Weed Detection in Maize Fields by UAV Images Based on Crop Row Preprocessing and Improved YOLOv4

Haotian Pei 1,2, Youqiang Sun 2, He Huang 2,*, Wei Zhang 1,2, Jiajia Sheng 2,3 and Zhiying Zhang 4

1 Institutes of Physical Science and Information Technology, Anhui University, Hefei 230601, China; htpei@stu.ahu.edu.cn (H.P.); wzhang@stu.ahu.edu.cn (W.Z.)
2 Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China; yqsun@iim.ac.cn (Y.S.); joy0413@mail.ustc.edu.cn (J.S.)
3 Science Island Branch of Graduate School, University of Science and Technology of China, Hefei 230026, China
4 Institute of Science and Technology Information, Jiangsu University, Zhenjiang 212013, China; zyzhang@stmail.ujs.edu.cn
* Correspondence: hhuang@iim.ac.cn

Abstract: Effective maize and weed detection plays an important role in farmland management, which helps to improve yield and save herbicide resources. Due to their convenience and high resolution, Unmanned Aerial Vehicles (UAVs) are widely used in weed detection. However, there are some challenging problems in weed detection: (i) the cost of labeling is high, the image contains many plants, and annotation of the image is time-consuming and labor-intensive; (ii) the number of maize is much larger than the number of weed in the field, and this imbalance of samples leads to decreased recognition accuracy; and (iii) maize and weed have similar colors, textures, and shapes, which are difficult to identify when an UAV flies at a comparatively high altitude. To solve these problems, we propose a new weed detection framework in this paper. First, to balance the samples and reduce the cost of labeling, a lightweight model YOLOv4-Tiny was exploited to detect and mask the maize rows so that it was only necessary to label weeds on the masked image. Second, the improved YOLOv4 was used as a weed detection model. We introduced the Meta-ACON activation function, added the Convolutional Block Attention Module (CBAM), and replaced the Non-Maximum Suppression (NMS) with Soft Non-Maximum Suppression (Soft-NMS). Moreover, the distributions and counts of weeds were analyzed, which was useful for variable herbicide spraying. The results showed that the total number of labels for 1000 images decrease by half, from 33,572 to 17,126. The improved YOLOv4 had a mean average precision (mAP) of 86.89%.

Keywords: crop row mask; weed detection; samples balance; YOLOv4; Meta-ACON; attention module; Soft-NMS; UAV images

1. Introduction

Maize plays an important role in agriculture because of its nutritional value and high consumption [1]. Excessive weeds in maize fields affect the growth and yield of maize. Weeds not only compete with crops for living space but also lead to the spread of diseases and pests, resulting in crop failure [2–4]. Therefore, we need to identify weeds in maize fields and ascertain their distribution and quantity. At present, the main method of weeding is large-scale spraying with herbicides, which destroys the ecological environment, consumes resources, and affects food safety [5]. The application of UAV technology in agriculture has developed rapidly [6–8] as it provides unique features that humankind cannot implement on the ground [9], thus facilitating the detection of weeds in maize fields. UAVs can obtain complete field images with high resolution [10,11]. The advantage of UAV images is that they have a wide collection area and can quickly obtain complete field data for statistics and analysis [12]. The analysis of farmland images and the implementation of precise spraying are of great significance for controlling the growth of weeds and
improving grain yield [13]. To implement precise spraying, the first problem is to identify the distribution of crops and weeds in any given area of farmland. Many experts have conducted research in the field of weed detection and achieved corresponding results.

Radhika Kamath et al. [14] looked at how texture features extracted from Laws’ texture masks can be used to distinguish between carrot crops and weeds in digital photographs. Based on the analysis of photos obtained by UAVs, Lottes, P et al. [15] suggested a system that performs vegetation detection, plant-tailored feature extraction, and classification, consequently estimating the distribution of crops and weeds in fields. Using UAV photos gathered from a chili crop field in Australia, Nahina Islam et al. [16] assessed the performance of numerous machine learning methods, including random forest, Support Vector Machine (SVM), and k-nearest neighbor, and detected weeds. Although traditional machine learning methods have certain effects on weed recognition, they still exhibit defects, such as difficult feature extraction and poor anti-interference ability. Since the development of deep learning, convolutional neural networks (CNNs) have been widely used in agriculture. Compared with traditional machine learning methods, CNNs have strong feature extraction and autonomous learning abilities, thereby achieving outstanding results and making them more useful in the detection and localization of weeds [17]. ResNeXt-SVM is a hybrid model of deep learning for weed detection proposed by Jabir Brahim et al. [18]. They created the ResNeXt-SVM framework by combining a ResNeXt and SVM network to improve the exploitation of structured characteristics in images and the understanding of their content. KunHu et al. [19] introduced the Graph Weeds Net (GWN), a unique graph-based deep learning architecture aimed at recognizing numerous species of weed from standard RGB photos taken from complicated rangelands. We experimented and tested our data with different object detection algorithms. The single-stage algorithm included YOLOv3 [20], YOLOv4 [21], and SSD [22], and the two-stage algorithm used Faster R-CNN [23]. We found that YOLOv4 achieved higher accuracy, so we chose YOLOv4 as the baseline and optimized it for the weed detection model.

