

## Article

# Real-Time Detection and Localization of Weeds in *Dictamnus dasycarpus* Fields for Laser-Based Weeding Control

Yanlei Xu <sup>1</sup>, Zehao Liu <sup>1</sup>, Jian Li <sup>1</sup>, Dongyan Huang <sup>1</sup>, Yibing Chen <sup>2</sup> and Yang Zhou <sup>1,\*</sup>

<sup>1</sup> College of Information and Technology, Jilin Agricultural University, Changchun 130118, China; yanleixu@jlau.edu.cn (Y.X.); 20221150@mails.jlau.edu.cn (Z.L.); lijianz@jlau.edu.cn (J.L.); huangdongyan@jlau.edu.cn (D.H.)

<sup>2</sup> Jilin Soil Fertilizer General Station, Changchun 130031, China; jlszrflzz@163.com

\* Correspondence: zhuyang@jlau.edu.cn

**Abstract:** Traditional Chinese medicinal herbs have strict environmental requirements and are highly susceptible to weed damage, while conventional herbicides can adversely affect their quality. Laser weeding has emerged as an effective method for managing weeds in precious medicinal herbs. This technique allows for precise weed removal without chemical residue and protects the surrounding ecosystem. To maximize the effectiveness of this technology, accurate detection and localization of weeds in the medicinal herb fields are crucial. This paper studied seven species of weeds in the field of *Dictamnus dasycarpus*, a traditional Chinese medicinal herb. We propose a lightweight YOLO-Riny weed-detection algorithm and develop a YOLO-Riny-ByteTrack Multiple Object Tracking method by combining it with the ByteTrack algorithm. This approach enables accurate detection and localization of weeds in medicinal fields. The YOLO-Riny weed-detection algorithm is based on the YOLOv7-tiny network, which utilizes the FasterNet lightweight structure as the backbone, incorporates a lightweight upsampling operator, and adds structure reparameterization to the detection network for precise and rapid weed detection. The YOLO-Riny-ByteTrack Multiple Object Tracking method provides quick and accurate feedback on weed identification and location, reducing redundant weeding and saving on laser weeding costs. The experimental results indicate that (1) YOLO-Riny improves detection accuracy for *Digitaria sanguinalis* and *Acalypha australis*, ultimately amounting to 5.4% and 10%, respectively, compared to the original network. It also diminishes the model size by 2 MB and inference time by 10 ms, making it more suitable for resource-constrained edge devices. (2) YOLO-Riny-ByteTrack enhances Multiple Object Tracking accuracy by 3%, reduces ID switching by 14 times, and improves overall tracking accuracy by 3.4%. The proposed weed-detection and localization method for *Dictamnus dasycarpus* offers fast detection speed, high localization accuracy, and stable tracking, supporting the implementation of laser weeding during the seedling stage of *Dictamnus dasycarpus*.

**Keywords:** laser weeding; weed detection; weed localization; *Dictamnus dasycarpus*; deep learning



**Citation:** Xu, Y.; Liu, Z.; Li, J.; Huang, D.; Chen, Y.; Zhou, Y. Real-Time Detection and Localization of Weeds in *Dictamnus dasycarpus* Fields for Laser-Based Weeding Control. *Agronomy* **2024**, *14*, 2363. <https://doi.org/10.3390/agronomy14102363>

Academic Editor: Silvia Arazuri

Received: 13 September 2024

Revised: 2 October 2024

Accepted: 11 October 2024

Published: 13 October 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

*Dictamnus dasycarpus* is a perennial herbaceous plant belonging to the *Rutaceae* family, favoring high, dry, sunny, and slightly acidic soil conditions. It is predominantly distributed across the Jilin, Liaoning, and Heilongjiang provinces in China. The root bark, known as Dictamni Cortex, is highly esteemed for its medicinal properties and is employed in traditional Chinese medicine for treating various forms of dermatitis. Its therapeutic effects include clearing heat, detoxifying the body, and alleviating wind and dampness. During the seedling stage, *Dictamnus dasycarpus* requires high soil fertility, necessitating the application of 1500–2000 kg of Sheep manure organic fertilizer per acre. This nutrient-rich environment often results in vigorous weed growth [1], making effective weed management essential during this developmental phase [2,3].

In China, the control of weed-related harm to crops mainly relies on chemical methods, with large-scale spraying using agricultural machinery being the prevalent approach [4,5]. *Dictamnus dasycarpus* has high soil environmental requirements, and extensive, quantitative spraying can lead to soil contamination, which negatively impacts the medicinal value of the herbs [6,7]. In managing medicinal herb fields, manual weeding is often preferred to mitigate the impact of herbicides on the quality of Chinese herbal medicines due to its precision and absence of pesticide residues. However, for *Dictamnus dasycarpus*, a perennial herb with valuable medicinal properties concentrated in its delicate rhizomes, manual weeding can harm the root system and is both labor-intensive and costly. With advancements in agricultural technology, laser weeding has emerged as an effective and eco-friendly solution [8,9]. This method works by inducing cell dehydration and rupture to efficiently eliminate weeds. It is particularly well-suited for the precise management of *Dictamnus dasycarpus* fields. The primary challenge with laser weeding technology is accurately distinguishing between weeds and crops and precisely targeting the weeds.

In recent years, with the ongoing advancements in deep learning, artificial intelligence technology has been increasingly integrated into agricultural production [10–13]. Among its various applications, target detection has become a key technology in this field. It has been widely employed in farmland weed management, and many researchers have explored the use of deep learning techniques in laser weeding, achieving notable results. Zhu et al. [14] proposed a YOLOX-based convolutional neural network for weeding robots to manage maize seedling fields, demonstrating the feasibility of using blue laser technology as a non-contact weeding tool. Mwitwa et al. [15] employed a target-detection algorithm for diode laser weeding and utilized target tracking for path navigation. While these studies have validated the potential of deep learning techniques for laser weeding, they did not address issues related to detection accuracy and the need for lightweight models. Given the dense planting of *Dictamnus dasycarpus* seedlings, the variety of weed species, and the constraints of edge computing resources, models must achieve high detection accuracy while remaining as lightweight as possible. As target-detection technology continues to advance and models evolve, numerous new weed-detection models have emerged. Zhu et al. [12] introduced the YOLOX weed-detection model, which integrates a deep network with a lightweight attention mechanism to effectively identify various species of weeds in corn seedling fields. The optimized model achieved an average detection accuracy of 94.8%. Liu et al. [16] proposed a corn seedling and weed-detection method based on the YOLOv4-minor model. This algorithm achieves an accuracy of 86.6% and a detection speed of 57.3 frames per second. Shao et al. [17] presented an optimized deep learning framework, GTCBS-YOLOv5s, designed to accurately identify six weed species in paddy fields, which achieves an average accuracy of 91.1% on the test set. Peng et al. [18] developed a deep learning model named WeedDet based on RetinaNet for weed detection in rice crop images, which achieves an average detection accuracy of 94.1%. All the studies mentioned employ deep learning for weed detection in crop fields, balancing high detection accuracy with the feasibility of edge deployment. The YOLOv7 algorithm is a single-stage detection algorithm that has the advantages of high robustness and high accuracy when used in real experiments. Due to the stability of the YOLOv7 algorithm, it is very suitable for pairing with target tracking algorithms as a trade-off between accuracy and stability. However, laser weeding faces challenges. As a single-detection algorithm, it may struggle to manage the times of spot removal for the same weed, leading to cost inefficiencies. Therefore, minimizing repeated weeding and effectively managing the numbers of spot weeding for each weed is crucial to avoid unnecessary costs. Developing fast, accurate, and stable Multiple Object Tracking (MOT) algorithms is essential for addressing this issue [19–21].

Recent reports on the advancements and applications of deep learning-based tracking algorithms in military aerospace, security monitoring, intelligent driving, and smart agriculture have provided valuable references for the work presented in this paper [21–24]. Wang et al. [25] demonstrated the use of a Kalman filter for tracking mango motion in orchard videos, facilitating automatic fruit counting. Meanwhile, Li et al. [26] utilized Byte-

Track along with detection algorithms to classify and count dragon fruit flowers, immature green fruits, and ripe red fruits in a plantation. Similarly, Özlüoymak et al. [27] combined machine vision with a mobile spraying system to achieve precise micro-dose spraying of weeds in the field. The above studies employed a combination of object detection and tracking methods, using tracking algorithms to achieve ID matching for the same target appearing across multiple frames in the video. Combining object detection with tracking is an effective solution to address the issue of duplicate detections of the same object, and it offers a cost-optimization strategy for laser weeding.

### 1.1. Main Objectives

The following are the main ideas behind the work in this paper:

1. Create a dataset of the most widely distributed and numerous weeds in real *Dictamnus dasycarpus* fields, covering a number of important real disturbances.
2. Propose a detection network that can trade off accuracy and light weight to enable weed localization in complex scenarios.
3. Provide joint tracking and detection algorithms and achieve stable performance as well as fast matching of tracking trajectories.
4. Perform goal-oriented training and improvement of the model for real-world noise as well as various disturbances.

