1.Introduction

[Author Name(s), First M. Last, Omit Titles and Degrees]

[Institutional Affiliation(s)]

Author Note

[Include any grant/funding information and a complete correspondence address.]

Abstract

This paper proposes an enhanced YOLOv5n model, named CTR\_YOLOv5n, that incorporates a Coordinate Attention (CA) mechanism and a Swin Transformer (STR) detection head to identify common maize diseases, such as leaf spot, gray spot, and rust, in mobile applications. Maize diseases are known to occur frequently and are complicated and difficult to control, which can have a significant impact on maize yield and quality. By building upon the lightweight YOLOv5n model, the accuracy of the CTR\_YOLOv5n model is improved through the addition of a CA attention module, which enhances the model's global information acquisition capabilities by using TR2 as the detection head. The algorithm model achieves an average recognition accuracy of 95.2%, which is 2.8% higher than the original model, and the memory size is significantly reduced to 5.1MB compared to 92.9MB of YOLOv5l, which meets the requirement of being lightweight. Compared with mainstream attention mechanisms like SE, CBAM, and ECA, the CA mechanism used in this model provides better recognition results and higher accuracy, enabling fast and accurate recognition of maize leaf diseases with fewer computational resources. These findings provide new insights and methods for real-time recognition of maize and other crop spots in mobile applications.

Keywords: Object Detection; Maize-Leaf Anomaly;Deep Learning;Attention Mechanism,Digital Agriculture;

1.Introduction

Maize is a vital crop worldwide, and its leaves significantly influence the crop's yield and quality. Early detection of maize leaf diseases and insect infestations is crucial for effectively managing these pests, which can cause significant losses in crop yield. However, traditional methods of monitoring maize leaves through visual inspection by farmers can be time-consuming and subjective. Object detection technology can streamline this process and provide more efficient and accurate detection of maize leaf pests.

Maize is one of the three primary food crops and a crucial source of revenue for farmers worldwide. With its high nutritional value, maize continues to play a significant role in addressing the issue of human food supply. In China, 60% of maize is used for feed for livestock and poultry industries, 30% for industrial purposes, and 10% for direct consumption by people. Therefore, increasing maize production and maintaining high quality is essential for China's agricultural industry. However, maize pests and diseases have the most significant negative impact on its production and quality, causing varying degrees of yield reduction and quality decline, which seriously affect the economic benefits of producers and the industry as a whole.

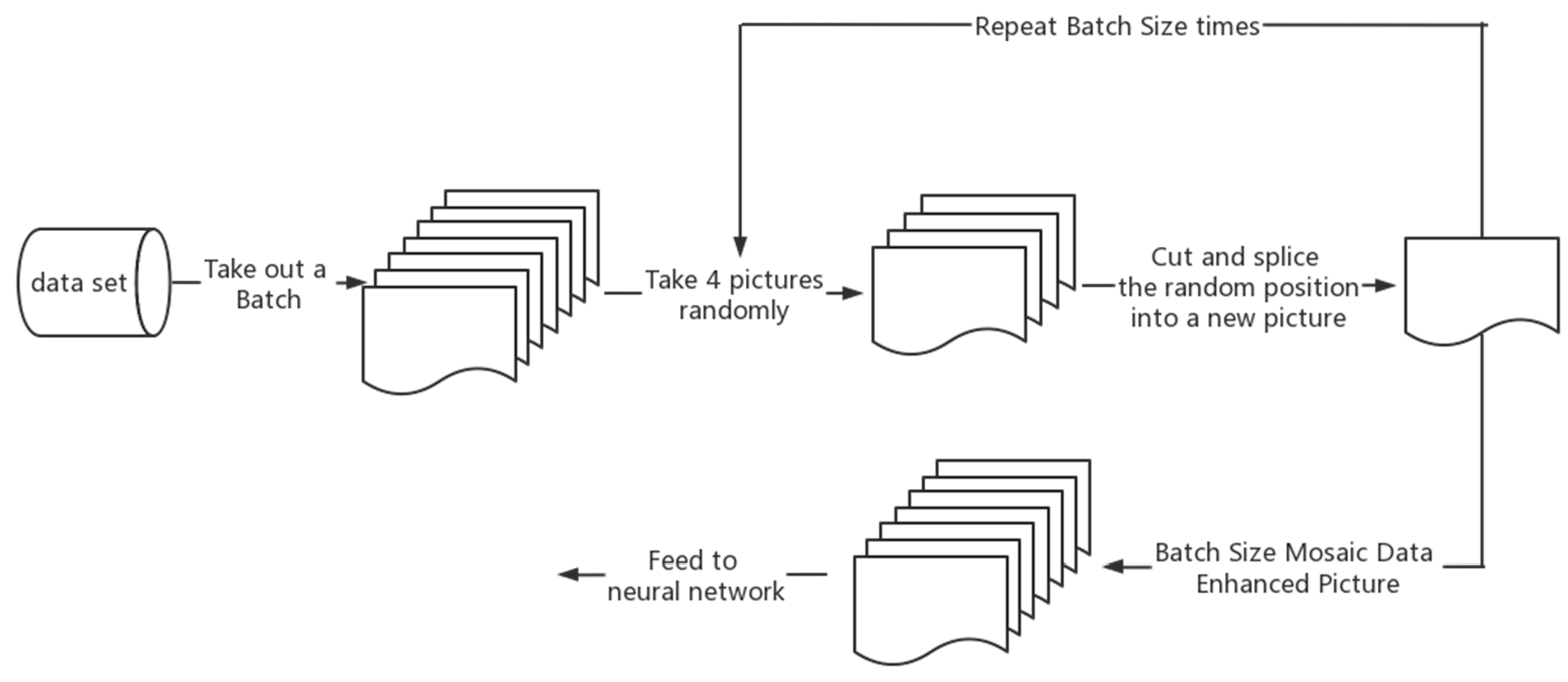
The current method for identifying maize diseases in China is based on the empirical judgment of crop pathologist experts in the field and technicians specialized in plant protection, relying on individual experience, which has limitations. The traditional disease identification method can lead to inaccuracies in the identification process due to human errors, especially when there are too many samples to test with many different disease types.

Recent advancements in big data analysis technology and GPUs have improved the computing power of computers, and deep learning techniques have been developed rapidly and used in many applications, such as agricultural pests and diseases. Various deep learning-based models have been proposed to identify maize leaf diseases with high accuracy rates.

However, there are some limitations in recognition techniques based on deep learning technology, such as less small-scale target data, larger memory consumption of the model, and being unsuitable for mobile deployment. Therefore, this paper aims to investigate the problem of maize leaf diseases and apply the current deep learning technology to design an experimental study, hoping that farmers can rely on their mobile phones in the field to identify diseases on maize in a timely and effective manner, thus alleviating the problems of reduced yields and reduced quality of maize. The focus of this study is on developing a lightweight and accurate model for real-time disease identification using mobile devices. The YOLOv5n is one of the commonly used target detection networks, and it belongs to a one-stage target detection algorithm with a simple structure, small computation, and fast operation speed, making it suitable for crop disease identification research.

# 2.Materials

In the YOLOv5n network model described in this paper, not only are some basic data enhancement methods included, but also the Mosaic data enhancement [[18](#_bookmark32)] is used, whose main idea is to select four images from the used dataset, crop and scale them randomly, and then arrange them randomly to form a new image. This has the advantage of increasing the number of datasets while augmenting the number of small sample targets, and it improves the training speed of the model. The flowchart of Mosaic data enhancement is shown in Figure [2](#_bookmark2).

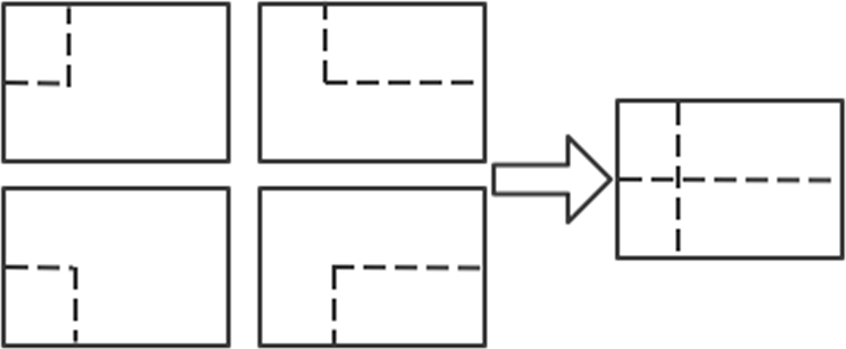


**Figure 2.** Mosaic data enhancement flow chart.

Mosaic data enhancement utilizes four images, which enriches the background of the detected objects and calculates the data of four images at once when BN calculates, so that the mini-batch size does not need to be large, and then a GPU can achieve better results.

In practice, Mosaic data enhancement first removes one batch of data from the total data set, takes out four images at random from it each time, crops and splices them at random positions, synthesizes new images, repeats the batch size several times, and finally gets a new batch size of one batch of images after mosaic data enhancement, then feeds to the neural network for training.

When cropping and splicing the images, the four randomly obtained images are cropped by a randomly positioned crosshair, and the corresponding parts are taken for splicing. At the same time, the target frame of each original image is limited by the crosshair crop, and will not exceed the original crop range. The implementation of Mosaic data enhancement in practice is shown in Figure [3](#_bookmark3).



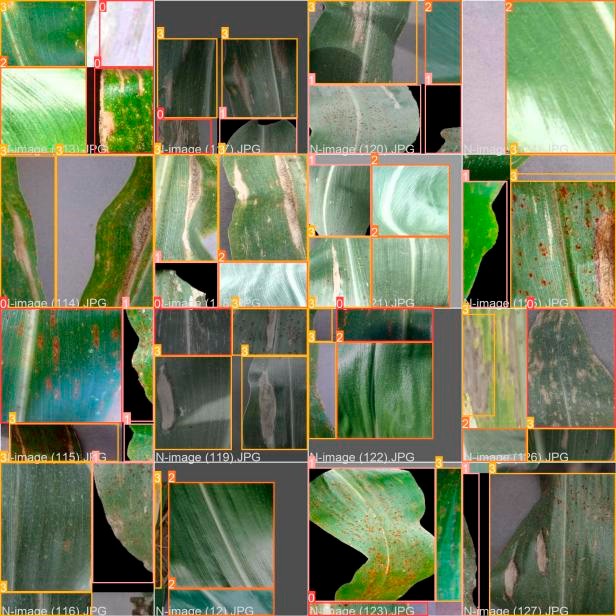
**Figure 3.** Implementation of Mosaic data enhancement in practice.

Mosaic has the following advantages: increases data diversity; randomly selects four images for combination; the number of images obtained from the combination is more than the number of original images; enhances model robustness; mixes four images with different semantic information; allows the model to detect targets beyond the conventional context; and enhances the effect of batch normalization. When the model is set to BN operation, the training will increase the total number of samples (BatchSize) as much as possible, because the BN principle is to calculate the mean and variance of each feature layer; if the total number of samples is larger, then the mean and variance calculated by BN will be closer to the mean and variance of the whole dataset, and the better the effect. The Mosaic data enhancement algorithm is helpful to improve the performance of small target detection. The enhanced images are stitched together from four original images, so that each image has a higher probability of containing small targets.