However, there are still some problems in the existing research: (i) the cost of labeling crops is high (an image contains many maize plants that need to be labeled), and annotation of the image is time-consuming and labor-intensive, (ii) the sample size is seriously unbalanced (more weed samples are needed, but labeled crop samples vastly outnumber the weed samples, which can lead to decreased recognition accuracy), and (iii) maize and weed have similar colors, textures, and shapes, which renders them difficult to identify when a UAV flies at a comparatively high altitude.

To solve the above problems, we designed a new weed detection framework. The framework of this paper is shown in Figure 1. In the first stage, it was necessary to detect crop rows. The traditional method of crop row detection is semantic segmentation. Semantic segmentation requires a pixel-level label, which cannot meet the requirement of simpler labeled content. Therefore, an object detection method was adopted to achieve it. We labeled a small number of maize row datasets. A lightweight model YOLOv4-Tiny was exploited to detect and mask the maize rows so that we only needed to label weeds on the masked image. Labeling the masked image in this way was convenient, and it also reduced the cost of labeling. In the second stage, we generated datasets by selecting images from the masked images to label the weeds, and we selected images from the original images to label the maize and weeds. Then, we considered the features of the detected image to optimize the model. We introduced the Meta−ACON [24] activation function, which can improve generalization and transmission performance. We added the CBAM [25] to allow our model to adaptively focus on more important areas. Due to the high height of the captured image, it often happens that maize or weeds are adjacent, so we used Soft-NMS [26] to avoid adjacent bounding boxes being removed. Finally, the optimized YOLOv4 model was used for weed detection.

The main contributions of this paper are as follows: (i) a new weed detection framework is proposed to deal with the high cost of labeling and the serious imbalance of samples, (ii) an optimized weed detection model is proposed, which has a good recognition accuracy,
and (iii) the analysis of high-resolution images of farmland collected by the UAV, which can calculate the quantity, distribution, and density of maize and weeds in large areas.

Figure 1. Framework used in this paper.

2. Materials and Methods

2.1. Data Collection

We used a DJI Phantom 4 RTK for image collection in the experimental farm of the Yellow River Delta Agricultural Hi-Tech Industry Demonstration Zone of Dongying City, Shangdong Province, China. A Phantom 4 RTK camera uses a 1 inch CMOS sensor to capture 20 mega-pixel imagery with a resolution of 4864 * 3648 pixels. During flight, a 3-axis gimbal on a Phantom 4 RTK provides a steady platform to keep the attached camera pointed close to the nadir. The capture area is 40 mu. The image was captured in the seedling stage (V1), and the specific image acquisition time was May 2021. We planned the flight route of the UAV according to the image resolution and its flight efficiency. We set the flying altitude to 25 m and the Ground Sampling Distance (GSD) to 0.685 cm/pixel. Finally, the visible photos were synthesized into a farmland image through two-dimensional reconstruction using pix4D [27] software.
The image collected by the UAV is shown in Figure 2. We cropped the edge of the non-farmland area to obtain an image of the whole farmland, as shown in Figure 3a. Finally, we cropped the image to a suitable resolution for the identification of maize and weed. The image was cropped to 416 * 416 pixels, as shown in Figure 3b.

Figure 2. Image collected by the UAV in the Yellow River Delta Agricultural Hi-Tech Industry Demonstration Zone Dongying City, of Shangdong Province, China.

Figure 3. Steps of image processing: (a) remove the edge, and (b) image cropping.
2.2. YOLOv4 and YOLOv4-Tiny

Convolutional Neural Networks have been widely employed in object detection and have shown promising results [28]. YOLO is a one-stage object detection network that turns object detection into a regression problem, as is typical of a standard object detection technique. By processing photos with a single CNN, YOLO can directly extract the target's categories and position coordinates. YOLO has the advantages of fast detection speed, a small model, and convenient deployment. The increase in efficiency and accuracy of YOLOv4 compared with YOLOv3 arise mainly from several improvements incorporated into the model: (i) the backbone extraction network is improved from Darknet53 to CSPDarknet53; (ii) the spatial pyramid pooling (SPP) module is introduced to significantly increase the receptive field, (iii) the use of the Path Aggregation Network (PANet) as the parameter aggregation method, and (iv) the Mosaic data augmentation and Mish activation functions are used to further improve the accuracy.