### 1.2. Research Contribution

Existing detection algorithms are difficult to trade off the accuracy of detecting multiple classes of weeds with the lightweight nature of the model, making it difficult to embed into the edge devices of laser weeding systems. To achieve precise weed detection in *Dictamnus dasycarpus* fields and address the constraints of edge devices, this paper introduces an enhanced lightweight detection algorithm called YOLO-Riny, based on YOLOv7-tiny. Furthermore, it combines YOLO-Riny with ByteTrack [28] to develop the YOLO-Riny-ByteTrack multi-object tracking algorithm. In this research, the edge device equipped with the YOLO-Riny-ByteTrack algorithm is employed as a reference system, where weeds entering the camera's field of view are treated as moving targets for tracking. The YOLO-Riny-ByteTrack system can establish continuous tracks for individual weeds within the video feed, assign unique IDs to detected objects in each frame, and synchronize these tracks with the IDs of weeds in the field. This approach addresses the issue of redundant detection and removal costs by preventing multiple detections of the same weed during laser spot weeding operations. It effectively balances cost control with the need for real-time performance and accuracy in weed detection, thereby enhancing the efficiency of precision agriculture practices.

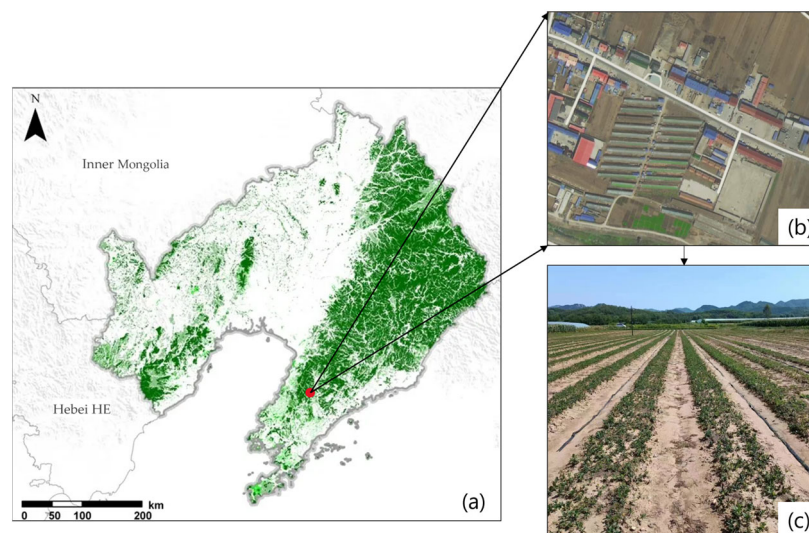
## 2. Materials and Methods

### 2.1. Dataset Acquisition

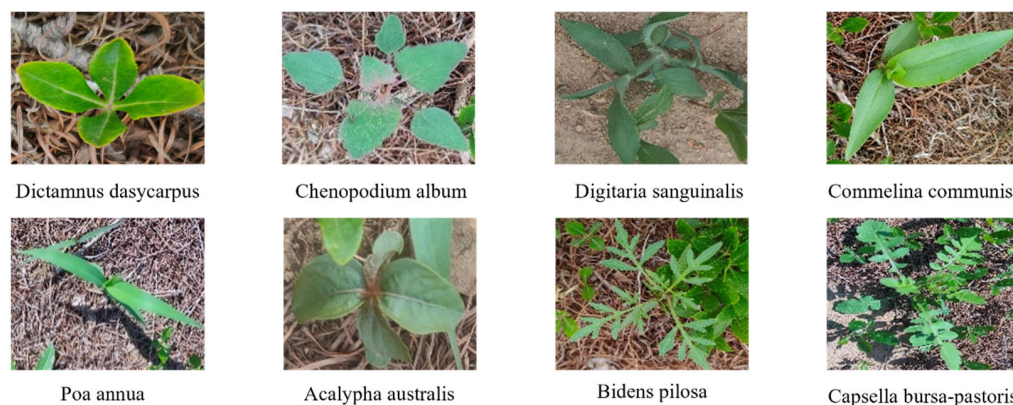
The dataset for this research was collected in Shizi Street Town, Gaizhou City, Liaoning Province, as illustrated in Figure 1. In late June 2023, the research team gathered both image and video data of weeds within the *Dictamnus dasycarpus* fields. This period coincided with the dense planting and seedling stage of *Dictamnus dasycarpus*, where the density reached approximately 50–100 seedlings per square meter.

To simulate the laser weeding operation, a camera mounted on a Realme GT was used (Rear primary camera: 64 MP), positioned 70–80 cm above the ground. The dataset comprises 3048 images, each of  $3456 \times 3456$ -pixel resolution, stored as JPG files. Additionally, 15 min of video footage was recorded in MP4 format at a resolution of  $720 \times 1280$  pixels. To ensure the robustness of the model, the dataset included various lighting conditions (Sunny 9 a.m.–10 a.m.), ranging from strong to low light (Cloudy 2 p.m.–4 p.m.). For the practical application value of the algorithm, the researchers selected the seven most abundant and widely distributed weeds in local *Dictamnus dasycarpus* fields: *Dictamnus dasycarpus*, *Chenopodium album*, *Digitaria sanguinalis*, *Poa annua*, *Acalypha australis*, *Commelina*

*communis*, *Bidens pilosa*, and *Capsella bursa-pastoris*. Figure 2 provides examples of these categories, cropped from the dataset images.



**Figure 1.** *Dictamnus dasycarpus* cultivation base in Shizijie Town, Gaizhou City, Liaoning Province. (a) Location map of Liaoning Province; (b) Location map of Shizijie Street, Gaizhou City; (c) *Dictamnus dasycarpus* cultivation base.

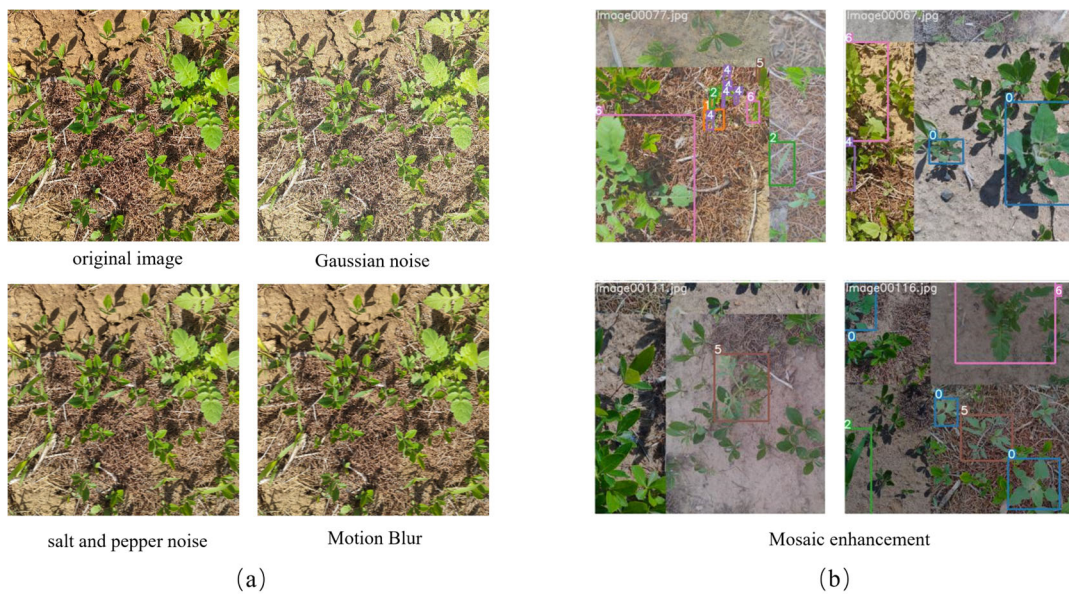


**Figure 2.** Examples of the seven selected weed species and crop *Dictamnus dasycarpus* from the dataset images.

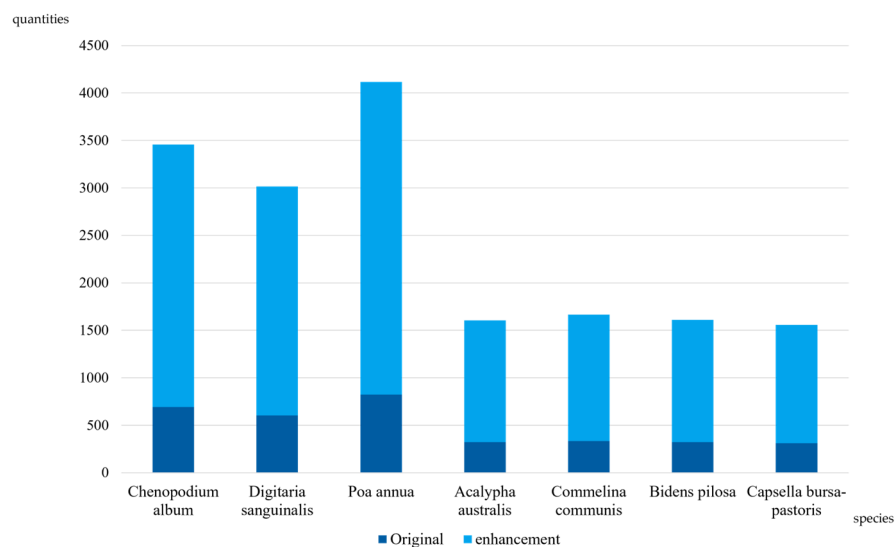
## 2.2. Data Preprocessing and Augmentation