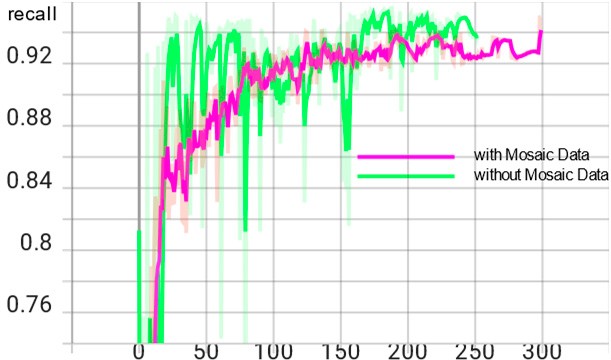
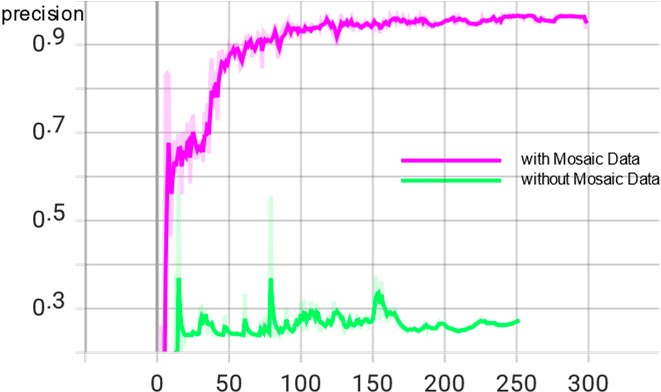
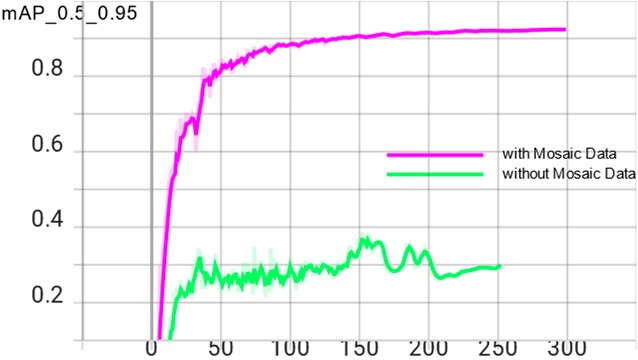
The operation principle of Mosaic data enhancement is equivalent to passing in four images for learning at one time during the training process, increasing the number of single training samples and target diversity, improving network training convergence speed and detection accuracy, and reducing large samples to small samples randomly, increasing the number of small-scale targets. Since the target of this paper is maize leaf disease and the disease spot is a small-scale target, Mosaic data enhancement provides important help for this study. Figure [4](#_bookmark4) shows 16 examples of data enhanced by Mosaic data. The file name in the picture is the file name of the image data involved in data enhancement. In the example, the file name is only for demonstration and will not be integrated into the picture to affect the subsequent model training. The colored boxes in the figure are identification boxes, where 0 indicates gray spot, 1 indicates rust, 2 indicates healthy maize leaves, and 3 indicates large spot disease. As shown in Figure [4](#_bookmark4).

To verify that the Mosaic data enhancement is real and effective for the experimental effect, this paper compares the parameters of the YOLOv5n network model with and without Mosaic data enhancement. The effect is shown in Figure [5](#_bookmark5).

From the Figure [5](#_bookmark5), it can be seen that the accuracy of the model is significantly and substantially improved after adding Mosaic data augmentation, and the convergence of the model is significantly improved compared to that without Mosaic data augmentation. At the same time, it can be seen that due to the Early Stopping method in the YOLOv5n model, which can resist overfitting, the model terminates early after 252 iterations in the training curve without the Mosaic data augmentation, because the accuracy no longer improves.



**Figure 4.** Sixteen examples of Mosaic data enhancement.



(**a**) (**b**) (**c**)

**Figure 5.** Comparison of enhancement effects with and without Mosaic data. (**a**) mAP\_0.5\_0.95 curve chart; (**b**) precision curve chart; (**c**) recall curve chart.

* 1. *YOLOv5 Network Model*

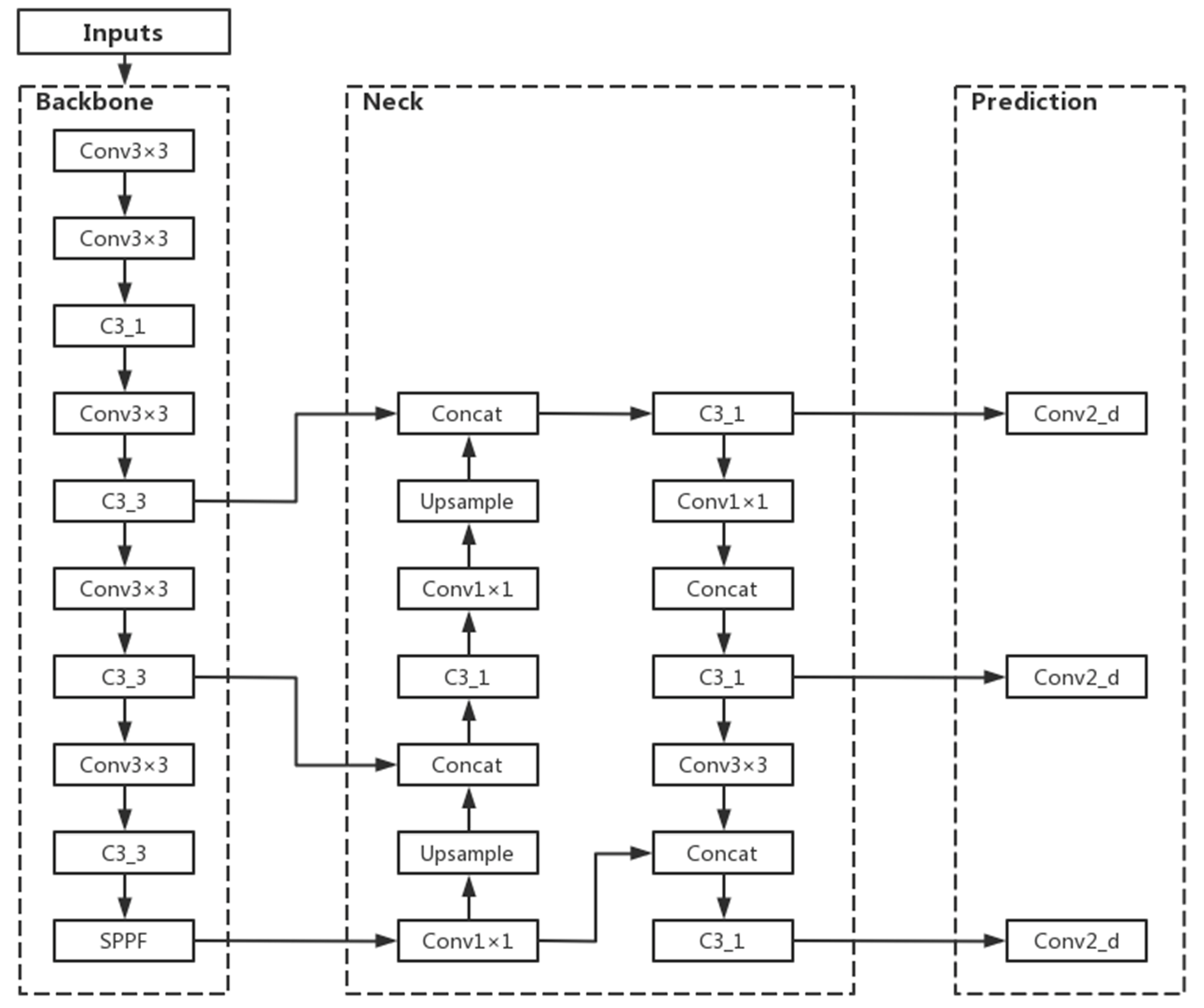
The YOLOv5 target detection algorithm is the 5th version of YOLO, whose core idea is to use the whole map as the input of the network and regress the location coordinates and category of the target directly in the output layer, which is characterized by high detection accuracy and fast detection speed to meet the demand of real-time monitoring.

The YOLOv5 network has been updated with five versions, YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x in order, with similar network structures and changes in network depth and width of feature maps based on YOLOv5s. Its accuracy and inference speed follow, of which YOLOv5n is in 2021 October after YOLOv5 update version 6.0, which has the advantage of being the fastest and the smallest model size compared to other versions. The ultimate goal is to deploy the model to mobile for real-time detection. To meet the lightweight requirement, the final study of this paper decided to use the YOLOv5n detection model with the lowest complexity to reduce the model storage footprint and increase the recognition speed.

The YOLOv5n algorithm consists of four parts: input, backbone, neck, and predic- tion [[19](#_bookmark33)]. Among them, Mosaic data enhancement is beneficial for detecting small targets and is suitable for leaf disease identification in this paper. The adaptive image scaling operation fixes images of different sizes to 640 pixels 640 pixels as input. In the backbone network, YOLOv5n mainly uses the Conv module CSP structure and SPPF module. The feature fusion stage mainly borrows the idea from PANet [[20](#_bookmark34)]. The FPN (Feature Pyramid Network) and PAN (Path Aggregation Network) are borrowed to form the FPN + PAN

*×*

structure. The prediction output continues the previous idea of YOLO by outputting three sizes of prediction maps at the same time, which are suitable for detecting small, medium and large targets. The network structure of YOLOv5n is shown in Figure [6](#_bookmark6).



**Figure 6.** YOLOv5n network structure.

* 1. *Improvements to the YOLOv5n Model*
     1. Adding CA to Improve Model Accuracy

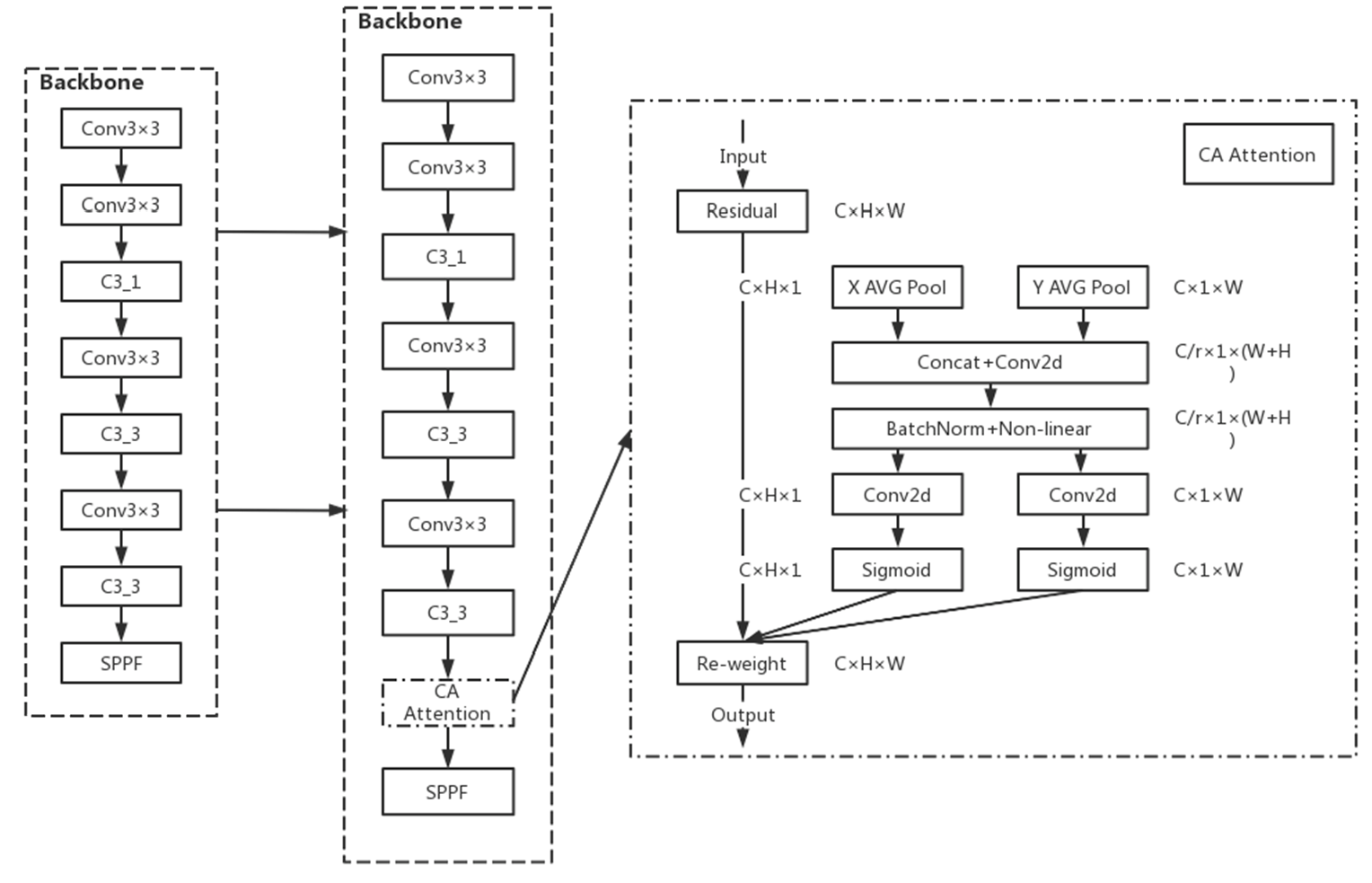
In the task of maize leaf disease detection, since the disease spots occupy relatively few pixels of the image, their feature information is easily lost in the deep network, resulting in errors such as the wrong detection and missed detection. At this point, it would be more beneficial for the network model to recognize the images if the unsupervised network can automatically acquire the ability to focus on smaller pixel blocks. Therefore, this paper introduces the CA (Coordinate Attention) mechanism [[21](#_bookmark35)] in the YOLOv5n backbone network, which is used to tell the model “what” and “where,” and which has been widely studied and deployed to improve the performance of neural networks. The use of lightweight attention modules can improve the network’s ability to extract features from maize leaf spots while saving parameters.

