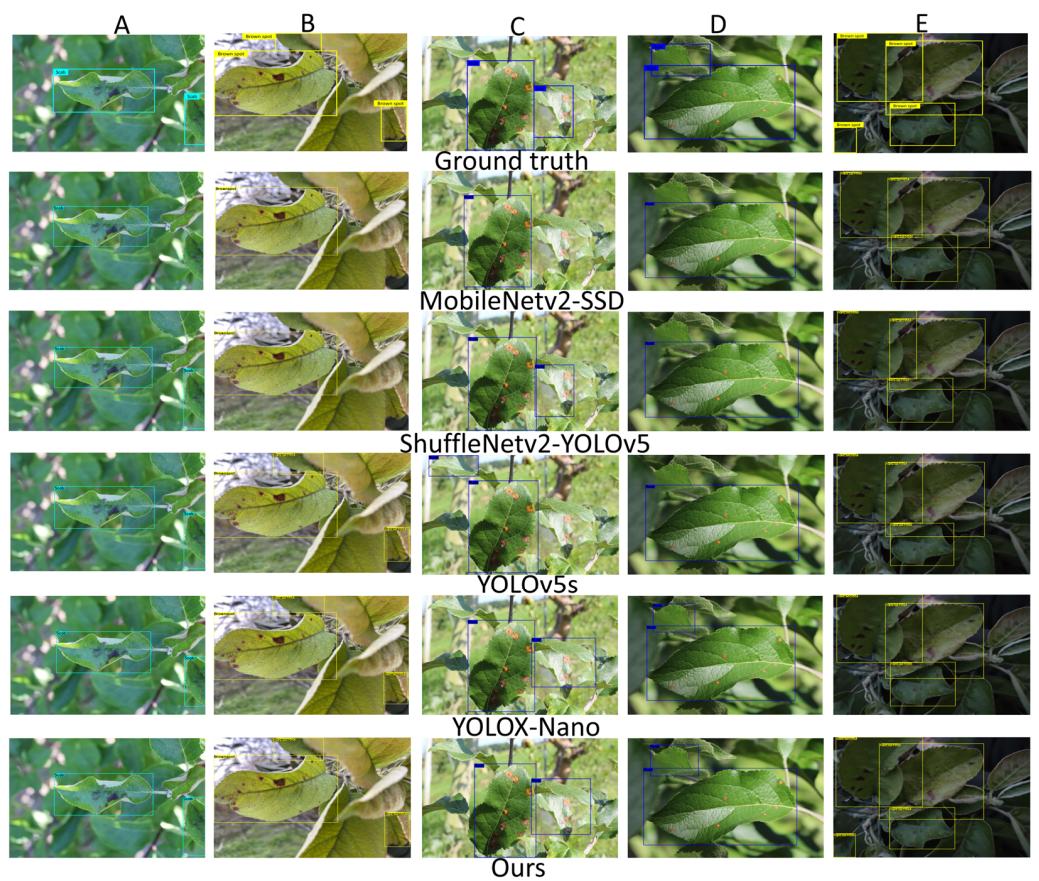


Figure 8. Visualization of detection results for different scenarios of the MSALDD. (A): Simple scenes; (B): obscured scenes; (C): Scenes in strong light; (D): Small target scenes; (E): Dim scenes.