In this paper, we apply bilinear interpolation to reduce image resolution and shorten model training time [29]. The dataset is further augmented using data enhancement techniques such as motion blur, Gaussian noise, and pretzel noise. Figure 3a illustrates an example of data enhancement applied to an image of the captured *Dictamnus dasycarpus* field weed. This measure helps prevent overfitting due to insufficient data in the detection model. It also improves the model's precision during target tracking and reduces the miss rate by training the detector on images affected by motion blur. Additionally, this paper employs the Mosaic algorithm to enhance images with online data, as shown in Figure 3b. The Mosaic enhancement algorithm is an image synthesis technique that randomly selects two to four images from the dataset at a time. It performs operations such as flipping (both horizontally and vertically), scaling (resizing the original image), and color adjustments (modifying brightness, saturation, and hue). These processed images are then stitched together to create enhanced training data. The Mosaic technique facilitates increased diversity of data, enriched background of images, and increased number of training targets concurrent learning of various image attributes, enhancing the model's generalization

capacity. Figure 4 illustrates the comparison of data quantities before and after enhancement. After enhancement, the dataset comprised 2764 plants of *Chenopodium album*, 2414 plants of *Digitaria sanguinalis*, 3292 plants of *Poa annua*, 1284 plants of *Acalypha australis*, 1332 plants of *Commelina communis*, 1288 plants of *Bidens pilosa*, and 1244 plants of *Capsella bursa-pastoris*. Among the data collected in this research, *Poa annua* was the most densely clustered, often growing in groups of four to six plants. This dense clustering makes *Poa annua* the most challenging species to detect. *Capsella bursa-pastoris* and *Chenopodium album*, with their larger target areas, were more easily detectable. In contrast, *Digitaria sanguinalis*, *Commelina communis*, and *Poa annua* shared similar shapes, which led to confusion and challenges in accurate classification. Additionally, *Acalypha australis*, due to its smaller target area, was more prone to being overlooked, which slightly increased the detection difficulty.



**Figure 3.** Examples of data-enhancement techniques applied to weed images. (a) Data enhanced visualization; (b) Mosaic enhanced visualization. Online enhanced labels 0: *Chenopodium album*; 1: *Acalypha australis*; 2: *Poa annua*; 4: *Acalypha australis*; 5: *Bidens Pilosa*; 6: *Capsella bursa-pastoris*.



**Figure 4.** Comparison of sample sizes before and after dataset pre-processing.

### 2.3. Weed-Detection Model

#### 2.3.1. YOLOv7-Riny Weed-Detection Model

YOLOv7-tiny is a lightweight target-detection model derived from YOLOv7 [30]. It features a lightweight network architecture, making it well-suited for deployment on resource-constrained devices.

In practical applications, the tracker demands both high detection speed and stability from the detector. Additionally, the complex field environment requires the detector to differentiate between multiple weeds with similar characteristics. When YOLOv7-tiny is deployed on edge devices and combined with the tracker, its floating points of operations (FLOPs) and inference speed fall short of meeting the tracking task’s requirements. The limited sensing field of YOLOv7-tiny restricts its ability to identify smaller or closely spaced weed targets and effectively capture fine-grained features in complex environments, thus impacting detection accuracy. To address these issues, this paper introduces the YOLO-Riny target-detection model, which builds upon the YOLOv7-tiny framework. The network architecture of YOLO-Riny is illustrated in Figure 5a, and its module composition is detailed in Figure 5b.

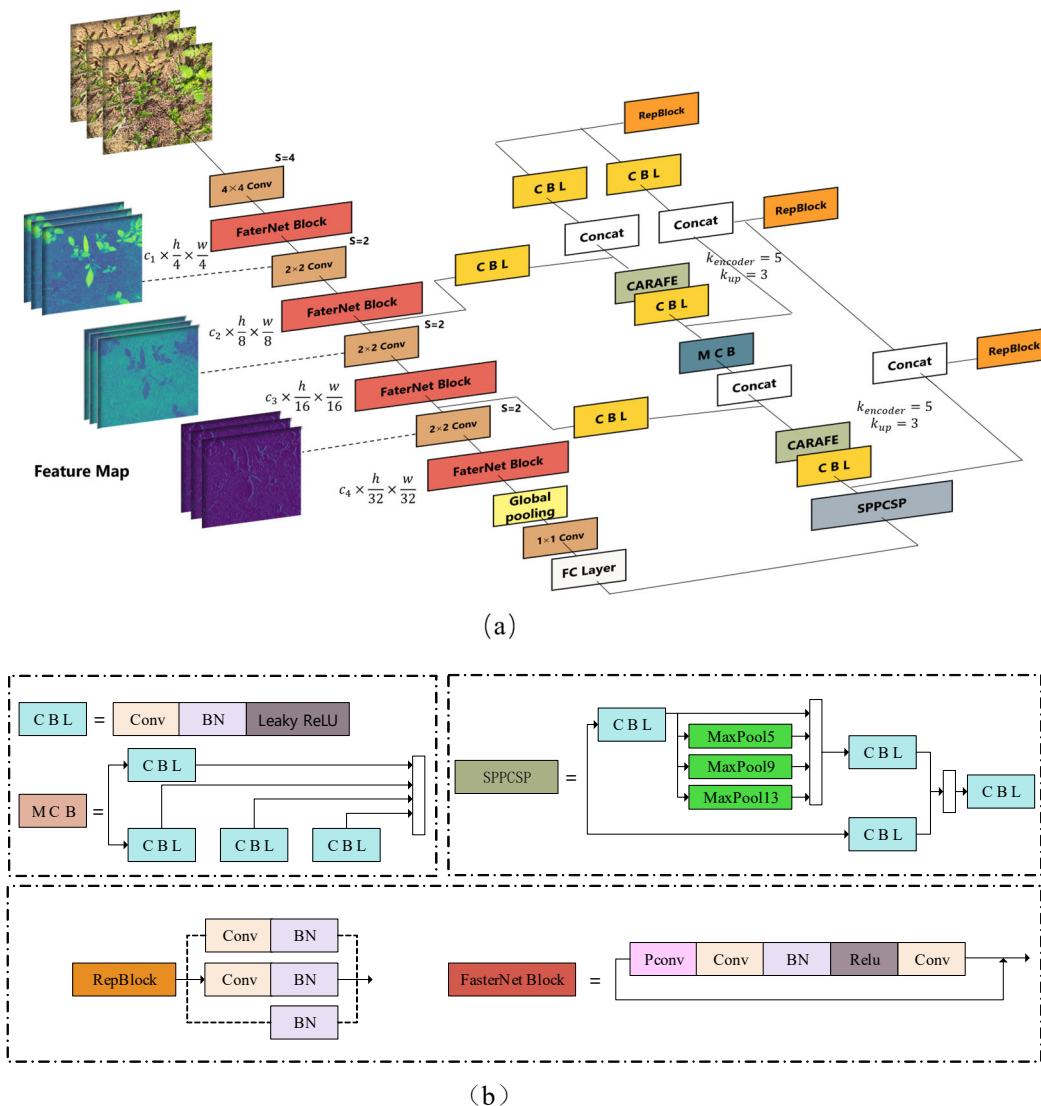


Figure 5. Architecture of YOLO-Riny network model. (a) General model structure; (b) Module internal structure.

In this research, a weed image from a natural *Dictamnus dasycarpus* field environment is used as input, with the image size set to  $640 \times 640 \times 3$ . The image is processed through the Backbone feature extraction component, which is organized into four stages. Each stage includes a FasterNet Block and either an Embedding Layer (a standard  $4 \times 4$  convolution with a stride of 4) or a Merging Layer (a standard  $2 \times 2$  convolution with a stride of 2) for spatial downsampling and channel expansion. The intermediate layer of the FasterNet Block is constructed with an increased channel count and includes shortcut connections for input feature reuse, thereby enhancing feature diversity and reducing latency.

The three feature layers output from the Backbone are fed into the Spatial Pyramid Pooling System (SPPCSP) module and the feature pyramid structure. This setup aggregates information from various scales through multiple branches. The fused feature layers are fed into the head network and into the upsampling layer. The lightweight upsampling operator (CARAFE [31]) employs a  $3 \times 3$  upsampling kernel and a  $5 \times 5$  convolutional layer. This design provides robust local feature fusion while maintaining a compact computational footprint, enhancing the model's ability to detect small and densely packed targets in complex scenes. Finally, the feature layer is fed into the RepBlock module, which uses structural reparameterization [32] to transform the inference into a single-branch structure. This approach improves inference speed while preserving accuracy and enables weed detection through score screening and non-maximum suppression.

YOLO-Riny surpasses YOLOv7-tiny in detecting smaller, ambiguously classified weeds. It offers faster inference, a smaller model size, and reduced computational complexity, which benefits tracking operations. Additionally, YOLO-Riny shows improved deployability on edge-computing platforms.