For other channel attentions, they are taken to transform the input into individual feature vectors by 2D global pooling. The general idea of Coordinate Attention used in this paper is to decompose channel attention into two 1D feature encodings of aggregated features along different directions in the H-direction as well as the W-direction, that is, into *C H* 1 and *C* 1 *W*. CA This idea has the advantage of capturing long-range de- pendencies along one spatial direction while retaining accurate location information along the other spatial direction. After that, the generated feature maps are encoded separately, resulting in two direction-aware, as well as position-sensitive, feature maps, which can be complementarily applied to the input feature maps to enhance the representation of the target of interest. The two directional feature maps are then Concept spliced and then fed into a shared convolution to reduce the dimensionality to C/r, after which they are separated and allowed to Sigmoid in different directions to obtain the coefficients and then multiplied together. Finally, the feature map is obtained.

*× × × ×*

After adding CA attention to the YOLOv5n backbone network [[22](#_bookmark36)], keeping the parameters unchanged, the model is trained again, and the trained model has significantly improved the effect compared with the original model; the average accuracy mean value is increased from 0.924 to 0.948, and the model size is not significantly increased, which meets the requirement of being lightweight.

In this paper, after adding CA attention to the YOLOv5n backbone network, the specific structure of the backbone network is shown in Figure [7](#_bookmark7).



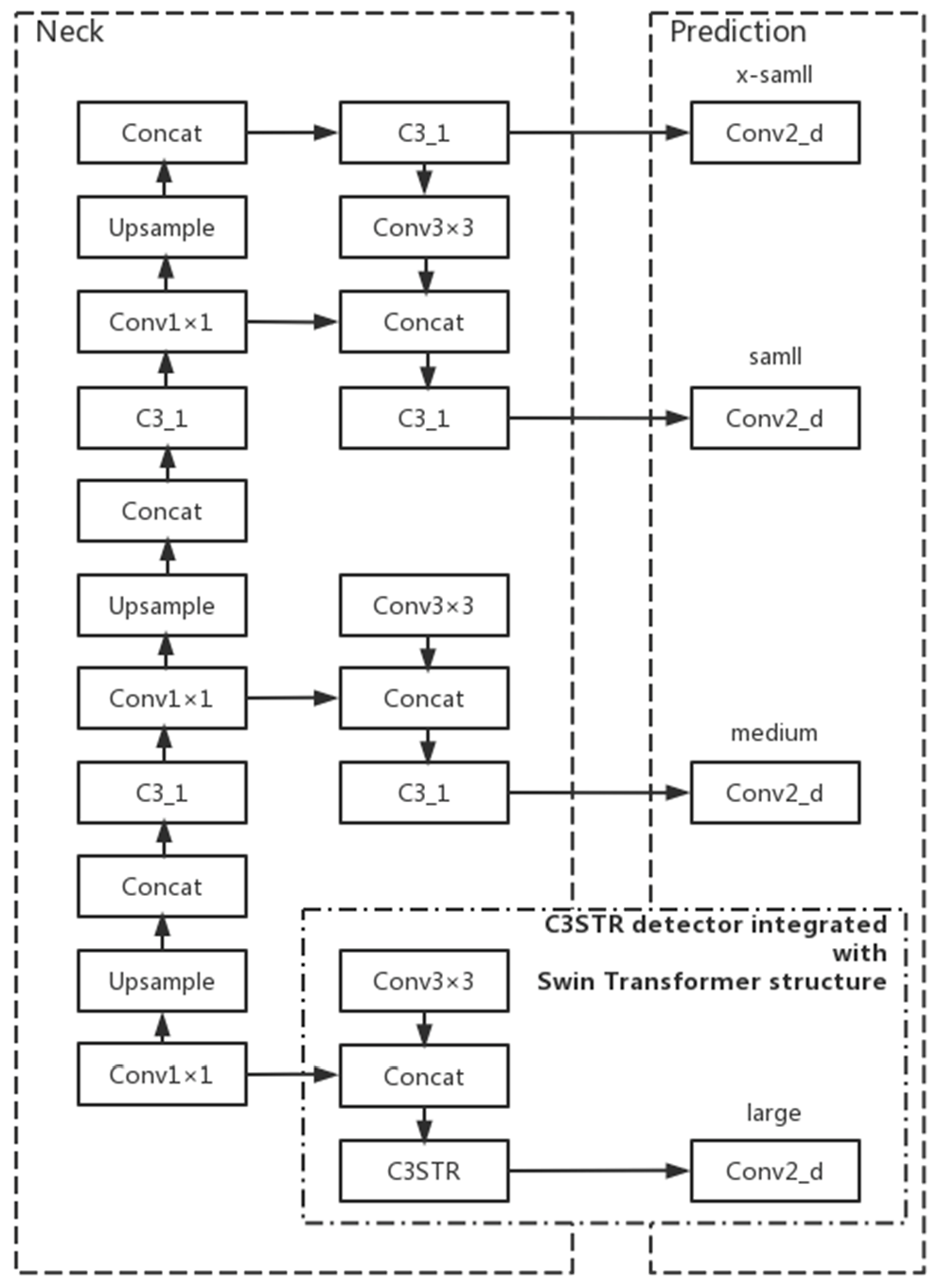
**Figure 7.** YOLOv5n’s integration into CA attention mechanism.

* + 1. Incorporating Swin Transformer Structure to Improve Model Generalization Performance In the detection of maize leaf spots, the distribution of different types of spots in

the leaf images differs: large spots occupy a small area of the leaf and rely more on local information of high-level features; rusts have a large distribution area and rely on global information more obviously; gray spots are moderate in size and rely on both local and global information. The performance of Convolutional Neural Networks (CNN) is more capable of capturing local information and has a certain disadvantage in global information acquisition. To alleviate the adverse effects of the non-uniform spot size, the model is improved by extracting global information using Swin Transformer [[23](#_bookmark37)]. In this paper, a smaller size target detection head was added to the original small, medium and large size detection heads of the YOLOv5n model to enhance its ability to identify small targets of the spots. The x-small size detection head in Figure [8](#_bookmark8), and the Swin Transformer structure, was incorporated into the large size detection head to replace the original C3 structure with the C3STR structure incorporated into the large size detection head to change the original C3 structure to C3STR structure, thus improving the model’s capture of feature information.The improved network structure is shown in Figure [8](#_bookmark8).

The Swin Transformer model was proposed by Microsoft Research in 2021. Swin Transformer uses hierarchical feature maps similar to those used in convolutional neural networks, such as feature map sizes with 4 , 8 , and 16 down-sampling of images, such that the backbone helps to build on top of this for tasks such as target detection, instance segmentation, etc. The Swin Transformer network is another collision of the Transformer model in the field of vision. The Swin Transformer network is another collision of Transformer model in vision field.

*× × ×*



**Figure 8.** Add x-small detection head and Swin Transformer structure.

The concept of Windows Multi-Head Self-Attention (W-MSA) is used in Swin Trans- former, for example, in the 4-fold downsampling and 8-fold downsampling in the figure below. The feature map is divided into multiple disjointed regions (Window), and Multi- Head Self-Attention is performed only within each window (Window).

The basic flow of the whole framework is as follows.

First, the image is input to the patch partition module for chunking, i.e., every 4 4 ad- jacent pixels is a patch, and then it is flattened in the channel direction. Assuming that the input is a three-channel RGB image, each patch has 4 4 = 16 pixels, and each pixel has three values, R, G, and B. The flattened image shape is 16 3 = 48, so the image shape changes from [H, W, 3] to [H/4, W/4, 48] after patch partition. Then the channel data of each pixel is linearly transformed by the linear embedding layer from 48 to C, i.e., the image shape is changed from [H/4, W/4, 48] to [H/4, W/4, C].

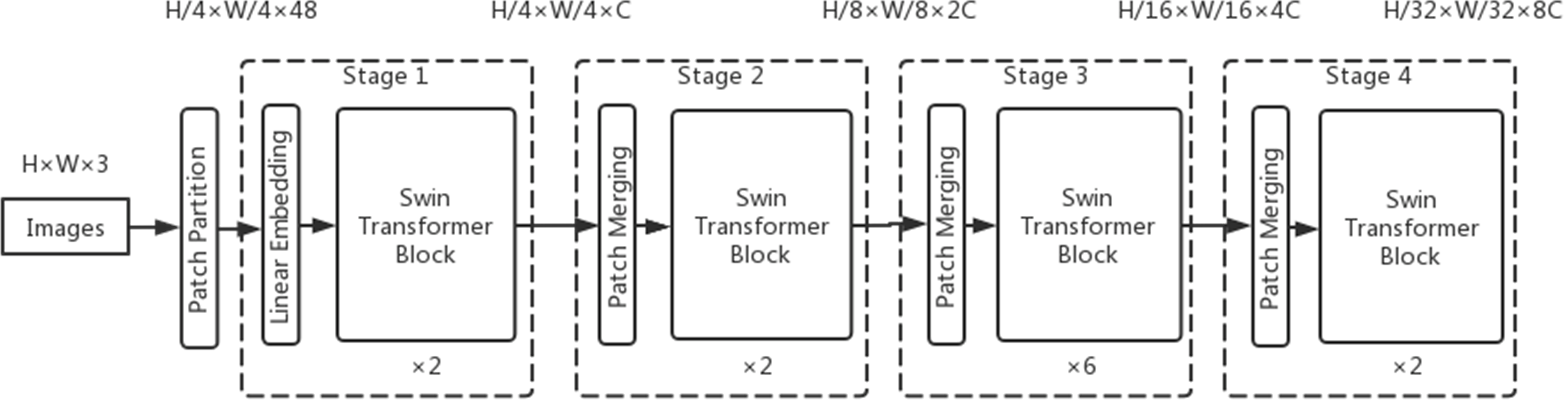
*×*

*×*

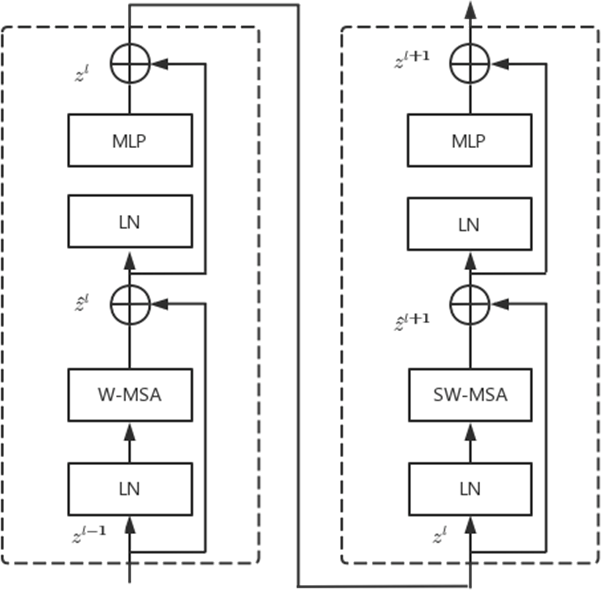
*×*

The Swin Transformer divides patches by first determining the size of each patch and then calculating the number of patches. The number of Swin Transformers decreases, and the perceptual range of each patch expands as the network depth deepens, which is designed to facilitate the construction of layers of Swin Transformer and to adapt to the multi-scale of visual tasks.The architecture of a Swin transformer (Swin-T) is shown in Figure [9](#_bookmark9).

Then the feature maps of different sizes are constructed by four stages; except for Stage 1, where a linear embedding layer is first passed, the remaining three stages are first downsampled by a patch merging layer. Note that there are two types of blocks, as shown in Figure [10](#_bookmark10), which differ only in that one uses the W-MSA structure and the other uses the SW-MSA structure. Moreover, these two structures are used in pairs, with one W-MSA structure used first and then one SW-MSA structure used.



**Figure 9.** The architecture of a Swin Transformer (Swin-T).



**Figure 10.** Two successive Swin Transformer blocks.

* 1. *Test Environment*

In this paper, we use the deep learning framework PyTorch to build and improve the model in the Anaconda3 environment, and train and test the model under the Windows 10 system. The computer CPU is 11th Gen Intel (R) Core (TM) i7–11700K @ 3.60 GHz and the GPU is NVIDIA GeForce GTX 1080 Ti. The GPU is used for acceleration to improve the network training speed, the Cuda version is 11.1.0, and the cudnn version is 8.1.0.