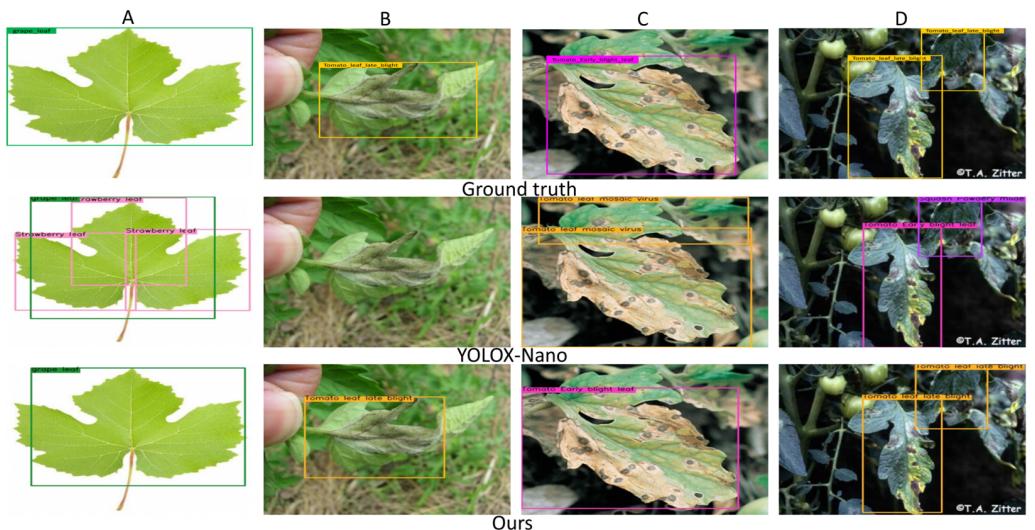


Figure 9. Visualization of detection results for different scenarios of the PlantDoc. (A): Simple scenes; (B): Small target scenes; (C): Complex scenes; (D): Multi-target scenes.

4. Conclusions

In this work, a deep-learning-based detector, YOLOX-ASSANano, is proposed for the detection of apple leaf diseases. The proposed asymmetric ShuffleBlock approach improves the feature-extraction capability of the network through feature fusion, while keeping the model lightweight. The designed CSP-SA module, which introduces attention mechanisms, effectively allows the network to focus on important features for disease

detection. Additionally, the use of BSConv and CIoU loss achieves faster convergence and better performance. In addition, we generated the MSALDD dataset by taking pictures of real apple orchards with complex backgrounds. The results show that the YOLOX-ASSANano can effectively extract the features of disease spots and detect apple leaf diseases with high accuracy and satisfactory detection speed, the mAP is 91.08% on MSALDD with 122 FPS detection speed. Moreover, compared with other methods on the public PlantDoc dataset, our algorithm achieves better detection results, which proves the effectiveness of our algorithm. Future work might include deploying the generated models to mobile devices to help farmers detect diseased leaves in apple orchards; furthermore, we intend to apply the proposed method to other plant diseases for evaluation.

Author Contributions: S.L.: investigation, data curation, methodology (lead), formal analysis, and writing—original draft preparation. Y.Q.: methodology (supporting) and writing—review and editing. M.W.: coordinating experiments in field, resources, funding acquisition, and paper revising. J.L.: data collation, methodology (supporting), and writing—review and editing. H.Z.: data collation and writing—review and editing. M.Z.: guidance on data collection and data annotation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by Xianyang Science and Technology Research Plan Project (2021ZDYF-NY-0014) and Xi'an Science and Technology Plan Project (2022JH-JSYF-0270).

Data Availability Statement: Data are available on request from the authors.

Acknowledgments: The authors appreciate the support of Baishui Apple Orchard in Shaanxi, China for our work. All supports and assistance are sincerely appreciated.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Snyder, F.; Ni, L. Chinese apples and the emerging world food trade order: Food safety, international trade, and regulatory collaboration between China and the European Union. *Chin. J. Comp. Law* **2017**, *5*, 253–307. [[CrossRef](#)]
2. Khan, A.I.; Quadri, S.; Banday, S. Deep learning for apple diseases: Classification and identification. *Int. J. Comput. Intell. Stud.* **2021**, *10*, 1–12.
3. Bhat, A.; Wani, M.; Bhat, G.; Qadir, A.; Qureshi, I.; Ganaie, S.A. Health cost and economic loss due to excessive pesticide use in apple growing region of Jammu and Kashmir. *J. Appl. Hortic.* **2020**, *22*, 220–225.
4. Abade, A.; Ferreira, P.A.; de Barros Vidal, F. Plant diseases recognition on images using convolutional neural networks: A systematic review. *Comput. Electron. Agric.* **2021**, *185*, 106125. [[CrossRef](#)]
5. Baranwal, S.; Khandelwal, S.; Arora, A. Deep learning convolutional neural network for apple leaves disease detection. In Proceedings of the International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Jaipur, India, 26–28 February 2019.
6. Sardogan, M.; Yunus, Ö.; Tuncer, A. Detection of Apple Leaf Diseases Using Faster R-CNN. *Düizce Üniversitesi Bilim Ve Teknol. Derg.* **2020**, *8*, 1110–1117.
7. Bansal, P.; Kumar, R.; Kumar, S. Disease detection in Apple leaves using deep convolutional neural network. *Agriculture* **2021**, *11*, 617. [[CrossRef](#)]
8. Xie, Q.; Luong, M.T.; Hovy, E.; Le, Q.V. Self-training with noisy student improves imagenet classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10687–10698.
9. Mukherjee, S.; Kumar, P.; Saini, R.; Roy, P.P.; Dogra, D.P.; Kim, B.G. Plant disease identification using deep neural networks. *J. Multimed. Inf. Syst.* **2017**, *4*, 233–238.
10. Adeel, A.; Khan, M.A.; Sharif, M.; Azam, F.; Shah, J.H.; Umer, T.; Wan, S. Diagnosis and recognition of grape leaf diseases: An automated system based on a novel saliency approach and canonical correlation analysis based multiple features fusion. *Sustain. Comput. Inform. Syst.* **2019**, *24*, 100349. [[CrossRef](#)]
11. Fu, L.; Feng, Y.; Wu, J.; Liu, Z.; Gao, F.; Majeed, Y.; Al-Mallahi, A.; Zhang, Q.; Li, R.; Cui, Y. Fast and accurate detection of kiwifruit in orchard using improved YOLOv3-tiny model. *Precis. Agric.* **2021**, *22*, 754–776. [[CrossRef](#)]
12. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
13. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.