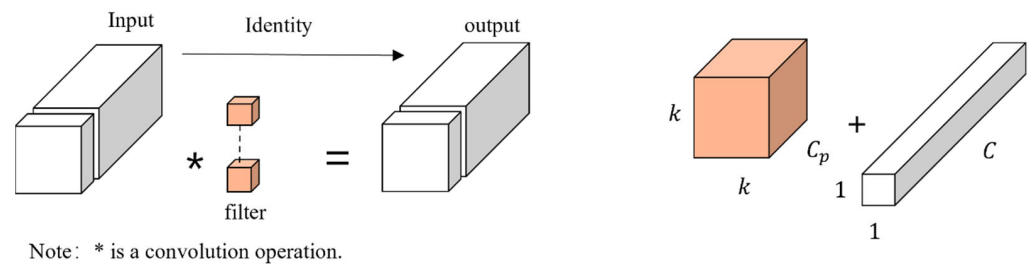
### 2.3.2. Constructing a Lightweight Backbone Network Using Partial Convolution

The backbone network of YOLOv7-tiny mainly consists of a convolutional module composed of LeakyReLU and four MCB modules. The network modules are concise, and the floating points of operations (FLOPs) are small, which reduces the computational complexity. However, the combination of the original YOLOv7-tiny with the tracking algorithm yields unsatisfactory results in testing. This is because the reduction in its floating-point operations (FLOPs) does not correspondingly maintain the Floating-point Operations Per Second (FLOPS).

To balance the trade-off between the size of FLOPs and FLOPS, this paper employs Partial Convolution [33] (PConv). The structure of PConv is illustrated in Figure 6. PConv is designed to address the drawbacks of frequent memory accesses in Deep Separable Convolution [34] (DWConv), and is uniquely designed to remove the redundant structure. Due to the redundancy of convolutional neural network features, the feature maps between different channels have high similarity. Therefore, to simultaneously reduce memory accesses and computational redundancy, PConv applies regular convolution to only a subset of the input channels for spatial feature extraction, while leaving the remaining channels unchanged. For consecutive or regular memory accesses, PConv computes either the first or last consecutive  $C_p h w$  channels as a representative of the entire feature map. Regular convolution is then performed only on the remaining  $(C - C_p)$  channels. PConv maintains the remaining channels unchanged to reduce redundancy and ensure that feature information passes through all channels, thus facilitating the subsequent layers of PWConv. PConv's distinctive structure preserves a high FLOPS rate while decreasing the number of floating-point operations.

The environment of the *Dictamnus dasycarpus* field is complex, and using PConv for primary feature extraction may result in excessive loss of detailed information while discarding redundant features. To address both the strengths and limitations of PConv, this paper introduces two  $1 \times 1$  pointwise convolutions following PConv. This approach maximizes the use of information across all feature channels and preserves more valuable detailed features, all without adding significant computational overhead. The combination of PConv and pointwise convolution not only preserves the algorithm's FLOPS size but

also enhances the richness and accuracy of weed feature extraction. This module structure is illustrated in the FasterNet Block shown in Figure 5b. It effectively captures texture and shape information, which is crucial for improving the accuracy of weed detection. In this paper, four FasterNet Blocks are employed in the Backbone component to progressively enhance the network's ability to recognize field weeds, including those in the *Dictamnus dasycarpus* field, through multiple stages of feature extraction and fusion. Additionally, a global pooling layer is introduced at the end of the Backbone to further aggregate global context information and improve feature characterization. Global pooling effectively compresses the entire feature map, extracts global features, and simplifies subsequent classification and detection tasks. The improved Backbone is designed to adapt to the complex environments of the *Dictamnus dasycarpus* field, achieving efficient and accurate weed detection while remaining lightweight.



**Figure 6.** PConv structure.

### 2.3.3. Lightweight Upsampling Operator: CARAFE

The upsampling used in the original YOLOv7-tiny is an upsampling operation using the traditional interpolation method. Traditional interpolation methods estimate the value of a new pixel by averaging the values of existing pixels, which can result in some information loss. At large upsampling factors, detailed information can become easily blurred. In *Dictamnus dasycarpus* fields, weeds such as *Digitaria sanguinalis*, *Commelina communis*, and *Acalypha australis* are small and require the detector to accurately extract fine features. Additionally, during tracking tasks, complex real-world environmental factors often cause video blurring, complicating the extraction of detailed weed features. To enhance the model's receptive field and maintain a lightweight architecture, this paper uses Content-Aware Reassembly of Features (CARAFE) in place of the traditional upsampling layer. CARAFE reconstructs features within a set domain centered at each location through a weighted aggregation, where weights are computed in a context-sensitive fashion. The weights are generated in a content-aware manner. At each location, CARAFE leverages inherent content information to estimate the reorganization kernel and refines features within a specified local region. By employing adaptive and optimized reorganization kernels at different locations, CARAFE rearranges the features into spatial blocks for upsampling. This approach provides improved performance compared to conventional upsampling operators, augmenting the model's capability to capture fine details even in complex and blurred video scenarios.

The structure of CARAFE is shown in Figure 7. CARAFE consists of two key components, namely the prediction kernel module and the content-aware reorganization module. In this figure, the feature mapping of size  $C \times H \times W$  is upsampled. The prediction kernel module consists of three sub-modules: the channel-compression module, the content-encoding module, and the kernel-regularization module. After the channel-compression module reduces the channels of the input feature maps, the content encoder takes the compressed feature maps as input and encodes the content to generate the reconstruction kernel. Finally, the kernel-regularization module uses a Softmax function for each reconstructed kernel. The restructured feature maps contain richer semantic information compared to the original ones, as they emphasize relevant details within local regions more effectively.



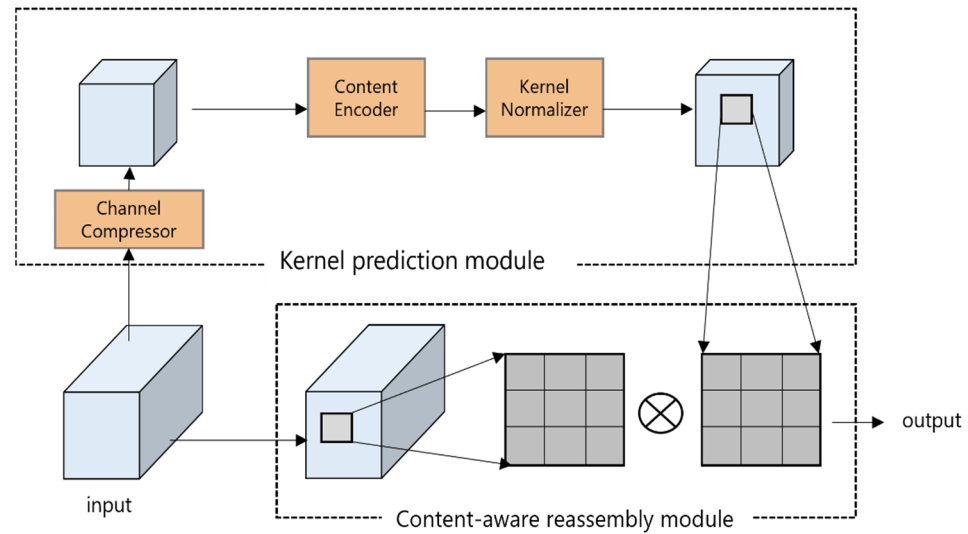


Figure 7. The structure of CARAFE.

### 2.3.4. Constructing the Reparameterized Convolution RepBlock

In target tracking tasks, the tracker must locate and update the target in each frame while using detection results to match the trajectory. This demands high inference speed from the detector and a careful balance between the detector’s performance and the tracker’s efficiency. Thus, finding the right balance between model performance and lightweight design is a key challenge addressed in this paper. Structural reparameterization involves integrating reparameterized structures into the network during training to enhance model representation. During inference, this technique simplifies complex structures, leading to faster computation. Given its nature, structural reparameterization benefits both detection and tracking tasks. By incorporating structural reparameterization into the original model, it is possible to boost performance while also reducing computational demands. The RepBlock module described in this paper features a three-branch structure, as illustrated in Figure 8. During training, RepBlock employs a multi-branch configuration with  $1 \times 1$ ,  $3 \times 3$ , and constant branches. However, during inference, all branches are consolidated into a single  $3 \times 3$  convolutional topology. This approach allows the network to leverage a multi-branch model for iterative training, where different branches apply various convolutional kernels to capture diverse receptive fields, thus enhancing model performance. In contrast, the unified single-branch structure during inference significantly reduces computation time.

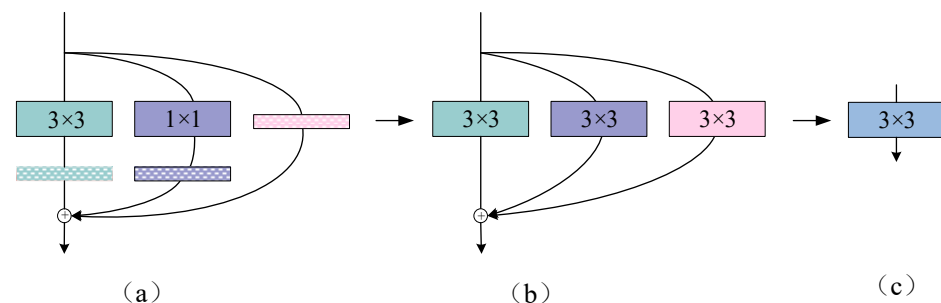


Figure 8. Structure of the RepBlock. (a) Training module structure; (b) BN Layer fusion; (c) Reasoning module structure.