YOLOv4-Tiny is a simplified version of YOLOv4 that greatly improves training and detection speeds by reducing the amount of network calculation [29]. YOLOv4-Tiny omits spatial pyramid pooling (SPP), and the Path Aggregation Network (PANet) uses the CSPDarknet53-Tiny network as the backbone network to replace the CSPDarknet53 network used in YOLOv4 [30]. Regarding feature fusion, YOLOv4-Tiny uses the feature pyramid network to extract and fuse features of different scales, which improves the accuracy of object detection. In addition, other lightweight versions of YOLOv4 include YOLOv4-Mobilenetv1 [31], YOLOv4-Mobilenetv3 [32], and YOLOv4-Ghost [33]. In YOLOv4-Mobilenetv1 and YOLOv4-Mobilenetv3, the original backbone CSPDarknet53 was replaced by Mobilenetv1 and Mobilenetv3. In YOLOv4-Ghost, the core concept is to use cheap operations to generate redundant feature maps.

2.3. Crop Row Detection and Mask

Accounting for the important fact that maize is grown in the crop row, we covered the maize by masking the crop row to obtain the image only with inter-row weeds. Then, we selected an appropriate number of original images and masked images and labeled them using LabelImg [34], which skillfully solves the problem of imbalanced samples. Moreover, by reducing the annotation of maize with low demand for sample size, a lot of time was saved. Therefore, greater significance is achieved if the training time of our crop row detection model satisfies Equation (1).

\[ T_{\text{train}} + T_M + T_N < T_{M+N} \]  

(1)

where \( T_{\text{train}} \) is the time of training crop row detection model, \( T_M \) is the time of labeling M original images, \( T_N \) is the time of labeling N masked images, and \( T_{M+N} \) is the time of labeling M and N original images.

This meant that it was necessary to obtain the crop row data set and train the crop row detection model faster. Therefore, the requirements for the crop row detection model were as follows: (i) fewer labeled samples, (ii) simpler labeled content, (iii) shorter training time, and (iv) a lightweight model. The traditional method of crop row detection is semantic segmentation. Semantic segmentation requires pixel-level label, which cannot meet the requirement of simpler labeled content. Therefore, an object detection method was adopted to achieve this.

2.3.1. Crop Row Detection Model Dataset

To meet the requirement of fewer labeled samples, only 150 images were randomly taken for labeling. We used the LabelImg tool to label crop rows, as shown in Figure 4. There are only 2–4 labels in any given image, and it took about 30 min to label 150 images. Finally, we augmented the dataset to 750 images.
2.3.2. Crop Row Detection Model

Since we needed to quickly obtain the best model to save time when outputting masked images, the model training time was a key factor when choosing the model. We experimented with several lightweight models of YOLOv4, such as YOLOv4-Tiny, YOLOv4-Ghost, YOLOv4-Mobilenetv1, and YOLOv4-Mobilenetv3. Considering the model performance and training time, YOLOv4-Tiny is an ideal model for crop row detection.

Finally, according to the detection results from the crop row detection model, we located the crop row coordinates, drew the bounding box, and masked the bounding box so that we obtained images only with inter-row weeds. The outputted masked image is shown in Figure 5.

2.4. Weed Detection Model Dataset

A total of 1000 images were selected for annotation. We randomly selected 300 images from the original images to label the maize and weeds. Furthermore, we randomly selected 700 images from the masked images to label the weeds. We labeled the original image and the masked image according to the ratio of 3:7, which greatly reduced the maize labels with similar features and helped to achieve sample balance. In total, 7700 maize labels and 9426 weed labels were labeled using this method. We estimated the number of labels for 1000 original images based on the number of maize and weed labels for 300 original images. It was estimated that 25,690 maize labels and 7882 weed labels were to be labeled.

![Figure 4. Labeling method of crop row dataset.](image_url)

![Figure 5. (a) The image inputted into the crop row detection model, and (b) the masked image outputted by the crop row detection model.](image_url)
The total number of labels for 1000 images decreased by half, from 33,572 to 17,126, and all the reduced labels were maize labels with low demand for sample size. The proposed method not only saves a lot of labeling time but also achieves sample balance.

Before the network training, data augmentation was performed on the dataset, including brightness enhancement and reduction, contrast enhancement and reduction, Gaussian noise addition, vertically flipping 30% of all images, flipping 30% of all images with the mirror, translation, rotation, and zooming, the purpose of which was to enrich the image training set, effectively extract the image features, and avoid overfitting. After data augmentation, 3000 photos yielded a total of 23,100 maize labels and 28,278 weed labels, which were used for network training and parameter adjustment.