* 1. *Evaluation Metrics*

In this paper, the performance of the YOLOv5n network model is evaluated using several metrics from the target detection algorithm, specifically Precision (P), Recall (R), and mean Average Precision (mAP) [[24](#_bookmark38)]. Average Precision (AP) is the integral of the PR curve formed by taking Precision (P) as the vertical axis and Recall (R) as the horizontal axis. A recall is a metric that reflects the ability of the model to find positive sample targets, precision is used to reflect the ability of the model to classify samples, and average precision is a metric that reflects the overall performance of the model to detect and classify targets. The mean Average Precision (mAP) represents the average of the mean accuracy of all categories. Among all the metrics, mAP is the most important evaluation metric in the target detection algorithm, which can measure the accuracy of the detection algorithm. mAP0.5 is the AP of the target detection model evaluated at an IoU threshold of 0.5. mAP0.5 is its mean value for all categories; AP0.5–0.95 is the mean value of the AP of the model evaluated at different IoU thresholds (0.5–0.95, step size 0.05); and AP0.5–0.95 is the average value of AP evaluated under different IoU thresholds (0.5–0.95, step size 0.05), which is a more stringent model accuracy index. mAP0.5–0.95 is the average value of all categories. In this paper, we choose mAP0.5–0.95 as the evaluation index. The additional evaluation index considered in this paper is the number of parameters, and the number of parameters indicates the size of the storage space occupied by the model file in MB.

The expressions for the calculation of Precision (P), Recall (R), Average Precision (AP), and mean Average Precision (mAP) are shown in Equations (1)–(4).

P = TP

TP + FP

R = TP

TP + FN

1

∫

(1)

(2)

AP =

0

P*·*RdR (3)

mAP =

N

i=1

∑

APi

(4)

N

where:

* The number of true samples.
* The number of false positive samples.
* Number of pseudo-negative samples.
* The number of species in the sample.

The positive and negative samples are judged by setting the average Intersection over Union (IoU) threshold between the predicted target area and the actual target area, and if the IoU of both exceeds the threshold, the sample is positive, and if vice versa, the sample is negative.

# 3.Results

1. **Results**
   1. *Model Trainsing Hyperparameter Setting*

In the model training stage, the total number of training rounds is set to 300 and the iteration batch\_size is set to 16. Setting too small a value for iteration batch size will lead to too slow training, and setting too large a value for iteration batch size will lead to insufficient video memory to run the experiment. In the experimental environment of this paper, the maximum YOLOv5x model runs at 83%, so setting a larger value will lead to run failure, so it is considered that the size of 16 is a more appropriate level.

In the model training process, if the learning rate is adjusted too large, the network will not converge, while if the learning rate is adjusted too small, the network convergence speed will become slow, so the appropriate learning rate is a key factor in the training process. This paper uses three different sizes of initial learning rate to compare the model experiments; the trained model parameters are shown in Table [2](#_bookmark11).

**Table 2.** Parameter comparison under different initial learning rates.

**Model Initial Learning Rate Precision (P) Recall (R) Mean Average**

**Precision (mAP0.5–0.95)**

**Parameter Quantity/MB**

YOLOv5n

0.1 0.938 0.924 0.885 3.9

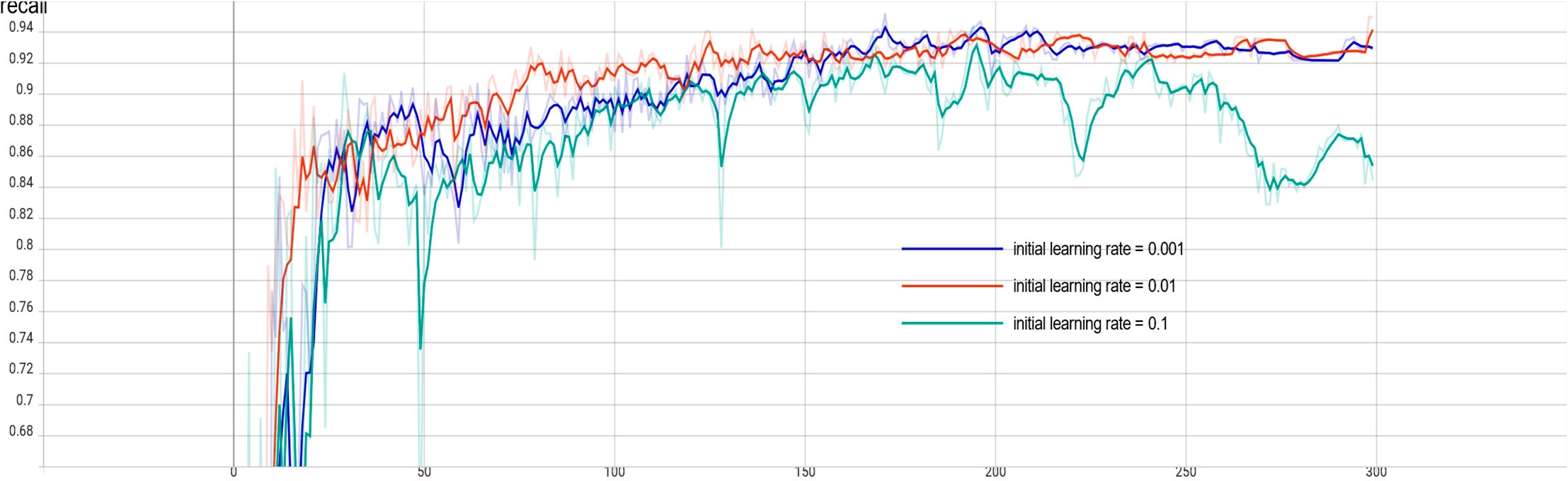
0.01 0.94 0.949 0.924 3.9

0.001 0.963 0.922 0.889 3.9

From the table, it can be seen that, for the YOLOv5n model studied in this paper, the average precision mean is highest at an initial learning rate of 0.01. In addition, in order to see the comparison of network convergence more intuitively, this paper shows the graphs of model recall for three different sizes of initial learning rates, and the effect is shown in Figure [11](#_bookmark12).

From the figure, it can be seen that the model converges fastest when the learning rate is 0.01, the model converges best and performs more stably, and its accuracy stays around the optimal value after 100 epochs, which avoids the interference of the model fluctuation by chance on the final result; when the learning rate is 0.1, the model recall

curve fluctuates more and converges extremely poorly; when the learning rate is 0.001. Although the convergence effect is better when the learning rate is 0.001, the convergence speed is not as fast as when the learning rate is set to 0.01, and the accuracy remains around the optimal value after 150 epochs. Therefore, the initial learning rate of 0.01 is used to carry out the subsequent experiments.



**Figure 11.** Comparison of recall rates under different initial learning rates.

To prevent the overfitting phenomenon, the weight decay coefficient is set to 0.0005, the confidence level is set to 0.5, and the non-maximum suppression threshold is set to 0.3.

* 1. *Comparison of Different Algorithm Models*

Since there are five versions of YOLOv5 so far, to ensure the authenticity, accuracy, and rigor of the experimental process, the five versions of YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x are trained on the same platform, the same framework with the same training parameters, and on the same data set in turn. The comparative data are shown in Table [3](#_bookmark13).

**Table 3.** Comparison of parameters of different network models.

**Model Precision (P) Recall (R) Mean Average**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | **Precision (mAP0.5–0.95)** | **Quantity/MB** |
| YOLOv5n | 0.94 | 0.949 | 0.924 | 3.9 |
| YOLOv5s | 0.973 | 0.942 | 0.932 | 14.5 |
| YOLOv5m | 0954 | 0.961 | 0.945 | 40.3 |
| YOLOv5l | 0.96 | 0.965 | 0.948 | 92.9 |
| YOLOv5x | 0.965 | 0.965 | 0.958 | 173.2 |

**Parameter**

From the above table, we can see that, due to the different depth and width of the models, among the five models of YOLOv5, the average accuracy of YOLOv5n is 0.924, which is the lowest accuracy among the five models, but its model occupies much less memory than the other models, only 3.9 M, and the training time of YOLOv5n is also much shorter than the other models during training. In contrast, the memory consumption of YOLOv5l and YOLOv5x models is around 100 MB after training with smaller data sets, which can be considered as suitable for deployment on large servers only.

Also, among the five models of YOLOv5, YOLOv5n has the lowest precision, but its model takes up far less memory than the other models, only 3.9 M, and the training time of YOLOv5n is also much shorter than the other models. In contrast, the memory consumption of the YOLOv5l and YOLOv5x models is around 100 MB after training with a small data set, which can be considered suitable for deployment on large servers only.

The ultimate goal of this paper is to help farmers accurately identify the diseases infecting maize leaves in real-time by using portable cell phones in the field so that timely

control can be carried out to reduce the adverse effects of low quality and yield of maize caused by the diseases.

Therefore, YOLOv5n is the most suitable lightweight model for mobile deployment to identify maize leaf spots in real-time. The model is also improved in the hope that the accuracy of the YOLOv5n model can be further enhanced to reach a similar level to other large network models without affecting the model size.

* 1. *Model Enhancement by Different Attention Mechanisms*

In this paper, the CTR\_YOLOv5n model is built by incorporating CA attention into the baseline model and using the Swin Transformer-based C3STR block as the prediction head to further improve the generalization ability of the model and improve the model accuracy. To verify the effectiveness of the improvements on the model, this paper adds the current mainstream attention mechanisms, such as SE (Squeeze-and-Excitation), CBAM (Convolutional Block Attention Module), and ECA (Efficient Channel Attention), to the model under the same platform and framework, using the same training parameters, and

the comparison results are shown in Table [4](#_bookmark14).

**Table 4.** Comparison of Yolov5n effect improvement under different attention mechanisms.

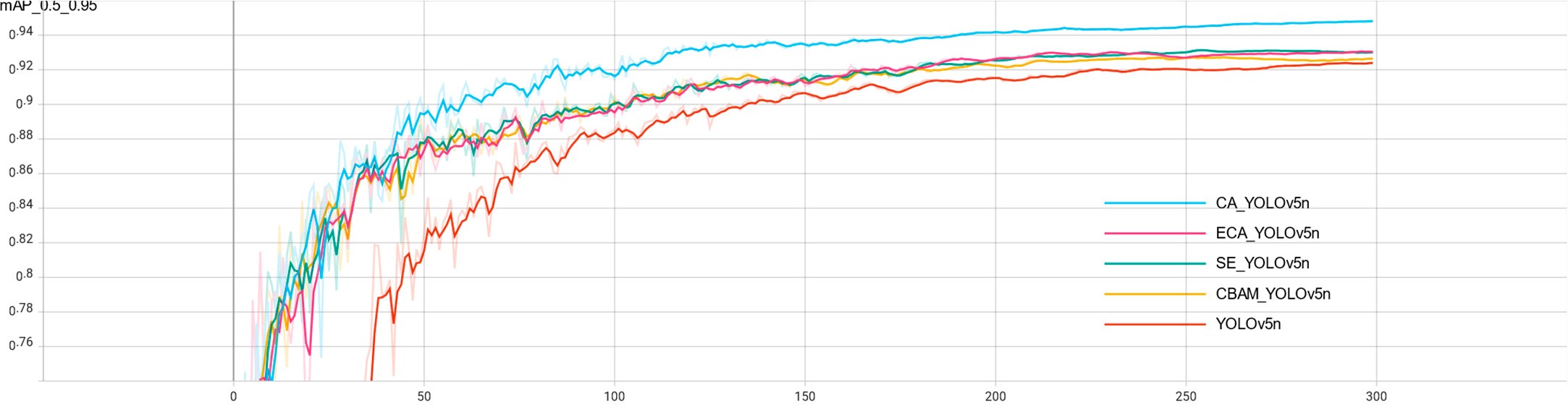
**Model Precision (P) Recall (R) Mean Average**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | **Precision (mAP0.5–0.95)** | **Quantity/MB** |
| YOLOv5n | 0.94 | 0.949 | 0.924 | 3.9 |
| SE\_YOLOv5n | 0.946 | 0.959 | 0.931 | 3.9 |
| CBAM\_YOLOv5n | 0.962 | 0.947 | 0.927 | 3.9 |
| ECA\_YOLOv5n | 0.961 | 0.944 | 0.931 | 3.9 |
| CA\_YOLOv5n | 0.948 | 0.961 | 0.948 | 3.9 |

**Parameter**

From the above table, we can see that the addition of the attention mechanism to the YOLOv5n backbone network improves the mean accuracy of YOLOv5n, and there is no significant change in the model size, which proves that the attention mechanism has an improvement effect on YOLOv5n. Among the mainstream attention mechanisms, the SE attention mechanism improves the mean accuracy of YOLOv5n by 0.7 percentage points, CBAM by 0.3 percentage points, ECA by 0.7 percentage points, and the improved CA\_YOLOv5n by 2.4 percentage points. In this study, the accuracy of the CA\_YOLOv5n network model is improved to the same level as YOLOv5l while keeping the model size the same as YOLOv5n, confirming that the improvements in this paper can allow CA\_YOLOv5n to be deployed on mobile devices with an accuracy similar to that of the server-deployed model.