When the trained RepBlock structure is reparameterized and the multi-branch convolution is equivalently converted to a single  $3 \times 3$  convolution for inference, the convolution layer is first fused with the Batch Normalization (BN) layer. This process produces the constant branch fusion output  $BN_{\gamma,\beta}(V_0, \mu_0, \sigma_0)$ , as represented in Equation (1).

$V$  stands for input,  $O$  stands for output,  $\forall 1 \leq i \leq C_2$  ( $C_2$  stands for the number of output channels);  $\mu, \sigma, \gamma, \beta$  stand for the mean, variance, scaling factor, and bias of the  $BN$  layer, respectively;  $\mu_{(0)}, \sigma_{(0)}, \gamma_{(0)}, \beta_{(0)}$  stand for the mean, variance, scaling factor, and bias of the constant branch  $BN$  layer.

$$BN_{\gamma,\beta}(V, \mu_0, \sigma_0) = (V - \mu_0) \frac{\gamma_0}{\sigma_0} + \beta_0 \quad (1)$$

For the  $1 \times 1$  convolutional branch with the  $3 \times 3$  convolutional branch,  $BN_{\gamma,\beta}(V * K_i, \mu_i, \sigma_i)$  fusion outputs are shown in the expression in Equation (2):

$$BN_{\gamma,\beta}(V, \mu_i, \sigma_i) = (V * K_i) + b'_i \quad (2)$$

where  $K_i$  represents the weight parameter and bias of each convolution kernel. For the Figure 8a  $1 \times 1$  convolutional branch with the  $3 \times 3$  convolutional branch structure re-parametrization output in Figure 8, see Equation (3) shown.  $\mu_{(1)}, \sigma_{(1)}, \gamma_{(1)}, \beta_{(1)}$  and  $\mu_{(3)}, \sigma_{(3)}, \gamma_{(3)}, \beta_{(3)}$  represent the mean, variance, scaling factor, and bias of  $1 \times 1$  convolutional branching and  $3 \times 3$  convolutional branching  $BN$  layers, respectively:

$$O = BN_{\gamma_{(1)},\beta_{(1)}}(V * K_{(1)}, \mu_{(1)}, \sigma_{(1)}) + BN_{\gamma_{(3)},\beta_{(3)}}(V * K_{(3)}, \mu_{(3)}, \sigma_{(3)}) \quad (3)$$

The output from the parameterization of the residual branching structure is added to Equation (3) and is shown in Equation (4):

$$O = BN_{\gamma_{(1)},\beta_{(1)}}(V * K_{(1)}, \mu_{(1)}, \sigma_{(1)}) + BN_{\gamma_{(3)},\beta_{(3)}}(V * K_{(3)}, \mu_{(3)}, \sigma_{(3)}) + BN_{\gamma_{(0)},\beta_{(0)}}(V, \mu_{(0)}, \sigma_{(0)}) \quad (4)$$

#### 2.4. Field Weed Tracking Model for *Dictamnus dasycarpus*

In practical applications, the diversity of weeds in *Dictamnus dasycarpus* fields and their morphological similarity to *Dictamnus dasycarpus* seedlings create substantial feature interference, complicating the task of maintaining stable tracking performance. Thus, the tracking model must efficiently and reliably handle complex scenes to accurately track these weeds.

The lightweight ByteTrack architecture enables effective tracking and localization of the same weed plant even when the edge device is in motion. ByteTrack employs a Multiple Object Tracking approach based on the Tracking-By-Detection paradigm, allowing repeated matching and updating of detection targets throughout the tracking process. Unlike many methods that associate only high-confidence detection boxes and discard low-confidence ones, ByteTrack retains both high-score and low-score detections. By extracting target information and eliminating background noise, ByteTrack reduces detection loss and enhances trajectory continuity.

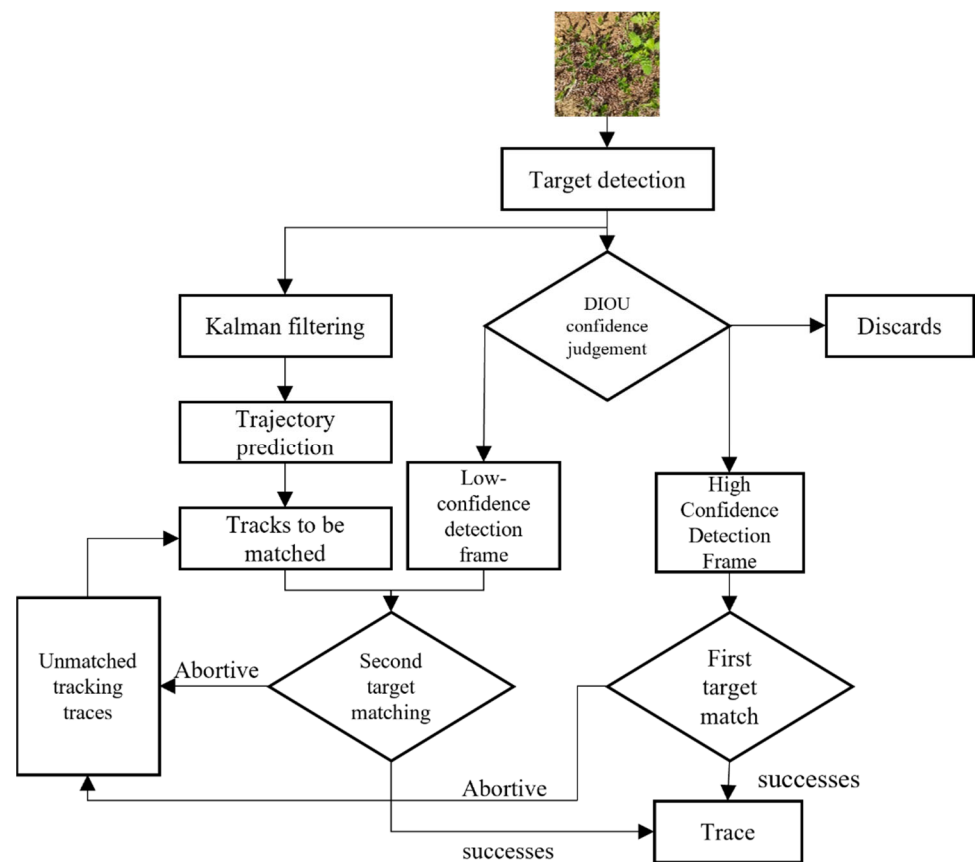
In real-world experiments, camera motion jitter or noise often causes the ByteTrack algorithm to suppress successive low-scoring detection frames with excessively high thresholds during non-maximal suppression (NMS), leading to missed detections in tracking. To overcome the limitations of traditional NMS, which depends exclusively on Intersection over Union (IoU) for filtering, this research proposes using Distance-IoU ( $DIOU$ ) as an alternative evaluation criterion. The  $DIOU$  evaluation criterion is detailed in Equation (5):

$$DIOU = \frac{\rho^2(A, B)}{z^2} \quad (5)$$

where  $\rho^2(A, B)$  is the Euclidean distance between the coordinates of the center points of frames  $A$  and  $B$ , and  $z$  is the diagonal length of the smallest outer rectangle of frames  $A$  and  $B$ .  $DIOU$  provides the direction of movement for the target frames that do not overlap between the two frames. By introducing the  $DIOU$  evaluation criterion, this research aims to improve the screening accuracy of the NMS process for low-scoring detection

frames. *DIOU* can better reflect the relative positional relationship between the target frames; especially when the target frames do not overlap or only overlap slightly, *DIOU* can provide a more effective screening basis than *IOU*. The experimental results show that, compared with the traditional *IOU*, *DIOU* is able to retain the low-scoring detection frames more effectively, decrease the rate of detection failures, and enhance tracking consistency when dealing with the target-tracking task in dynamic scenes and complex backgrounds.

The flowchart of the tracking algorithm is shown in Figure 9. First, the target detector identifies the weed's position in the initial frame, determines its category, and identifies features such as color, texture, and shape. In the subsequent frame, the Kalman filter algorithm estimates the weed's movement relative to the camera, adjusts the detection frame based on this estimation, and then the detector re-infers the target around the adjusted position, extracts its features, matches them with the previous frame's target, and determines if the targets in both frames are the same weed.



**Figure 9.** Flowchart of ByteTrack algorithm.

Three outcomes can result from the matching process: (1) Successful match—track the target and update the trajectory; (2) Fail to match the detector frame—create a new tracking trajectory for the target; (3) Match failure—delete the existing trajectory. Failed tracks are retained for 30 frames; if the target is successfully matched again within this period, it returns to outcome 1. If no match is found within this time frame, the track is deleted.