14. 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.
15. Lee, H.; Kim, H.E.; Nam, H. Srm: A style-based recalibration module for convolutional neural networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1854–1862.
16. Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C.C.; Lin, D.; Jia, J. Psanet: Point-wise spatial attention network for scene parsing. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 267–283.
17. Li, X.; Zhong, Z.; Wu, J.; Yang, Y.; Lin, Z.; Liu, H. Expectation-maximization attention networks for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9167–9176.
18. Zhu, Z.; Xu, M.; Bai, S.; Huang, T.; Bai, X. Asymmetric non-local neural networks for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 593–602.
19. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
20. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
21. Yang, H.; Shen, Z.; Zhao, Y. AsymmNet: Towards ultralight convolution neural networks using asymmetrical bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2339–2348.
22. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
23. 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.
24. Haase, D.; Amthor, M. Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved MobileNets. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14600–14609.
25. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
26. Thapa, R.; Zhang, K.; Snavely, N.; Belongie, S.; Khan, A. The Plant Pathology Challenge 2020 data set to classify foliar disease of apples. *Appl. Plant Sci.* **2020**, 8, e11390. [CrossRef] [PubMed]
27. Tzutalin. Labelimg. 2016. Available online: <https://github.com/tzutalin/labelimg> (accessed on 9 June 2021).
28. Buda, M.; Maki, A.; Mazurowski, M.A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* **2018**, 106, 249–259. [CrossRef] [PubMed]
29. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.109342020.
30. Jocher, G. Yolov5. 2021. Available online: <https://github.com/ultralytics/yolov5> (accessed on 3 October 2021).
31. 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] [PubMed]
32. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
33. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
34. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An advanced object detection network. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520.
35. 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, Korea, 27 October–2 November 2019; pp. 1314–1324.
36. 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.
37. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
38. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. Supplementary material for ‘ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13–19.
39. Li, X.; Hu, X.; Yang, J. Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. *arXiv* **2019**, arXiv:1905.09646.
40. Zhang, Q.L.; Yang, Y.B. Sa-net: Shuffle attention for deep convolutional neural networks. In Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 2235–2239.

41. Wu, Y.; He, K. Group normalization. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
42. 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, 14–19 June 2020; pp. 390–391.
43. Guo, J.; Li, Y.; Lin, W.; Chen, Y.; Li, J. Network decoupling: From regular to depthwise separable convolutions. *arXiv* **2018**, arXiv:1808.05517.
44. 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.
45. Singh, D.; Jain, N.; Jain, P.; Kayal, P.; Kumawat, S.; Batra, N. PlantDoc: A dataset for visual plant disease detection. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, Hyderabad, India, 5–7 January 2020; pp. 249–253.
46. Wang, D. Crop disease classification with transfer learning and residual networks. *Trans. Chin. Soc. Agric. Eng.* **2021**, *37*, 199–207.
47. Shill, A.; Rahman, M.A. Plant Disease Detection Based on YOLOv3 and YOLOv4. In Proceedings of the 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), Rajshahi, Bangladesh, 8–9 July 2021; pp. 1–6.