In order to see the improvement effect of different attention mechanisms on YOLOv5n more intuitively, this paper shows the accuracy comparison of YOLOv5n with different attention mechanisms added, and the effect is shown in Figure [12](#_bookmark15).



**Figure 12.** Precision comparison of different attention mechanisms.

From the figure, we can see that the accuracy of the YOLOv5n network without the attention mechanism starts to increase rapidly around 40 training sessions, and the increase in accuracy starts to slow down around 100 training sessions, while the accuracy of the model with the attention mechanism starts to increase rapidly around 15 training sessions, in which CA\_YOLOv5n has the fastest growth rate and the largest increase in the first 50 training sessions, and the accuracy is always higher than the other models in the subsequent training sessions. The accuracy of CA\_YOLOv5n is consistently higher than the other models in the subsequent training.

Comparing CA with other mainstream attention mechanisms, CA attention takes into account not only the relationship between channels but also the location information in the feature space. CA uses a more efficient method to capture location information and channel relationships. It does this by decomposing the two-dimensional global pooling operation into two directions, width and height, and then averaging the pools globally to obtain feature maps in the two different directions, width and height. The feature maps in both directions are then processed to obtain the final feature maps, which is the main reason why the CA attention is more effective than other attentions.

The experiments in this paper can prove that the model improvement approach studied in this paper is better than adding other mainstream attention mechanisms. In addition, from Figure [6](#_bookmark6), we can see that all the models reach a relatively flat accuracy after 200 training sessions, which indicates that the proposed model achieves the ideal state.

* 1. *Model Improvement to Improve the Effect*

Based on the CA\_YOLOv5n model, the experiments in this paper make further im- provements by adding a small target detection layer to the neck network neck of the CA\_YOLOv5n model to enhance its ability to recognize small targets of disease spots, and by replacing the original C3 detection head into the Swin Transformer structure with the C3STR detection head to improve the model’s capture of feature information.

The following Table [5](#_bookmark16) shows the integrated inference results of each model for dif- ferent types of maize leaf diseases at 256 256 pixels. The YOLOv5n model showed high recognition ability for maize rust and healthy maize leaves, reaching 0.953 and 0.99, respectively, but low recognition rate for maize gray spot disease and maize blotch disease, reaching only 0.905 and 0.849. The overall accuracy of the model was further improved by adding the CA attention mechanism. Compared with YOLOv5n, the model CA\_YOLOv5n showed a significant improvement in the recognition performance of maize gray spot disease and maize blotch disease, which was comparable to the performance of YOLOv5l model. The final proposed model, CTR\_YOLOv5n with mAP0.5–0.95, improved by 2.8% compared to the initial model YOLOv5n and by 0.4% compared to the YOLOv5l model. It is noteworthy that the mAP0.5–0.95 of model CTR\_YOLOv5n reached 0.928 and 0.925 for maize gray spot disease and maize blotch disease, which were significantly improved compared with each model, proving that the accuracy of the final model proposed in this paper is effectively improved for small target detection of disease spots. The accuracy distribution of the CTR\_YOLOv5n model is more balanced, which shows the effectiveness of the improvement strategy of this paper. Although the number of parameters in the final model is slightly increased compared with YOLOv5n, it still meets the requirements of being lightweight.

*×*

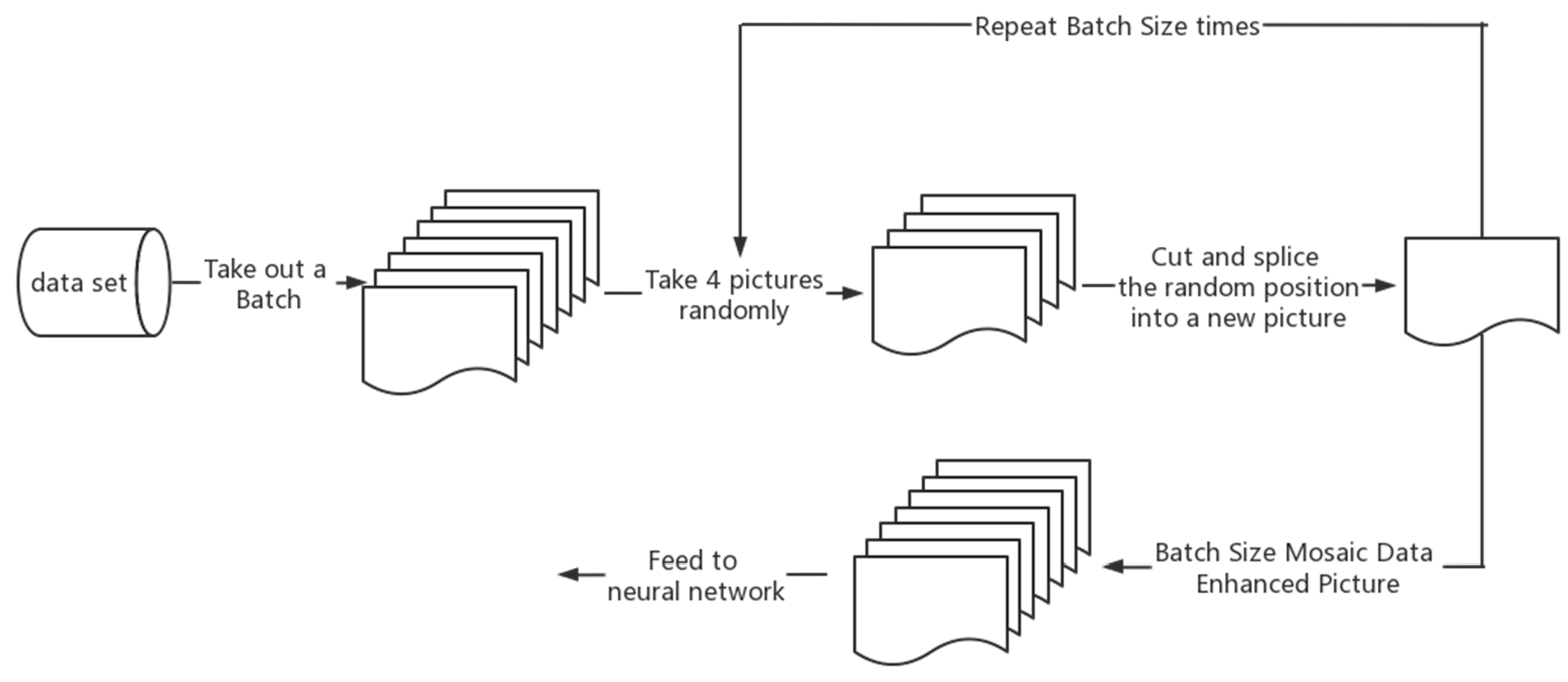
**Table 5.** Performance of each model under different disease categories.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model Maize Gray Maize Rust Healthy Maize** | | | | **Maize Blotch Mean Average Parameter** | | |
|  | **Spot Disease** |  | **Leaves** | **Disease** | **(mAP0.5–0.95)** | **Quantity/MB** |
| YOLOv5n | 0.905 | 0.953 | 0.99 | 0.849 | 0.924 | 3.9 |
| YOLOv5l | 0.916 | 0.986 | 0.992 | 0.898 | 0.948 | 92.9 |
| CA\_YOLOv5n | 0.924 | 0.985 | 0.992 | 0.891 | 0.948 | 3.9 |
| CTR\_YOLOv5n | 0.928 | 0.964 | 0.99 | 0.925 | 0.952 | 5.1 |

**Precision**

It is confirmed that the improvements in this paper allow CTR\_YOLOv5n to be deployed on mobile with an accuracy comparable to that of the server-deployed model.

In the YOLOv5n network model described in this paper, not only are some basic data enhancement methods included, but also the Mosaic data enhancement [[18](#_bookmark32)] is used, whose main idea is to select four images from the used dataset, crop and scale them randomly, and then arrange them randomly to form a new image. This has the advantage of increasing the number of datasets while augmenting the number of small sample targets, and it improves the training speed of the model. The flowchart of Mosaic data enhancement is shown in Figure [2](#_bookmark2).

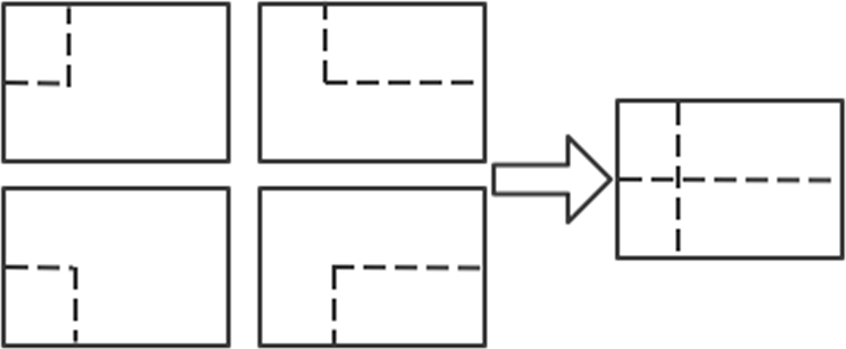


**Figure 2.** Mosaic data enhancement flow chart.

Mosaic data enhancement utilizes four images, which enriches the background of the detected objects and calculates the data of four images at once when BN calculates, so that the mini-batch size does not need to be large, and then a GPU can achieve better results.

In practice, Mosaic data enhancement first removes one batch of data from the total data set, takes out four images at random from it each time, crops and splices them at random positions, synthesizes new images, repeats the batch size several times, and finally gets a new batch size of one batch of images after mosaic data enhancement, then feeds to the neural network for training.

When cropping and splicing the images, the four randomly obtained images are cropped by a randomly positioned crosshair, and the corresponding parts are taken for splicing. At the same time, the target frame of each original image is limited by the crosshair crop, and will not exceed the original crop range. The implementation of Mosaic data enhancement in practice is shown in Figure [3](#_bookmark3).



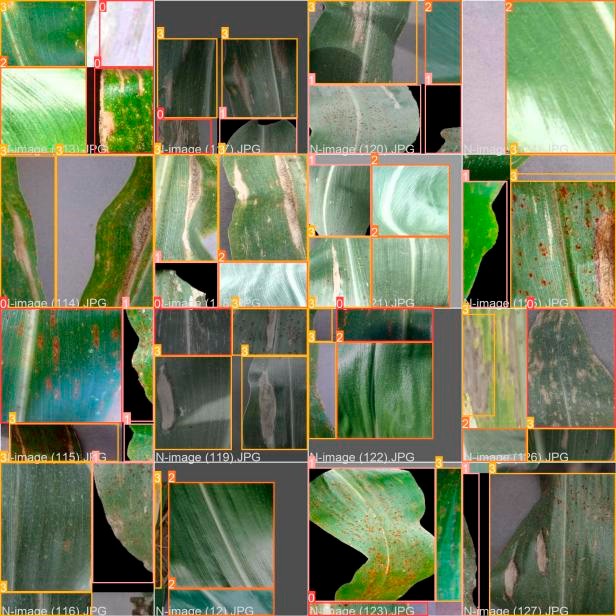
**Figure 3.** Implementation of Mosaic data enhancement in practice.

Mosaic has the following advantages: increases data diversity; randomly selects four images for combination; the number of images obtained from the combination is more than the number of original images; enhances model robustness; mixes four images with different semantic information; allows the model to detect targets beyond the conventional context; and enhances the effect of batch normalization. When the model is set to BN operation, the training will increase the total number of samples (BatchSize) as much as possible, because the BN principle is to calculate the mean and variance of each feature layer; if the total number of samples is larger, then the mean and variance calculated by BN will be closer to the mean and variance of the whole dataset, and the better the effect. The Mosaic data enhancement algorithm is helpful to improve the performance of small target detection. The enhanced images are stitched together from four original images, so that each image has a higher probability of containing small targets.