## 2.5. Experimental Evaluation Indicators

### 2.5.1. Performance Metrics Evaluation Metrics for YOLO-Riny Model Performance

In order to objectively assess the YOLO-Riny model's effectiveness, this paper evaluates key performance metrics including memory occupation, floating point of operations (FLOPs, denoted as  $F$ ), mean average precision ( $AP$ ), mean average precision across seven weed species (mAP), Precision, Recall, server computational speed, and Frames Per Second (FPS).  $AP$  represents the precision for each class, computed by the area under the precision-

recall curve, as indicated in Equation (6), where  $p$  denotes Precision and  $r$  denotes Recall. mAP is employed to provide a comprehensive evaluation of the target-detection model's performance by averaging the  $AP$  values across the seven weed species.  $F$  is calculated as shown in Equation (7), where  $C_i$  denotes the count of input channels of the convolutional layer,  $K$  represents the size of the convolutional kernel of the convolutional layer,  $H$  and  $W$  represent the dimensions (height and width) of the convolutional layer output,  $C_O$  signifies the number of output channels for this convolutional layer,  $I$  denotes the input parameter of the fully connected layer, and  $O$  denotes the number of output channels of the fully connected layer.

$$AP_i = \int_0^1 p_i(r_i) dr_i \quad (6)$$

$$F = (2C_i K_s^2 - 1) HWC_0 + (2I - 1)O \quad (7)$$

### 2.5.2. Evaluation Metrics for Tracking Algorithm Performance

In this paper, the number of weed ID switching, Multiple Object Tracking Accuracy (MOTA), and Multiple Object Tracking Precision (MOTP) are employed to assess the performance of the tracking algorithm. The number of weed ID switches refers to the number of times a weed with a changed ID appears in the video. A smaller value indicates better performance. The Multiple Object Tracking Accuracy metric quantifies the efficacy of the tracking algorithm in preserving tracking trajectories; elevated values correspond to superior performance, as delineated in Equation (8):

$$MOTA = 1 - \frac{\sum(N_{FN} + N_{FP} + N_{IS})}{\sum N_{GT}} \quad (8)$$

where  $N_{FN}$  is the number of false negatives,  $N_{FP}$  is the number of false positives,  $N_{IS}$  is the number of ID jumps, and  $N_{GT}$  is the number of ground truth targets.

Multiple Object Tracking Accuracy denotes the success rate of correctly associating all tracked targets. Increased accuracy reflects an enhanced performance of the detection algorithm, as expressed in Equation (9):

$$MOTP = \frac{\sum_{i,i} d_{i,t}}{\sum_t c_t} \quad (9)$$

where  $d_{i,t}$  is the intersection and concatenation ratio of the predicted and real frames,  $t$  is the video frame sequence number,  $i$  is the current predicted target, and  $c_t$  is the number of successful matches.

## 3. Results and Discussion

### 3.1. Experimental Environment

The server equipment used for training the models ran on the Windows 10 operating system, while the software environment was configured with Python 3.8 and Pytorch 1.9.0. The hardware equipment used an Intel i7-7820X CPU with a main frequency of 3.60 GHz, as well as two TitanXp GPUs with 12.0 GB memory each. The CUDA version used was 11.0.

### 3.2. Results and Analysis of the Weed-Detection Experiment

#### 3.2.1. Ablation Research Analysis

To evaluate the efficacy of the introduced approach for identifying weeds, an experiment was conducted by incorporating and removing the enhanced modules based on YOLOv7-tiny. A comparative performance analysis was carried out, and the results are presented in Table 1. This research comprehensively evaluates the performance effect of

the model before and after the model improvement in seven aspects: mAP, AP, Precision, Recall, model inference time, model occupied memory, and FLOPs, respectively. The data presented in Table 1 indicate that the original YOLOv7-tiny demonstrates very high detection accuracy for *Chenopodium album* and *Capsella bursa-pastoris*. However, it shows slightly lower accuracy for *Digitaria sanguinalis*, *Acalypha australis*, and *Commelina communis*. This reduced performance is attributed to the similarities between *Digitaria sanguinalis* and *Commelina communis* at certain growth stages, which makes classification challenging for the detector. Additionally, *Acalypha australis* displays varying characteristics across different growth stages, making it difficult to accurately extract features from small targets.

**Table 1.** Comparative Results of Model Designs in Average Precision, Precision, and Recall Performance.

Model			Accuracy/%									
Faster-Net	CARAFE	Rep-Block	mAP /%	Chenopodium Album	Digitaria Sanguinalis	Poa Annua	Acalypha Australis	Commelina Communis	Bidens Pilosa	Capsella Bursa-Pastoris	Precision /%	Recall /%
×	×	×	89.8	98.0	81.2	94.4	81.4	81.1	94.2	98.3	83.8	88.6
✓	×	×	88.7	90.7	86.6	87.5	85.9	83.6	92.0	94.6	85.6	89.1
×	✓	×	89.4	92.9	84.1	90.5	86.1	85.4	91.5	93.7	84.4	87.1
×	×	✓	89.4	97.5	80.1	92.2	80.6	80.8	94.0	98.0	83.2	88.4
✓	✓	×	89.7	91.9	87.4	91.5	87.3	84.4	92.1	93.4	86.1	88.1
✓	×	✓	89.8	94.7	82.0	87.9	90.1	83.3	94.5	95.9	84.7	89.2
×	✓	✓	90.4	94.3	85.4	92.1	86.3	87.2	93.0	94.2	87.7	85.6
✓	✓	✓	91.7	98.3	86.6	92.7	91.4	82.0	93.2	98.0	86.7	90.7

Note: The symbol “✓” indicates that improvements were made using the module, and “×” indicates that they were no.

Based on the analysis presented in Table 1, the improved YOLO-Riny shows increases in precision and recall by 2.9% and 2.1%, respectively, with a 1.9% improvement in mean average precision. These enhancements indicate that the feature reorganization better captures semantic information in multi-category images. Additionally, appropriately reducing the depth of the backbone network enhances the extraction of shallow features from the data. Although YOLO-Riny exhibited lower accuracy in some individual weed categories, it achieved improved detection accuracy for *Digitaria sanguinalis*, *Acalypha australis*, and *Commelina communis* by 5.4%, 10%, and 0.9%, respectively, compared to the original network. This led to a more balanced overall accuracy across categories, addressing the issue of significant underrepresentation of certain categories in the original model. These improvements suggest that expanding the sensory field enhances the detection of small targets and better captures relevant local information.

Table 2 displays the performance-comparison results for model design, including inference time, memory usage, and floating-point computation. It reveals that the model size is significantly reduced after YOLOv7-tiny incorporates the FasterNet structure and streamlines the backbone. During inference, RepBlock’s three branches are consolidated into a single-branch structure through reparameterization, significantly reducing the model’s floating-point computation. According to the data in the table, YOLO-Riny decreases the original network model size by 2 MB, cuts floating-point computation by 2 GFLOPs, and improves inference time for the entire image set by 10 ms. Due to consistent errors in the inference time measurements of the images during experiments, the test set in this paper is divided into five groups, each containing 10 images, with the average inference time calculated from these groups. The results indicate that YOLO-Riny significantly improves inference time for complex images, demonstrating a 2–3 ms speed advantage over the original network when processing multi-weed category images. This suggests that the model has robust pixel inference capabilities, making it well-suited for multi-class weed detection and providing a reliable solution for tracking a broader range of weed classes.

**Table 2.** Comparative Results of Model Design in Inference Time, Memory Usage, and Floating-Point Operations.

Model			Floating Points of Operations/G	Memory Usage /M	GPU Speed /ms
FasterNet	CARAFE	RepBlock			
×	×	×	13.1	12.9	46
√	×	×	11.9	11.4	40
√	√	×	11.6	11.6	39
√	√	√	10.1	11.2	36

Note: The model inference time refers to the time taken to infer a set of images. The test set used for model inference consists of five complex images and five simple images. Complex images contain more than seven targets and at least three species of weeds. Images that do not meet the criteria for complex images are classified as simple images. The symbol “√” indicates that improvements were made using the module, and “×” indicates that they were no.

### 3.2.2. Comparative Analysis of Multi-Model Performance

The YOLO-Riny model employed in this research is utilized to the weed-detection task and is compared with 10 other deep learning networks known for their performance in detection tasks. To ensure a fair comparison across models for the weed-detection task, the training parameters for all models are kept consistent in this research. The evaluation includes comparisons of the average mAP, memory usage, FLOPs, and inference time of each model. The results of these comparisons are presented in Table 3.

**Table 3.** Comparative analysis of multi-model performance.

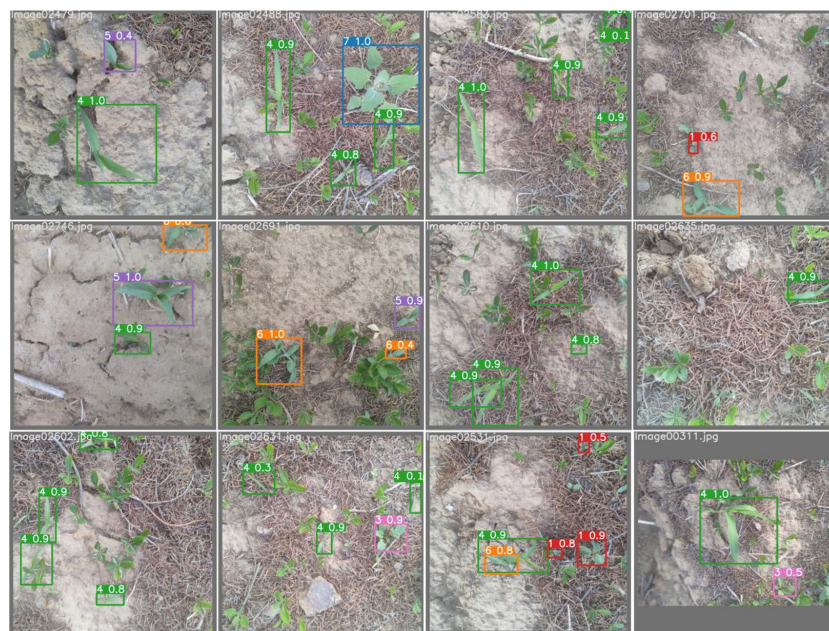
Model	mAP /%	Memory Usage /M	Floating Points of Operations/G	GPU Speed /ms
FasterRCNN	76.3	129.0	56.3	78
ShuffleNetv2	74.2	12.3	13.6	41
MobileNetV3	81.4	24.3	26.0	42
GhostNetV2	82.9	15.0	14.1	41
YOLOX	85.3	33.3	27.9	44
YOLOv5	87.3	43.2	22.4	47
YOLOv7	92.6	77.4	24.2	49
YOLOv8l	92.9	48.6	160.0	42
YOLOv8s	89.1	12.1	29.4	38
YOLO-Riny	91.7	11.2	10.1	36