The method of labeling the original image and masked image is shown in Figure 6a,b. Data augmentation of the original image and masked image are shown in Figure 6c,d. We randomly selected 70% of images as the training set, 10% of images as the validation set, and 20% of images as the test set.

![Figure 6. Examples of different forms of sample labeling, where the green label box is maize and the red label box is weed. (a) The original image was labeled with maize and weed; (b) the masked image was labeled with weed; (c) data augmentation of the original image, and (d) data augmentation of the masked image.](image-url)
2.5. Weed Detection Model

In this paper, we optimized the YOLOv4 model. The structure of YOLOv4 is composed of three parts: (i) backbone, (ii) neck, and (iii) YOLO head. First, a new activation function, Meta-ACON, was introduced into our model, which can adaptively determine the upper and lower bounds and dynamically control the degree of linearity and nonlinearity. Second, we also added the CBAM and optimized it by changing the activation function of the channel attention module to Meta-ACON. Finally, the traditional NMS [35] function was replaced with the Soft-NMS function. Thus, the optimized model can improve the recognition accuracy and prove the applicability of the Meta-ACON activation function, optimized CBAM, and Soft-NMS in the weed detection model. In addition, we drew a distribution map of maize and weed in the whole farmland. The numbers of maize and weed were calculated to provide a basis for variable-rate spraying. The structure of improved YOLOv4 is shown in Figure 7.

![Figure 7. Improved YOLOv4 network structure diagram.](image)

2.5.1. CBA Module

We introduced the Meta-ACON activation function into our model, and the CBL module (conv, BN, and Leaky ReLU) in the neck was changed to the CBA module (conv, BN, and Meta-ACON), which improves the performance of the model. Meta-ACON is a new activation function. It is obtained by further analysis and extending the maxout form of ReLU with smooth maximum, which can adaptively determine the upper and lower bounds of the first derivative of the function and dynamically control the degree of linearity and nonlinearity of the network. This customized activation behavior helps to improve generalization and transmission performance. The calculation formula of Meta-ACON is shown in Appendix A(a).

2.5.2. CBAM

The attention module can effectively eliminate interference factors, allowing the network to focus on areas that need more attention. We added the CBAM to the model and introduced the Meta-ACON activation function in the channel attention part. We changed its activation function from ReLU to Meta-ACON to improve the detection effect. To
facilitate the comparison between our optimized attention module and the original CBAM, we named the attention module with the attached Meta-ACON A_CBAM. A_CBAM, as a lightweight convolution block attention module, infers the attention map through two independent dimensions (channel and space), and then multiplies the attention map by the input feature map for adaptive feature refinement, improving the performance of the model through end-to-end training. The process and formulas of CBAM are shown in Appendix A(b).

2.5.3. Soft-NMS

The traditional NMS sorts all bounding boxes according to their scores. It selects the bounding box \( M \) with the highest score and suppresses all other bounding boxes \( b_i \) that significantly overlap with \( M \). The problem with the NMS is that it sets the score of the adjacent bounding box to zero, so if an object exists in the overlap threshold, it can be missed. In farmland, it often happens that maize or weeds are adjacent, and the bounding box of maize or weeds with lower scores can be missed. Soft-NMS considers both the score and degree of overlap when performing non-maximum suppression. This is achieved by decaying the detection score of all other objects to a continuous function overlapped with \( M \), rather than directly setting the score to zero, as in NMS. Therefore, no objects are eliminated in this process. The comparison between Soft-NMS and NMS is shown in Figure 8. The calculation formula of Soft-NMS is shown in Appendix A(c).

**Figure 8.** Comparison of the detection results between NMS and Soft-NMS. (a1) NMS cannot detect overlapping maize; (a2) Soft-NMS can detect overlapping maize; (b1) NMS cannot detect overlapping weed; and (b2) Soft-NMS can detect overlapping weed.
2.6. Methods Evaluation Indicator

Precision and recall should both be addressed while developing a detection model, so measures such as precision, recall, F1-score, AP, and mAP were utilized to test the model’s performance and evaluate the detection outcomes in this study. The precision, recall and F1-score can be calculated by Formulas (2)–(4).

\[ P_r = \frac{TP}{TP + FP} \quad (2) \]
\[ R_e = \frac{TP}{TP + FN} \quad (3) \]
\[ F1 = \frac{2 \times P_r \times R_e}{P_r + R_e} \quad (4) \]

where \( P_r \) represents precision, \( R_e \) represents recall, and \( F1 \) represents F1-score. \( TP \) (true positive) represents the number of positive samples correctly classified, and \( TN \) (true negative) represents the number of negative samples correctly classified. \( FP \) (false positive) indicates the number of negative samples incorrectly classified as positive samples, and \( FN \) (false negative) indicates the number of positive samples incorrectly classified as negative samples.