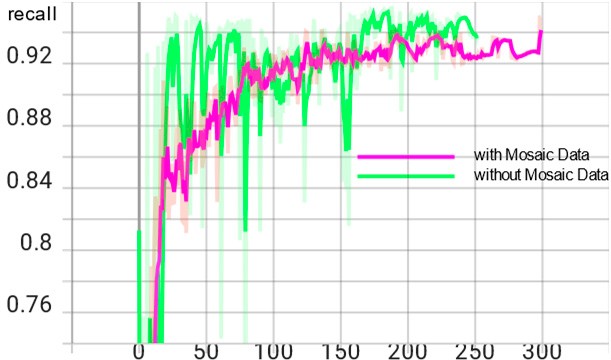
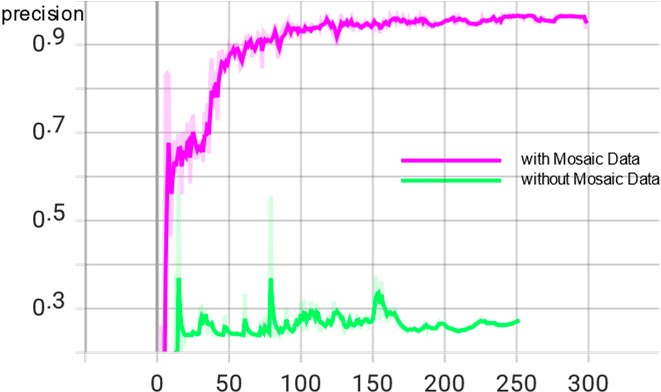
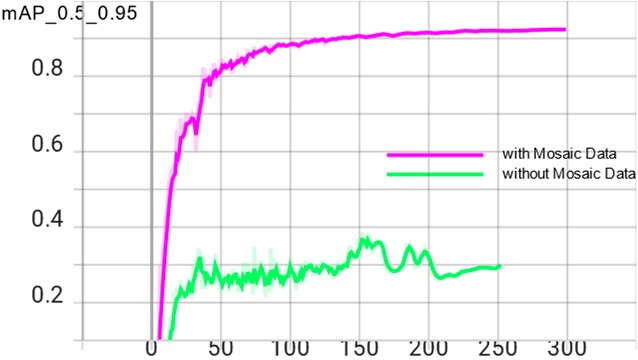
The operation principle of Mosaic data enhancement is equivalent to passing in four images for learning at one time during the training process, increasing the number of single training samples and target diversity, improving network training convergence speed and detection accuracy, and reducing large samples to small samples randomly, increasing the number of small-scale targets. Since the target of this paper is maize leaf disease and the disease spot is a small-scale target, Mosaic data enhancement provides important help for this study. Figure [4](#_bookmark4) shows 16 examples of data enhanced by Mosaic data. The file name in the picture is the file name of the image data involved in data enhancement. In the example, the file name is only for demonstration and will not be integrated into the picture to affect the subsequent model training. The colored boxes in the figure are identification boxes, where 0 indicates gray spot, 1 indicates rust, 2 indicates healthy maize leaves, and 3 indicates large spot disease. As shown in Figure [4](#_bookmark4).

To verify that the Mosaic data enhancement is real and effective for the experimental effect, this paper compares the parameters of the YOLOv5n network model with and without Mosaic data enhancement. The effect is shown in Figure [5](#_bookmark5).

From the Figure [5](#_bookmark5), it can be seen that the accuracy of the model is significantly and substantially improved after adding Mosaic data augmentation, and the convergence of the model is significantly improved compared to that without Mosaic data augmentation. At the same time, it can be seen that due to the Early Stopping method in the YOLOv5n model, which can resist overfitting, the model terminates early after 252 iterations in the training curve without the Mosaic data augmentation, because the accuracy no longer improves.



**Figure 4.** Sixteen examples of Mosaic data enhancement.



(**a**) (**b**) (**c**)

**Figure 5.** Comparison of enhancement effects with and without Mosaic data. (**a**) mAP\_0.5\_0.95 curve chart; (**b**) precision curve chart; (**c**) recall curve chart.

* 1. *YOLOv5 Network Model*

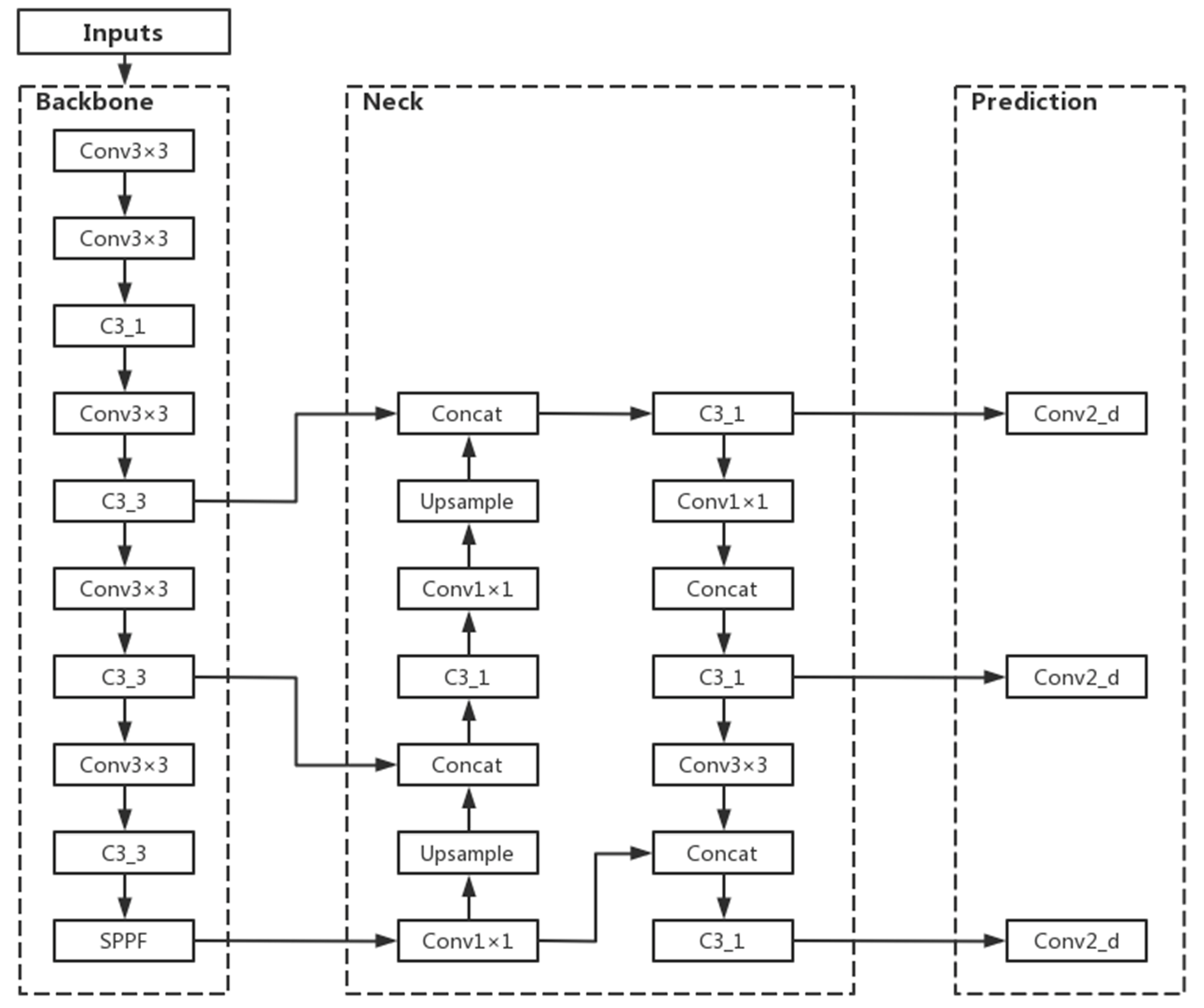
The YOLOv5 target detection algorithm is the 5th version of YOLO, whose core idea is to use the whole map as the input of the network and regress the location coordinates and category of the target directly in the output layer, which is characterized by high detection accuracy and fast detection speed to meet the demand of real-time monitoring.

The YOLOv5 network has been updated with five versions, YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x in order, with similar network structures and changes in network depth and width of feature maps based on YOLOv5s. Its accuracy and inference speed follow, of which YOLOv5n is in 2021 October after YOLOv5 update version 6.0, which has the advantage of being the fastest and the smallest model size compared to other versions. The ultimate goal is to deploy the model to mobile for real-time detection. To meet the lightweight requirement, the final study of this paper decided to use the YOLOv5n detection model with the lowest complexity to reduce the model storage footprint and increase the recognition speed.

The YOLOv5n algorithm consists of four parts: input, backbone, neck, and predic- tion [[19](#_bookmark33)]. Among them, Mosaic data enhancement is beneficial for detecting small targets and is suitable for leaf disease identification in this paper. The adaptive image scaling operation fixes images of different sizes to 640 pixels 640 pixels as input. In the backbone network, YOLOv5n mainly uses the Conv module CSP structure and SPPF module. The feature fusion stage mainly borrows the idea from PANet [[20](#_bookmark34)]. The FPN (Feature Pyramid Network) and PAN (Path Aggregation Network) are borrowed to form the FPN + PAN

*×*

structure. The prediction output continues the previous idea of YOLO by outputting three sizes of prediction maps at the same time, which are suitable for detecting small, medium and large targets. The network structure of YOLOv5n is shown in Figure [6](#_bookmark6).



**Figure 6.** YOLOv5n network structure.

* 1. *Improvements to the YOLOv5n Model*
     1. Adding CA to Improve Model Accuracy

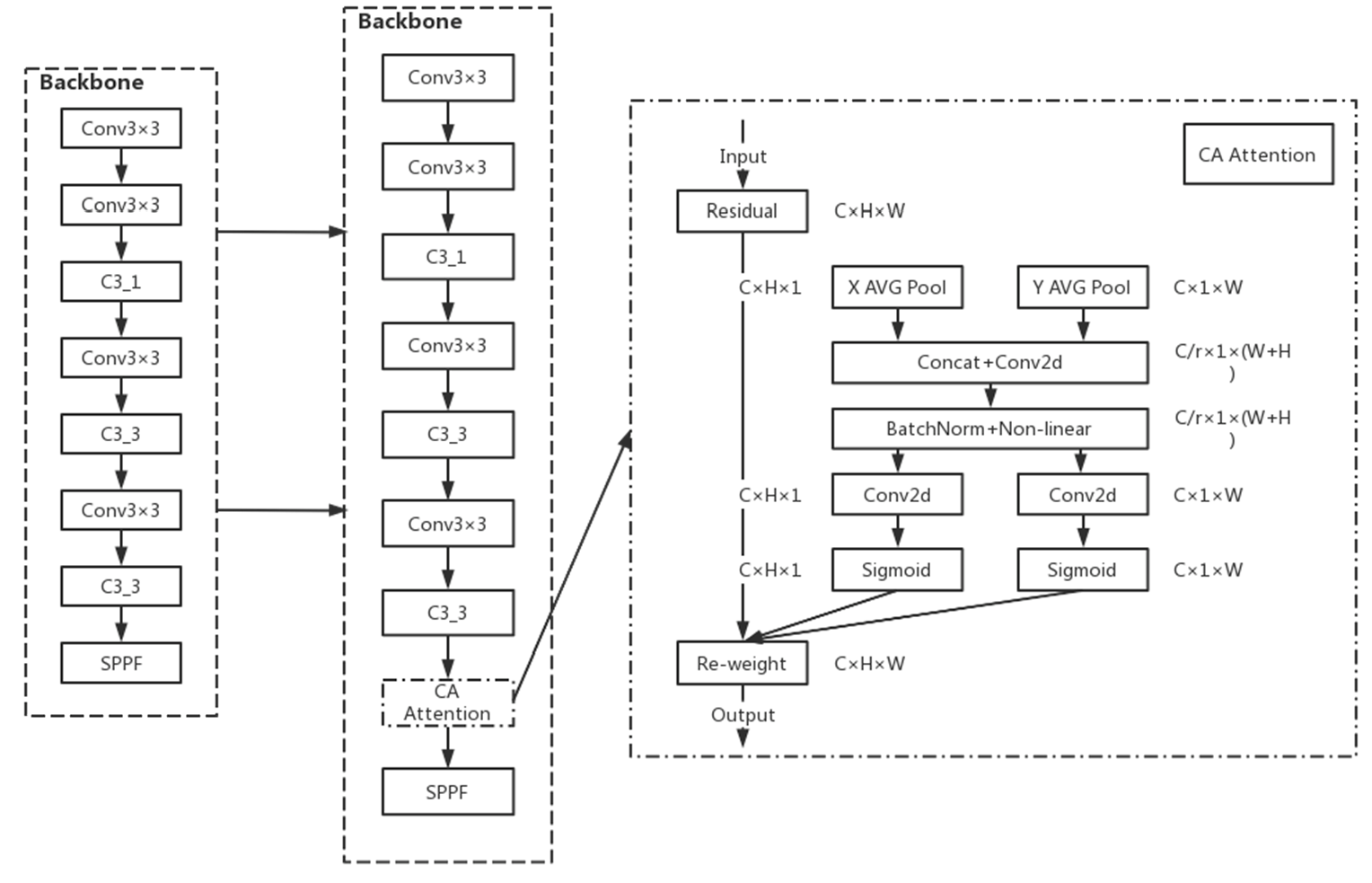
In the task of maize leaf disease detection, since the disease spots occupy relatively few pixels of the image, their feature information is easily lost in the deep network, resulting in errors such as the wrong detection and missed detection. At this point, it would be more beneficial for the network model to recognize the images if the unsupervised network can automatically acquire the ability to focus on smaller pixel blocks. Therefore, this paper introduces the CA (Coordinate Attention) mechanism [[21](#_bookmark35)] in the YOLOv5n backbone network, which is used to tell the model “what” and “where,” and which has been widely studied and deployed to improve the performance of neural networks. The use of lightweight attention modules can improve the network’s ability to extract features from maize leaf spots while saving parameters.