As shown in Table 3, the memory sizes for the FasterRCNN, YOLOv7, and YOLOv8l models are 310 MB, 77.4 MB, and 29.4 MB, respectively. The memory usage of these models is quite high, making them challenging to adapt to devices with limited resources. Although the ShuffleNetV2, MobileNetV3, and GhostNetV2 models are lightweight, their average accuracy did not reach 85%. This indicates that these models struggled to accurately capture key feature information, particularly when distinguishing weeds against a background of *Dictamnus dasycarpus* fields. YOLOX, YOLOv8l, and YOLOv5 achieve a high mean average accuracy of 85% or more compared to other networks, but their long reasoning times make them less suitable for real-time tracking tasks. Although YOLOv8s and the model in this paper have similar data, YOLO-Riny stands out by providing high mean average accuracy while using minimal memory resources. Its inference speed also ensures stable tracker trajectories.

### 3.2.3. Analysis of Weed-Detection Visualization Experiment Results

Figure 10 presents the visualization results for YOLO-Riny’s detection performance. The green boxes in the image indicate the detected *Poa annua*. The figure illustrates that *Poa annua* often grows in overlapping clusters, and YOLO-Riny demonstrates high detection accuracy for this weed, achieving a confidence level of 0.9 or above. In the visualization, the orange and purple detection frames represent *Digitaria sanguinalis* and *Commelina communis*,

respectively. These weeds are more challenging to classify due to their similar shapes. The figure shows that several test boxes for these weeds have confidence levels lower than 0.5; however, none of them were misclassified. *Acalypha australis* is highlighted in red in the figure. Most instances of *Acalypha australis* are small, isolated targets, and the figure indicates that none of these were missed in the detection process. The self-constructed dataset, which includes various growth states of weeds, enables the network to learn detailed characteristic information for multiple weed morphologies. This comprehensive training allows the network to effectively capture the morphological differences among *Commelina communis*, *Digitaria sanguinalis*, and *Poa annua* in their various growth stages. As a result, the detection accuracy for these small-target weeds is significantly improved.



**Figure 10.** Visualization of YOLO-Riny detection results. In this figure, 1: *Acalypha australis*; 3: *Capsella bursa-pastoris*; 4: *Poa annua*; 5: *Commelina communis*; 6: *Digitaria sanguinalis*; 7: *Chenopodium album*.

Figure 11 presents the confusion matrix results [35] for YOLO-Riny. The matrix indicates that the detection accuracy for *Commelina communis* was 0.76, the lowest among the seven weed species. This lower accuracy is attributed to the morphological similarities between *Commelina communis*, *Poa annua*, and *Digitaria sanguinalis*. Additionally, the large number of *Poa annua* and the tendency of *Commelina communis* to grow in close proximity to *Poa annua* complicate the network's ability to distinguish between them effectively. *Capsella bursa-pastoris* and *Chenopodium album* demonstrated the best performance in the detection task, achieving an accuracy of 0.97 for both species. This high accuracy is due to the distinct morphological differences between these weeds and others. In contrast, as shown in the last column, *Poa annua* experienced relatively high rates of omission and false detection. This issue arises because *Poa annua* grows densely, has less distinct characteristics, and features small targets, which make it prone to detection errors.

To visually represent the effective information extracted by the YOLO-Riny model, various heatmap methods were used to achieve the results shown in Figure 12. In Figure 12, columns 1 and 4 display weed images collected on sunny days, while columns 2 and 3 show weed images collected on cloudy days. Row (B) illustrates the EigenCAM method, which is relatively stable and highlights areas that are strongly correlated with the target class information, effectively demonstrating the model's attention to different weed categories. From the figure, it is evident that YOLO-Riny efficiently captures information on fine-leaf weed categories such as *Poa annua*, *Commelina communis*, and *Digitaria sanguinalis*. The

GradCAM method in row (C) highlights the entire weed plant, showcasing the model’s ability to identify the location of different weeds within the image. However, YOLO-Riny appears to be less precise with the location of *Capsella bursa-pastoris*, focusing more on extracting location information for elongated and smaller targets. According to the LayerCAM visualization results in row (D), the model’s primary focus on the entire image is on *Commelina communis*, *Bidens pilosa*, *Digitaria sanguinalis*, and *Poa annua*. Overall, the proposed detection method demonstrates its effectiveness in accurately locating and classifying multiple species of weeds, confirming the efficacy of the research approach.

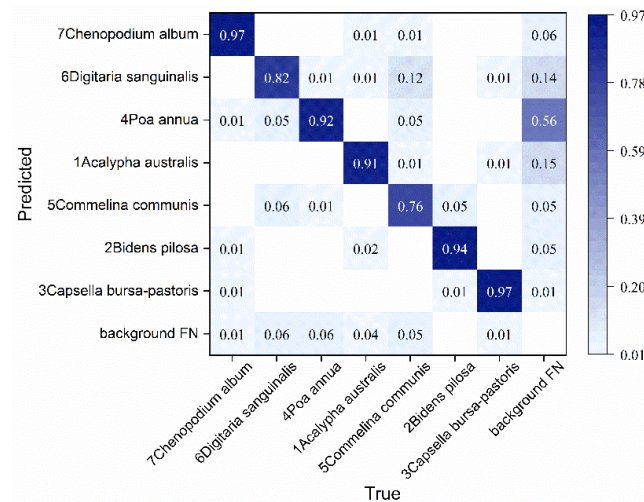


Figure 11. YOLO-Riny confusion matrix.

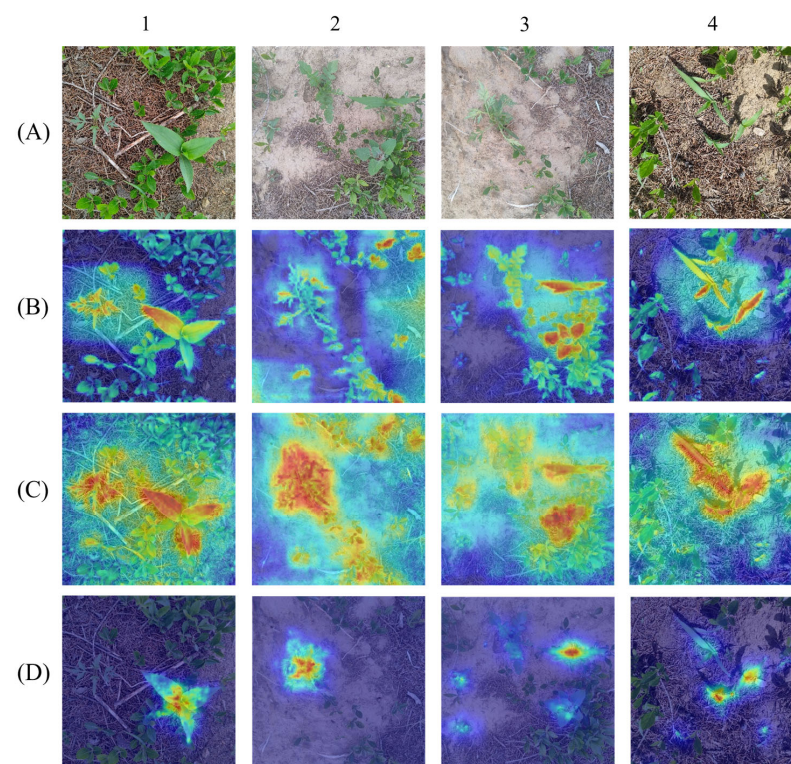
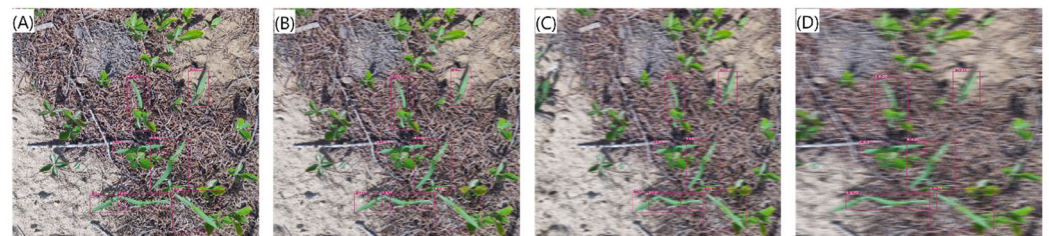


Figure 12. Multiple thermogram visualization tests. Columns 1 and 4 show weed images collected on a sunny day, while columns 2 and 3 display weed images collected on a cloudy day. Row (A) presents the original images; row (B) shows the EigenCAM visualization results; row (C) shows the GradCAM visualization results; and row (D) shows the LayerCAM visualization results. The more the model focuses on the target the closer its color is to a warm color.