PR curve can be drawn by taking different precision and recall values. The area of PR curve is defined as \( AP \), and the mean value of \( AP \) of all detection categories is \( mAP \). \( AP \) and \( mAP \) can be calculated by Formulas (5) and (6).

\[ AP = \int_{0}^{1} p(r)dr \quad (5) \]
\[ mAP = \frac{1}{n} \sum_{i=1}^{n} AP_i \quad (6) \]

where \( p(r) \) represents the function of PR curve, and \( AP_i \) represents the \( AP \) value of each category.

3. Results and Discussion

The tests were carried out on a system that had an Intel Xeon Gold 5220 CPU and an NVIDIA Tesla V100 GPU. CUDNN 7.6.5 and CUDA 10.2 were the accelerated environments. The crop row detection network and weed detection network were trained and tested using the Ubuntu 18.04 operating system, with Python 3.7.0 as the programming language.

3.1. Crop Row Detection Model Experiment

We experimented with several lightweight models of YOLOv4, such as YOLOv4-Ghost, YOLOv4-Mobilenetv1, and YOLOv4-Mobilenetv3. We conducted nine experiments for each model. Under the condition that other parameters are the same, the batch size was set to 4, 8, and 16, and the epoch was set to 50, 100, and 200, respectively. We compared the training time of each model on the same device. The evaluation indicators of each model and the training time under the same conditions are shown in Tables 1 and 2.

<table>
<thead>
<tr>
<th>Model</th>
<th>AP</th>
<th>Precision</th>
<th>Recall</th>
<th>FPS</th>
<th>Weight</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv4-Mobilenet v1</td>
<td>90.19%</td>
<td>98.03%</td>
<td>89.52%</td>
<td>20.78</td>
<td>53.6 MB</td>
<td>48.42 MB</td>
</tr>
<tr>
<td>YOLOv4-Mobilenet v3</td>
<td>92.81%</td>
<td>98.12%</td>
<td>89.16%</td>
<td>18.91</td>
<td>56.4 MB</td>
<td>48.42 MB</td>
</tr>
<tr>
<td>YOLOv4-Ghost</td>
<td>92.56%</td>
<td>98.59%</td>
<td>90.28%</td>
<td>16.04</td>
<td>44.5 MB</td>
<td>48.42 MB</td>
</tr>
<tr>
<td>YOLOv4</td>
<td>93.15%</td>
<td>98.06%</td>
<td>90.77%</td>
<td>18.53</td>
<td>256.3 MB</td>
<td>245.3 MB</td>
</tr>
<tr>
<td>YOLOv4-Tiny</td>
<td>92.97%</td>
<td>99.28%</td>
<td>89.74%</td>
<td>42.24</td>
<td>23.6 MB</td>
<td>23.10 MB</td>
</tr>
</tbody>
</table>
Table 2. Comparison of the training time of different models.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>YOLOv4-Mobilenet v1</th>
<th>YOLOv4-Mobilenet v3</th>
<th>YOLOv4-Ghost</th>
<th>YOLOv4</th>
<th>YOLOv4-Tiny</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch size = 4</td>
<td>39 m 48 s</td>
<td>49 m 48 s</td>
<td>56 m 48 s</td>
<td>92 m 08 s</td>
<td>10 m 45 s</td>
</tr>
<tr>
<td>Epochs = 50</td>
<td>78 m 37 s</td>
<td>102 m 31 s</td>
<td>113 m 26 s</td>
<td>181 m 19 s</td>
<td>21 m 29 s</td>
</tr>
<tr>
<td>Batch size = 4</td>
<td>154 m 47 s</td>
<td>204 m 48 s</td>
<td>215 m 39 s</td>
<td>349 m 48 s</td>
<td>42 m 23 s</td>
</tr>
<tr>
<td>Epochs = 200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Batch size = 8</td>
<td>11 m 30 s</td>
<td>13 m 17 s</td>
<td>14 m 45 s</td>
<td>19 m 43 s</td>
<td>8 m 36 s</td>
</tr>
<tr>
<td>Epochs = 50</td>
<td>22 m 31 s</td>
<td>25 m 28 s</td>
<td>29 m 30 s</td>
<td>38 m 35 s</td>
<td>17 m 08 s</td>
</tr>
<tr>
<td>Batch size = 8</td>
<td>48 m 29 s</td>
<td>48 m 09 s</td>
<td>48 m 14 s</td>
<td>75 m 57 s</td>
<td>33 m 05 s</td>
</tr>
<tr>
<td>Epochs = 100</td>
<td>11 m 05 s</td>
<td>11 m 30 s</td>
<td>12 m 08 s</td>
<td>17 m 10 s</td>
<td>8 m 18 s</td>
</tr>
<tr>
<td>Batch size = 16</td>
<td>22 m 13 s</td>
<td>22 m 30 s</td>
<td>24 m 01 s</td>
<td>34 m 15 s</td>
<td>16 m 33 s</td>
</tr>
<tr>
<td>Epochs = 50</td>
<td>48 m 30 s</td>
<td>47 m 45 s</td>
<td>47 m 20 s</td>
<td>67 m 33 s</td>
<td>32 m 31 s</td>
</tr>
<tr>
<td>Epochs = 200</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It can be seen that each model achieved good results. YOLOv4 achieved the highest AP (93.15%) and the highest recall (90.77%), but the parameter size and training time of YOLOv4 were much larger than other lightweight models. YOLOv4-Tiny achieved the second highest AP (92.97%) and the highest precision (99.28%), with the lowest parameter value and the fastest detection speed. According to Table 2, YOLOv4-Tiny, which can obtain the best model at the fastest speed, had the shortest training time. Combining the results of Tables 1 and 2, it can be concluded that YOLOv4-Tiny is the ideal model.