For other channel attentions, they are taken to transform the input into individual feature vectors by 2D global pooling. The general idea of Coordinate Attention used in this paper is to decompose channel attention into two 1D feature encodings of aggregated features along different directions in the H-direction as well as the W-direction, that is, into *C H* 1 and *C* 1 *W*. CA This idea has the advantage of capturing long-range de- pendencies along one spatial direction while retaining accurate location information along the other spatial direction. After that, the generated feature maps are encoded separately, resulting in two direction-aware, as well as position-sensitive, feature maps, which can be complementarily applied to the input feature maps to enhance the representation of the target of interest. The two directional feature maps are then Concept spliced and then fed into a shared convolution to reduce the dimensionality to C/r, after which they are separated and allowed to Sigmoid in different directions to obtain the coefficients and then multiplied together. Finally, the feature map is obtained.

*× × × ×*

After adding CA attention to the YOLOv5n backbone network [[22](#_bookmark36)], keeping the parameters unchanged, the model is trained again, and the trained model has significantly improved the effect compared with the original model; the average accuracy mean value is increased from 0.924 to 0.948, and the model size is not significantly increased, which meets the requirement of being lightweight.

In this paper, after adding CA attention to the YOLOv5n backbone network, the specific structure of the backbone network is shown in Figure [7](#_bookmark7).



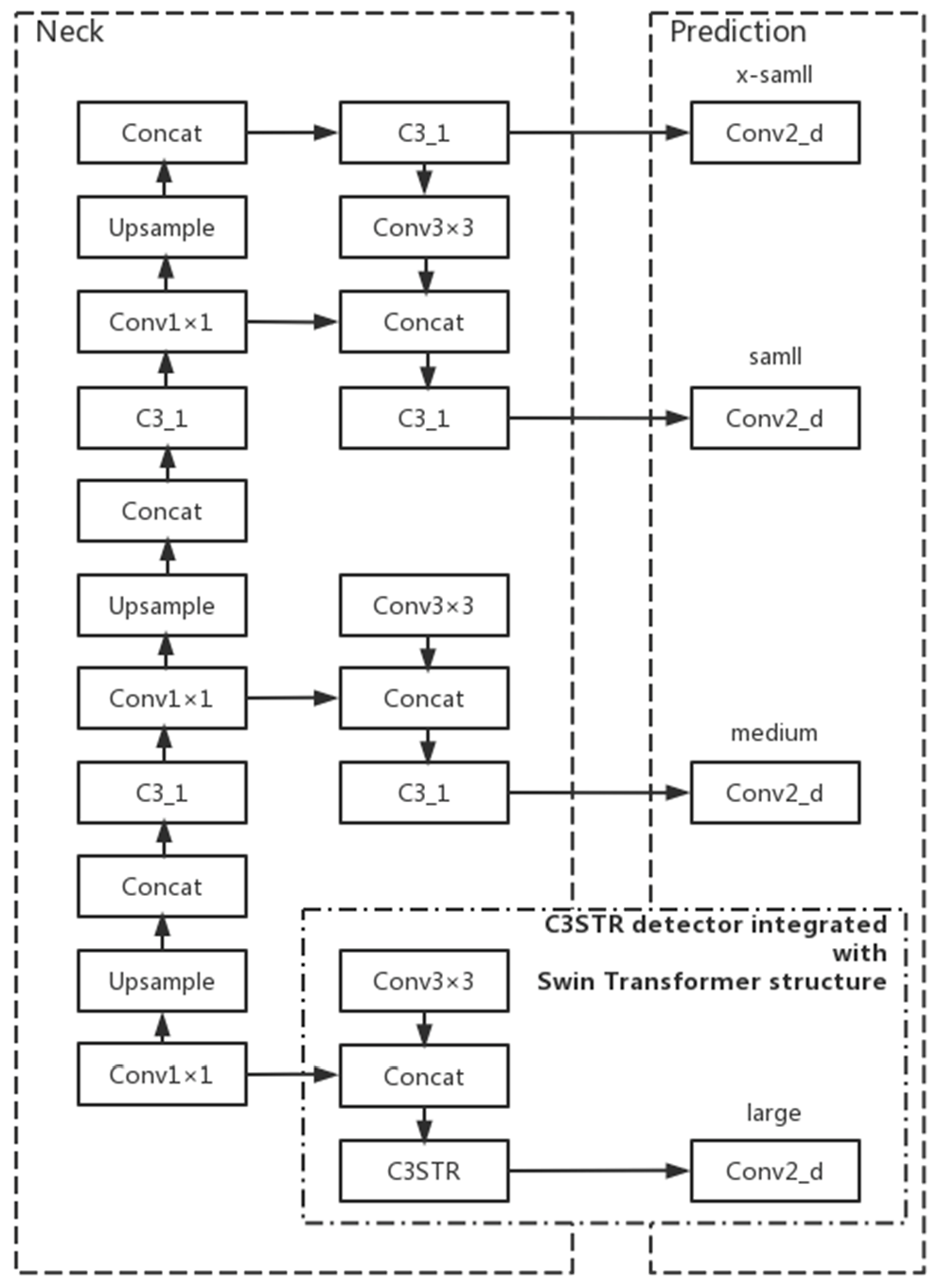
**Figure 7.** YOLOv5n’s integration into CA attention mechanism.

* + 1. Incorporating Swin Transformer Structure to Improve Model Generalization Performance In the detection of maize leaf spots, the distribution of different types of spots in

the leaf images differs: large spots occupy a small area of the leaf and rely more on local information of high-level features; rusts have a large distribution area and rely on global information more obviously; gray spots are moderate in size and rely on both local and global information. The performance of Convolutional Neural Networks (CNN) is more capable of capturing local information and has a certain disadvantage in global information acquisition. To alleviate the adverse effects of the non-uniform spot size, the model is improved by extracting global information using Swin Transformer [[23](#_bookmark37)]. In this paper, a smaller size target detection head was added to the original small, medium and large size detection heads of the YOLOv5n model to enhance its ability to identify small targets of the spots. The x-small size detection head in Figure [8](#_bookmark8), and the Swin Transformer structure, was incorporated into the large size detection head to replace the original C3 structure with the C3STR structure incorporated into the large size detection head to change the original C3 structure to C3STR structure, thus improving the model’s capture of feature information.The improved network structure is shown in Figure [8](#_bookmark8).

The Swin Transformer model was proposed by Microsoft Research in 2021. Swin Transformer uses hierarchical feature maps similar to those used in convolutional neural networks, such as feature map sizes with 4 , 8 , and 16 down-sampling of images, such that the backbone helps to build on top of this for tasks such as target detection, instance segmentation, etc. The Swin Transformer network is another collision of the Transformer model in the field of vision. The Swin Transformer network is another collision of Transformer model in vision field.

*× × ×*



**Figure 8.** Add x-small detection head and Swin Transformer structure.

The concept of Windows Multi-Head Self-Attention (W-MSA) is used in Swin Trans- former, for example, in the 4-fold downsampling and 8-fold downsampling in the figure below. The feature map is divided into multiple disjointed regions (Window), and Multi- Head Self-Attention is performed only within each window (Window).

The basic flow of the whole framework is as follows.

First, the image is input to the patch partition module for chunking, i.e., every 4 4 ad- jacent pixels is a patch, and then it is flattened in the channel direction. Assuming that the input is a three-channel RGB image, each patch has 4 4 = 16 pixels, and each pixel has three values, R, G, and B. The flattened image shape is 16 3 = 48, so the image shape changes from [H, W, 3] to [H/4, W/4, 48] after patch partition. Then the channel data of each pixel is linearly transformed by the linear embedding layer from 48 to C, i.e., the image shape is changed from [H/4, W/4, 48] to [H/4, W/4, C].

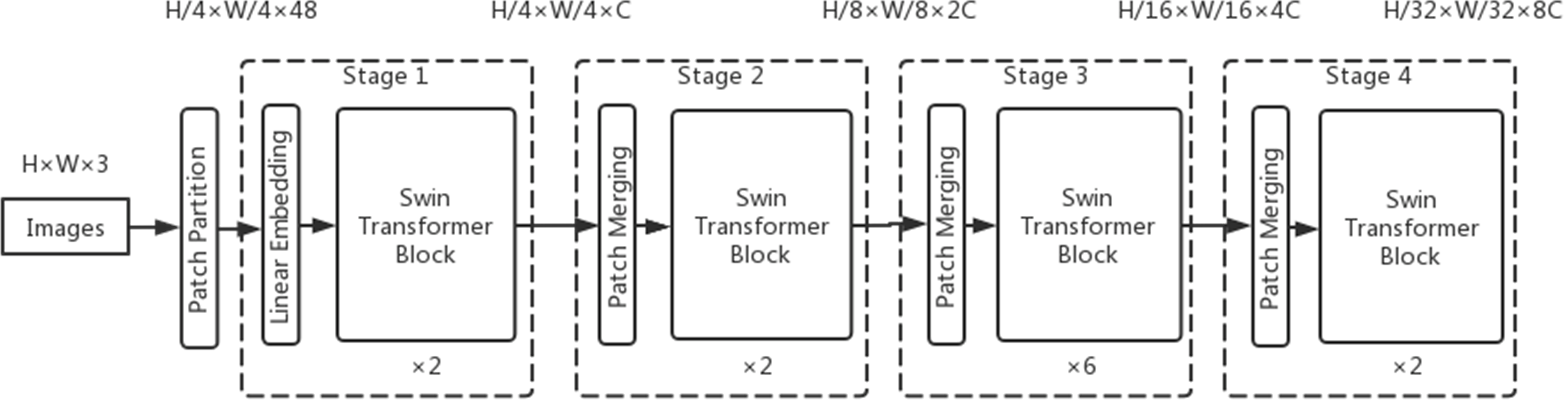
*×*

*×*

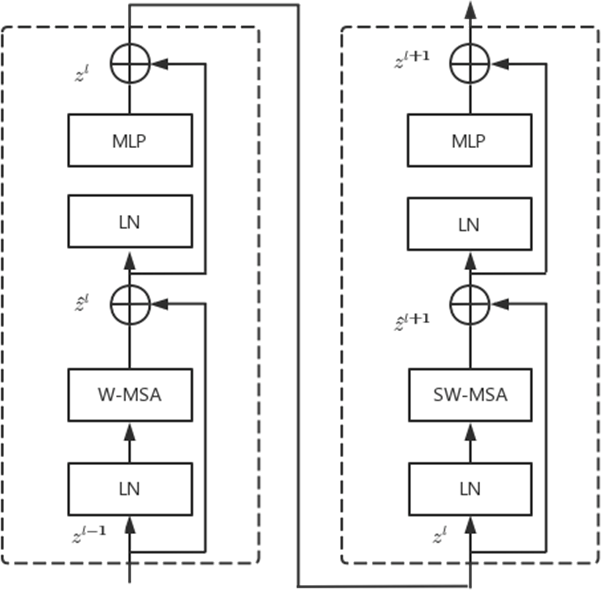
*×*

The Swin Transformer divides patches by first determining the size of each patch and then calculating the number of patches. The number of Swin Transformers decreases, and the perceptual range of each patch expands as the network depth deepens, which is designed to facilitate the construction of layers of Swin Transformer and to adapt to the multi-scale of visual tasks.The architecture of a Swin transformer (Swin-T) is shown in Figure [9](#_bookmark9).