3.2. Weed Detection Model Ablation Experiment

To prove that the CBA module can effectively improve the performance of the model, we compared the evaluation indicators with the original YOLOv4. The comparison is shown in Table 3.

Table 3. CBA module ablation experiment.

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP</th>
<th>AP Maize</th>
<th>AP Weed</th>
<th>Recall Maize</th>
<th>Recall Weed</th>
<th>Precision Maize</th>
<th>Precision Weed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original YOLO-v4</td>
<td>84.96%</td>
<td>85.97%</td>
<td>83.94%</td>
<td>81.49%</td>
<td>73.80%</td>
<td>92.60%</td>
<td>92.71%</td>
</tr>
<tr>
<td>YOLOv4+CBA</td>
<td>85.75%</td>
<td>86.67%</td>
<td>84.83%</td>
<td>83.25%</td>
<td>77.95%</td>
<td>92.51%</td>
<td>92.69%</td>
</tr>
</tbody>
</table>

From the above table, we can see that the mAP of the improved model increased by 0.79% compared with the original YOLOv4. The AP of maize increased by 0.7%, reaching 86.67%, and the AP of weed increased by 0.89%, reaching 84.83%. Its recall greatly improved, especially that of weed (4.15%). Since Meta-ACON can dynamically control the degree of linearity and nonlinearity of the activation function, the performance significantly improved.

We added the CBAM to the model and introduced the Meta—ACON activation function in the channel attention part. As the attention module is a plug-and-play module, we studied the effect of adding the attention module at different positions. We conducted three ablation studies on CBAM and A_CBAM to evaluate the benefits of adding them after the three effective feature layers were extracted from the backbone network and after upsampling. The comparison is shown in Table 4.
Table 4. Attention module ablation experiment.

<table>
<thead>
<tr>
<th>Model</th>
<th>Position</th>
<th>mAP</th>
<th>AP</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Maize</td>
<td>Weed</td>
<td>Maize</td>
</tr>
<tr>
<td>Original YOLOv4</td>
<td></td>
<td>84.96%</td>
<td>85.97%</td>
<td>83.94%</td>
<td>81.49%</td>
</tr>
<tr>
<td>CBAM effective feature layers</td>
<td></td>
<td>85.10%</td>
<td>86.13%</td>
<td>84.07%</td>
<td>81.53%</td>
</tr>
<tr>
<td>A_CBAM</td>
<td></td>
<td>85.31%</td>
<td>86.18%</td>
<td>84.45%</td>
<td>82.42%</td>
</tr>
<tr>
<td>CBAM upsampling</td>
<td></td>
<td>85.32%</td>
<td>86.52%</td>
<td>84.12%</td>
<td>81.87%</td>
</tr>
<tr>
<td>A_CBAM</td>
<td></td>
<td>85.49%</td>
<td>86.44%</td>
<td>84.55%</td>
<td>82.23%</td>
</tr>
<tr>
<td>CBAM effective feature layers +upsampling</td>
<td></td>
<td>85.77%</td>
<td>86.82%</td>
<td>84.72%</td>
<td>81.44%</td>
</tr>
<tr>
<td>A_CBAM</td>
<td></td>
<td>86.08%</td>
<td>87.08%</td>
<td>85.08%</td>
<td>82.37%</td>
</tr>
</tbody>
</table>

It can be seen from the above table that the effect of adding after upsampling is better than adding after the effective feature layers, and the effect of adding on both positions is the best. The mAP of A_CBAM was 0.31% higher than that of CBAM and 1.12% higher than the original YOLOv4. The AP of maize increased by 1.11%, reaching 87.08%, and the AP of weed increased by 1.14%, reaching 85.08%. These results show that the attention module focuses on important information and suppresses irrelevant details through weight parameters, which can effectively improve the performance of the model.