Then the feature maps of different sizes are constructed by four stages; except for Stage 1, where a linear embedding layer is first passed, the remaining three stages are first downsampled by a patch merging layer. Note that there are two types of blocks, as shown in Figure [10](#_bookmark10), which differ only in that one uses the W-MSA structure and the other uses the SW-MSA structure. Moreover, these two structures are used in pairs, with one W-MSA structure used first and then one SW-MSA structure used.



**Figure 9.** The architecture of a Swin Transformer (Swin-T).



**Figure 10.** Two successive Swin Transformer blocks.

* 1. *Test Environment*

In this paper, we use the deep learning framework PyTorch to build and improve the model in the Anaconda3 environment, and train and test the model under the Windows 10 system. The computer CPU is 11th Gen Intel (R) Core (TM) i7–11700K @ 3.60 GHz and the GPU is NVIDIA GeForce GTX 1080 Ti. The GPU is used for acceleration to improve the network training speed, the Cuda version is 11.1.0, and the cudnn version is 8.1.0.

* 1. *Evaluation Metrics*

In this paper, the performance of the YOLOv5n network model is evaluated using several metrics from the target detection algorithm, specifically Precision (P), Recall (R), and mean Average Precision (mAP) [[24](#_bookmark38)]. Average Precision (AP) is the integral of the PR curve formed by taking Precision (P) as the vertical axis and Recall (R) as the horizontal axis. A recall is a metric that reflects the ability of the model to find positive sample targets, precision is used to reflect the ability of the model to classify samples, and average precision is a metric that reflects the overall performance of the model to detect and classify targets. The mean Average Precision (mAP) represents the average of the mean accuracy of all categories. Among all the metrics, mAP is the most important evaluation metric in the target detection algorithm, which can measure the accuracy of the detection algorithm. mAP0.5 is the AP of the target detection model evaluated at an IoU threshold of 0.5. mAP0.5 is its mean value for all categories; AP0.5–0.95 is the mean value of the AP of the model evaluated at different IoU thresholds (0.5–0.95, step size 0.05); and AP0.5–0.95 is the average value of AP evaluated under different IoU thresholds (0.5–0.95, step size 0.05), which is a more stringent model accuracy index. mAP0.5–0.95 is the average value of all categories. In this paper, we choose mAP0.5–0.95 as the evaluation index. The additional evaluation index considered in this paper is the number of parameters, and the number of parameters indicates the size of the storage space occupied by the model file in MB.

The expressions for the calculation of Precision (P), Recall (R), Average Precision (AP), and mean Average Precision (mAP) are shown in Equations (1)–(4).

P = TP

TP + FP

R = TP

TP + FN

1

∫

(1)

(2)

AP =

0

P*·*RdR (3)

mAP =

N

i=1

∑

APi

(4)

N

where:

* The number of true samples.
* The number of false positive samples.
* Number of pseudo-negative samples.
* The number of species in the sample.

The positive and negative samples are judged by setting the average Intersection over Union (IoU) threshold between the predicted target area and the actual target area, and if the IoU of both exceeds the threshold, the sample is positive, and if vice versa, the sample is negative.

# 4.Discussion

Crop disease identification plays a crucial role in the development of smart agriculture and digital agricultural information technology. However, current crop disease recognition methods suffer from low accuracy and poor targeting, which limits their practical applications for maize growers. Deep learning, as a powerful image recognition technology, has great potential in this field.

In this study, we proposed an improved CTR\_YOLOv5n model to detect and identify various maize leaf diseases, such as maize blotch disease, gray spot, and rust. Our model aims to provide technical reference for maize disease control and help farmers solve the problem of difficult maize disease identification. The main contributions of this study are as follows:

1. We utilized mosaic data augmentation to increase the number of small-scale targets, thereby improving the model's ability to recognize small-sized disease spots.
2. To address the challenge of distinguishing between different maize leaf diseases with similar color and shape, we incorporated the CA attention mechanism into the YOLOv5n model backbone network, added a minimum size detection head to detect small targets, and introduced a Swin Transformer structure in the large size detection head to fuse global and local information, thereby improving the model's generalization performance and recognition accuracy by 2.8%.
3. Using the improved CTR\_YOLOv5n model, we achieved an average recognition accuracy of 95.2%, which is 2.8% higher than the original model. Furthermore, our model size was maintained at 5.1MB with 94.5% less size compared to the YOLOv5l model. This indicates that our model is not only accurate but also more lightweight and suitable for real-time detection on mobile devices, improving the detection efficiency.

Our results demonstrate the effectiveness of the improved CTR\_YOLOv5n model in maize leaf disease recognition. However, further research is needed to evaluate the model's performance under complex backgrounds and occluded cases. Additionally, our study focused only on identifying diseases on the leaves of maize, but it can be extended to the identification of diseases on the roots and ears of maize, such as maize stalk rot, maize silky black ears, and maize rot, which have a high incidence rate and can cause significant losses.

# 5.Conclusion

This study proposes a method to enhance the YOLOv5n model's recognition and detection of maize leaf spots. We incorporate a CA attention module into the model's backbone network, which increases the importance of feature information related to maize leaf spots. Additionally, we add a smaller size target detection head to improve the model's ability to detect small targets of spots, and we integrate the Swin Transformer structure into the larger detection head to enhance the model's feature information capture ability. These modifications effectively increase the model's accuracy for maize leaf spot recognition. Experimental results demonstrate that our proposed method achieves an average recognition accuracy of 95.2% for maize leaf spots, and the recognition accuracy for maize gray spot and maize large spot diseases, which are difficult to recognize, are improved to 92.8% and 92.5%, respectively. Compared to other deep learning network models, our method offers higher accuracy and a smaller size. Moving forward, we plan to collect images of maize leaf diseases in natural environments and refine the model further to develop a crop disease recognition system that can be applied to mobile devices.

References

1. Cao, T.; Zhang, X.; Chen, X.; Peng, X.; Lin, J. Maize Disease Classification Method Based on Spatial Attention Mechanism and DenseNet. *Radio Eng.* **2022**, *52*, 1710–1717.
2. Fan, X.; Zhou, J.; Xu, Y.; Peng, X. Maize Disease Recognition under Complicated Background Based on Improved Convolutional Neural Network. *Trans. Chin. Soc. Agric. Mach.* **2021**, *52*, 210–217.
3. Bao, W.; Huang, X.; Hu, G.; Liang, D. Identification of maize leaf diseases using improved convolutional neural network. *Trans.* *Chin. Soc. Agric. Eng.* **2021**, *37*, 160–167.
4. Wang, Y.; Wu, J.; Lan, P.; Li, F.; Ge, C.; Sun, F. Apple disease identification using improved Faster R-CNN. *J. For. Eng.* **2022**, *7*, 153–159. [[CrossRef](http://doi.org/10.13360/j.issn.2096-1359.202104028)]
5. Huang, Y.; Ai, X. Research on Classification of Corn Leaf Disease Image by Improved Residual Network. *Comput. Eng. Appl.*

**2021**, *57*, 178–184. [[CrossRef](http://doi.org/10.3778/j.issn.1002-8331.2105-0321)]

1. Wu, Y. Identification of Maize Leaf Diseases Based on Convolutional Neural Network. *J. Phys. Conf. Ser.* **2021**, *1748*, 032004. [[CrossRef](http://doi.org/10.1088/1742-6596/1748/3/032004)]
2. Wang, C.; Wang, C.; Liu, J. Identification of Maize Leaf Diseases based on Deep Learning. *Mod. Agric. Res.* **2022**, *28*, 102–106. [[CrossRef](http://doi.org/10.19704/j.cnki.xdnyyj.2022.06.020)]
3. Azlah, M.A.F.; Chua, L.S.; Rahmad, F.R.; Abdullah, F.I.; Wan Alwi, S.R. Review on Techniques for Plant Leaf Classification and Recognition. *Computers* **2019**, *8*, 77. [[CrossRef](http://doi.org/10.3390/computers8040077)]
4. Koklu, M.; Unlersen, M.F.; Ozkan, I.A.; Aslan, M.F.; Sabanci, K. A CNN-SVM Study Based on Selected Deep Features for Grapevine Leaves Classification. *Measurement* **2022**, *188*, 110425. [[CrossRef](http://doi.org/10.1016/j.measurement.2021.110425)]
5. Argüeso, D.; Picon, A.; Irusta, U.; Medela, A.; San-Emeterio, M.G.; Bereciartua, A.; Alvarez-Gila, A. Few-Shot Learning Approach for Plant Disease Classification Using Images Taken in the Field. *Comput. Electron. Agric.* **2020**, *175*, 105542. [[CrossRef](http://doi.org/10.1016/j.compag.2020.105542)]
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–15 December 2015; Curran Associates, Inc.: Red Hook, NY, USA, 2015; Volume 28.
7. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision, ECCV, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
9. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
10. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
11. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
12. Chen, L.; Fu, D. Survey on Machine Learning Methods for Small Sample Data. *Comput. Eng.* **2022**, *48*, 1–13. [[CrossRef](http://doi.org/10.19678/j.issn.1000-3428.0065347)]
13. Chen, C.; Fan, Y.; Wang, L. Logo Detection Based on Improved Mosaic Data Enhancement and Feature Fusion. *Comput. Meas.* *Control.* **2022**, *30*, 188–201. [[CrossRef](http://doi.org/10.16526/j.cnki.11-4762/tp.2022.10.029)]
14. Tian, M.; Liao, Z. Research on Flower Image Classification Method Based on YOLOv5. *J. Phys. Conf. Ser.* **2021**, *2024*, 012022. [[CrossRef](http://doi.org/10.1088/1742-6596/2024/1/012022)]
15. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. *arXiv* **2018**, 8759–8768.
16. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. *arXiv* **2021**, 13713–13722.
17. Li, S.; Li, K.; Qiao, Y.; Zhang, L. A Multi-Scale Cucumber Disease Detection Method in Natural Scenes Based on YOLOv5. *Comput.* *Electron. Agric.* **2022**, *202*, 107363. [[CrossRef](http://doi.org/10.1016/j.compag.2022.107363)]
18. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. *arXiv* **2021**, arXiv:2103.14030.
19. Shang, Y.; Zhang, Q.; Song, H. Application of deep learning based on YOLOv5s to apple flower detection in natural scenes. *Trans. Chin. Soc. Agric. Eng.* **2022**, *38*, 222–229. [[CrossRef](http://doi.org/10.11975/j.issn.1002-6819.2022.09.024)]

Footnotes

1[Add footnotes, if any, on their own page following references. For APA formatting requirements, it’s easy to just type your own footnote references and notes. To format a footnote reference, select the number and then, on the Home tab, in the Styles gallery, click Footnote Reference. The body of a footnote, such as this example, uses the Normal text style. (Note: If you delete this sample footnote, don’t forget to delete its in-text reference as well. That’s at the end of the sample Heading 2 paragraph on the first page of body content in this template.)]

Tables

Table 1

[Table Title]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Head | Column Head | Column Head | Column Head | Column Head |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |
| Row Head | 123 | 123 | 123 | 123 |
| Row Head | 456 | 456 | 456 | 456 |
| Row Head | 789 | 789 | 789 | 789 |

Note: [Place all tables for your paper in a tables section, following references (and, if applicable, footnotes). Start a new page for each table, include a table number and table title for each, as shown on this page. All explanatory text appears in a table note that follows the table, such as this one. Use the Table/Figure style, available on the Home tab, in the Styles gallery, to get the spacing between table and note. Tables in APA format can use single or 1.5 line spacing. Include a heading for every row and column, even if the content seems obvious. A default table style has been setup for this template that fits APA guidelines. To insert a table, on the Insert tab, click Table.]

Figures title:

Figure 1. [Include all figures in their own section, following references (and footnotes and tables, if applicable). Include a numbered caption for each figure. Use the Table/Figure style for easy spacing between figure and caption.]

For more information about all elements of APA formatting, please consult the APA Style Manual, 6th Edition.