3.3. Weed Detection Model Comparison Experiment

To further analyze the performance, we compared our method with several other types of object detection models, such as YOLOv3, SSD, and YOLOv4-Tiny. We used the same training set, validation set, and test set to train and test these networks. Table 5 shows the comparisons made for each of these methods.

Table 5. Comparison of the performance of different object detection models.

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP</th>
<th>AP</th>
<th>Recall</th>
<th>Precision</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Maize</td>
<td>Weed</td>
<td>Maize</td>
<td>Weed</td>
</tr>
<tr>
<td>Original YOLOv4</td>
<td>84.96%</td>
<td>85.97%</td>
<td>83.94%</td>
<td>81.49%</td>
<td>73.80%</td>
</tr>
<tr>
<td>SSD</td>
<td>81.76%</td>
<td>83.42%</td>
<td>80.10%</td>
<td>78.59%</td>
<td>69.36%</td>
</tr>
<tr>
<td>YOLOv3</td>
<td>80.55%</td>
<td>82.32%</td>
<td>78.79%</td>
<td>72.77%</td>
<td>63.05%</td>
</tr>
<tr>
<td>YOLOv4-Tiny</td>
<td>73.24%</td>
<td>74.56%</td>
<td>71.93%</td>
<td>67.97%</td>
<td>56.02%</td>
</tr>
<tr>
<td>Improved YOLOv4</td>
<td>86.89%</td>
<td>87.49%</td>
<td>86.28%</td>
<td>83.55%</td>
<td>78.02%</td>
</tr>
</tbody>
</table>

It can be seen from Table 5 that the AP, recall precision, and F1-score of the improved YOLOv4 were all higher than those of the other detection models. The mAP of our model was 1.93%, 5.13%, 6.34%, and 13.65% higher than those of original YOLOv4, SSD, YOLOv3, and YOLOv4-Tiny, respectively. In addition, YOLOv4-Tiny had the worst effect, indicating that the lightweight model is not suitable for identifying complex targets such as maize and weed in the image. The mAP of YOLOv3 and SSD were 4.41% and 3.2%, respectively, which is lower than the original YOLOv4, indicating that it is correct to select YOLOv4 as the baseline. Thus, our improved model can achieve the best effect. The mAP was 86.89%, while the AP of maize was 87.49%, and the AP of weed was 86.28%.

The PR curves of different models are shown in Figure 9. Our proposed model had the best PR curve, especially the PR curve of weed.
Figure 9. The PR curves of different models. The curve area from largest to smallest is: improved YOLOv4, YOLOv4, SSD, YOLOv3, and YOLOv4-Tiny. (a) PR curve of maize, and (b) PR curve of weed.

The detection results of different models are shown in Figure 10. Other models have the problem of missing detection, and some even recognize maize as weed. The findings show that our suggested algorithm accurately and quickly detected maize and weed in images collected in natural situations.

Figure 10. Comparison of prediction maps from different models. The red labels are weeds, and the blue and green labels are maize.

3.4 Maize and Weed Distribution and Counts

We input the image to be predicted into the weed detection model, determined whether the object in the image was maize or weed, and drew bounding boxes for them. We counted the different bounding boxes separately to ascertain the amount of maize and weed, a method which can help estimate the yield of maize and the number of weeds. We spliced all the output images into a complete farmland image, thus obtaining the distribution of maize and weed for the whole farmland, as shown in Figure 11. The number of maize in the experimental field was 88,845, and the number of weeds was 16,976.
3.5. Regional Data Analysis

We divided the whole farmland into several areas, calculated the number and proportion of maize and weed in each area, and plotted the data on the original map. As shown in Figure 12, where \( m \) represents the number of maize, \( w \) represents the number of weeds, and \( r \) represents the proportion of weeds in the number of maize. The green area indicates that the proportion is less than 10%, the blue area indicates that the proportion is between 10 and 20%, the yellow area indicates that the proportion is between 20 and 30%, and the red area indicates that the proportion is greater than 30%. It can be seen that the green area accounts for 22%, the blue for 44%, the yellow for 25%, and the red for 9%. For the green area, we believe that weeds do not affect the living space of maize, so we did not spray herbicides in this area. For the blue, yellow, and red areas, we increased the use of herbicides in turn. Through variable spraying of herbicides, herbicide resources are saved, and the environment is protected. Furthermore, the yield of maize is affected by many factors. Several studies have shown that plant density is related to the yield in fields [36,37]. Through analysis of the statistics of maize in each area, the plant density can be understood and adjusted, as well as helping to facilitate data support for experts.
Figure 12. The image is divided into small areas for data analysis.

4. Conclusions

In this paper, we identified weeds in maize fields by UAV image and achieved good results. A new weed detection framework was proposed. First, a lightweight model YOLOv4-Tiny was exploited to detect and mask the maize rows so that we only needed to label weeds on the masked image, a process that can deal with the serious imbalance of samples and the high cost of labeling. Second, to improve the network recognition accuracy, the Meta-ACON activation function was used to change the CBL module, and the A_CBAM was added to the network. Furthermore, the NMS function was replaced with the Soft-NMS function. The results showed that the proposed model had a maize AP of 87.49%, a weed AP of 86.28%, and a mAP of 86.89% and can thus accurately identify maize and weed. We drew a distribution map of maize and weed, which provided data support for variable herbicide spraying and estimation of yield. In future work, we will propose new versions, such as YOLOv5 [38], to further improve the performance of weed detection.

Author Contributions: Conceptualization, H.P., H.H. and Y.S.; methodology, H.P., W.Z. and J.S.; software, H.P.; validation, H.P., Z.Z., W.Z. and J.S.; investigation, H.P., H.H. and Z.Z.; resources, Y.S. and H.H.; data curation, H.P.; writing—original draft preparation, H.P.; writing—review and editing, H.P., H.H. and Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (Grant 2021YFD200060102) and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant XDA28120400).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.
Appendix A

(a) Meta-ACON

The calculation formula of Meta-ACON is shown in Formula (A1).

\[ f(x) = (p_1 - p_2)x\sigma[\beta(p_1 - p_2)x] + p_2x \]  

(A1)

where \( p_1 \) and \( p_2 \) are two learnable parameters, which are used for adaptive adjustment to control the upper and lower bounds of the function. \( \beta \) dynamically controls the linearity or nonlinearity of the activation function. Specifically, when \( \beta \to \infty, f(x) \to \max(p_1x, p_2x) \); when \( \beta \to 0, f(x) \to \mean(p_1x, p_2x) \). \( \sigma \) denotes the sigmoid function.

(b) CBAM

In the part of the channel attention module, we first use global average-pooling and global max-pooling to aggregate the input feature maps. Then, we merge the output feature maps through shared multi-layer perceptron (MLP). Finally, the channel attention map \( M_c(F) \) is obtained by the sigmoid function. The channel attention process can be summarized by Formulas (A2) and (A3).

\[ M_c(F) = \sigma(\text{MLP}(F_{\text{avg}}^c) + \text{MLP}(F_{\text{max}}^c)) \]  

(A2)

\[ F' = M_c(F) \odot F \]  

(A3)

where \( F_{\text{avg}}^c \) and \( F_{\text{max}}^c \) denote the features of average-pooling and max-pooling, \( \sigma \) denotes the sigmoid function, and \( \odot \) denotes element-wise multiplication. \( F' \) is the output feature maps obtained by the part of channel attention module.

In the part of the spatial attention module, we first apply average-pooling and max-pooling operations along the channel axis and concatenate them. Then, we apply a convolution layer and sigmoid function to generate a spatial attention map \( M_s(F') \). The spatial attention process can be summarized by Formulas (A4) and (A5):

\[ M_s(F') = \sigma(f^{7*7}([F_{\text{avg}}^s, F_{\text{max}}^s])) \]  

(A4)

\[ F'' = M_s(F') \odot F' \]  

(A5)

where \( F_{\text{avg}}^s \) and \( F_{\text{max}}^s \) denote the features of average-pooling and max-pooling, \( f^{7*7} \) represents a convolution operation with the filter size of 7 * 7, \( \sigma \) denotes the sigmoid function, \( \odot \) denotes element-wise multiplication, and \( F'' \) is the attention maps of the final output.

(c) Soft-NMS

The calculation formula of detection score \( S_i \) is shown in Formula (A6).

\[ S_i = S_i e^{-\frac{\text{area}(M, b_i)^2}{\sigma}} \]  

(A6)

where \( M \) is the bounding box with the highest score, \( b_i \) is the bounding box that significantly overlaps with \( M \), and \( \sigma \) denotes the sigmoid function.

References


34. LabelImg. Available online: https://github.com/tzutalin/labellImg (accessed on 26 April 2022).