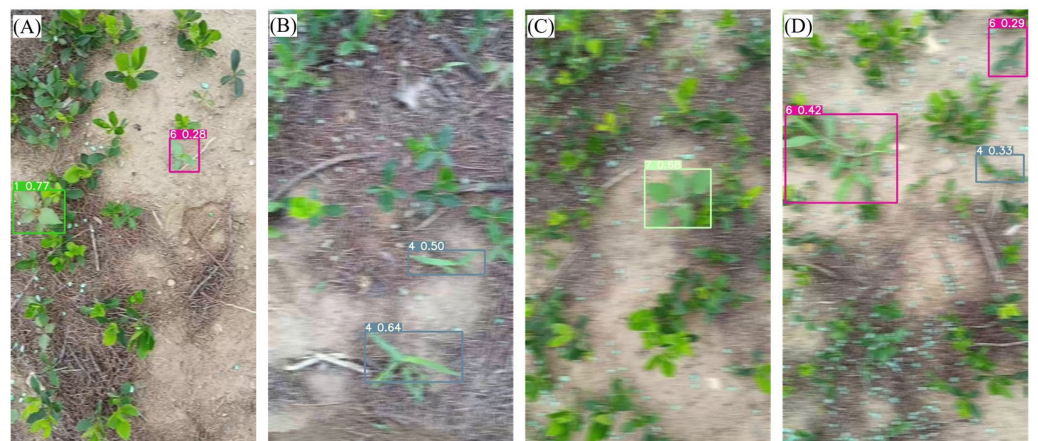


### 3.2.4. Analysis of Weed-Detection Model Robustness

To assess the robustness of the proposed model, this research employs a comparative experiment involving both simulated motion noise samples and real motion noise images from video frames, as depicted in Figures 13 and 14. Figure 13 illustrates the impact of simulated motion noise on detection, with varying levels of noise added across columns (A) to (D) ((A) represents the original image, (B) shows a 20% increase in noise, (C) a 40% increase, and (D) a 60% increase). The figure clearly demonstrates that as the noise level rises, the image progressively blurs, increasing the challenge of accurately extracting feature information. When noise increases to 20%, the model may mistakenly detect two overlapping weeds as one, but it still provides accurate location information for each weed. When noise rises to 60%, confidence drops from 0.4 to 0.03, and due to significant loss of detail, two overlapping weeds might be misclassified as one. However, the model still provides accurate overall location information for the misclassified weeds. This demonstrates that YOLO-Riny can effectively capture weed detail features and provide accurate location information despite noise interference.



**Figure 13.** Visualization of the impact of simulated motion noise tests. (A) Original image; (B) Image with 20% added blur; (C) Image with 40% added blur; (D) Image with 60% added blur.

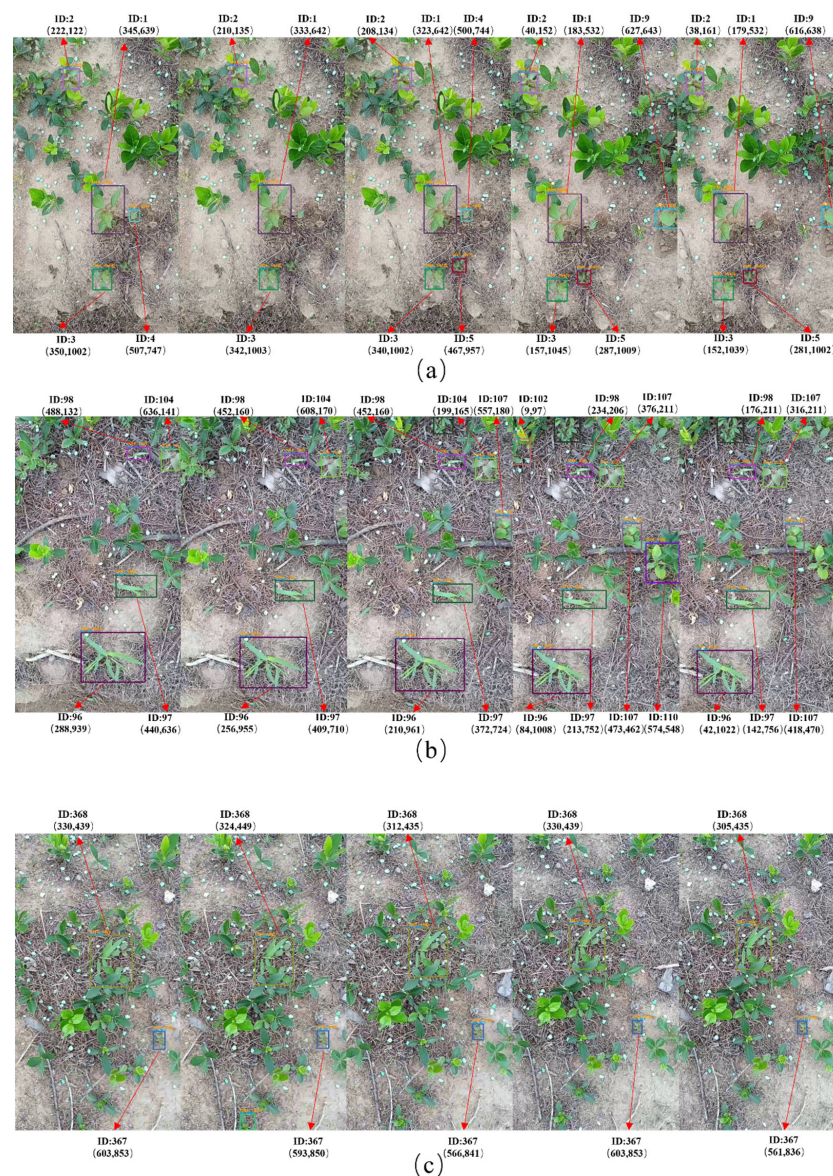


**Figure 14.** Visualization of the effects of real motion noise tests. (A) No blurring; (B) Approximate 20% blur; (C) Approximate 40% blur; (D) Approximate 60% or more blur.

As shown in Figure 14, the frames were extracted from video data collected in a field of *Dictamnus dasycarpus*, where the blurriness of the images resulted from real-world factors such as uneven shooting speed and camera noise. To assess the robustness of the model against real motion noise, researchers selected samples with four different noise levels ((A), (B), (C), (D)) based on the blur levels used in the simulation experiments. When the images are severely blurred, as illustrated in Figure 14D, YOLO-Riny detects multiple species of weeds in highly blurred images with confidence levels ranging from 0.29 to 0.42, while providing accurate location and class information. This demonstrates that although the quality and pixel resolution of the video-captured images interfere with the detection of real blurred images, YOLO-Riny can still accurately provide location and classification information under various factors, reflecting the model's high robustness to motion noise.

### 3.2.5. Results and Analysis of Weed Tracking Experiments

To visualize the localization effect on the tracked targets, this paper records the coordinates of the center point of the detection frame during the tracking trials, quantifying the exact location of each detected weed target. The tracking results of the algorithm on video sequences are shown in Figure 15, where a, b, and c represent three times segments of video clips, with each set consisting of five images extracted from a 40-frame video. In Figure 15a, weeds No. 1, No. 2, and No. 3 are successfully detected and tracked across all 40 consecutive frames of video, demonstrating the effectiveness of the algorithm. However, due to blurring from camera movement, weed No. 1 is repeatedly detected in the fourth and fifth images. Weeds No. 4 and No. 5 are affected by noise from uneven camera movement, complicating the detector’s ability to extract feature information. Weed No. 5 is matched with the trajectory in the third frame, while weed No. 4 is quickly reassigned an ID after a brief loss of trajectory. This is attributed to motion blur in the training dataset. The pre-processing of this dataset allows the detection model to better handle motion blur from camera shake, thereby minimizing transient target ID losses due to image clarity and detail degradation.



**Figure 15.** Visualization of YOLO-Riny-ByteTrack tracking performance. (a–c) represent three times segments of video clips, with each set consisting of five images extracted from a 40-frame video.

In Figure 15b, weed No. 107 gradually enters the camera’s field of view from the top of the image. It is assigned a track in the third image, and its tracking trajectory remains stable. However, Track No. 102 in the fourth image mistakenly identifies a *Dictamnus dasycarpus* seedling, which gradually moves out of the field of view from the left side. The remaining feature information interferes with the detector, causing a transient misdetection, and the track is deleted in the fifth image. This demonstrates that while the algorithm effectively detects targets entering the camera’s field of view slowly, in a medicinal herb cultivation area, residual *Dictamnus dasycarpus* seedlings at the edge of the view can cause the detector to misjudge feature information. Despite this, the matching process remains accurate.

In Figure 15c, two weeds were detected and tracked with stability. Weed No. 368 was positioned amidst multiple *Dictamnus dasycarpus* seedlings, overlapping with the surrounding seedlings in several areas. This illustrates that the algorithm demonstrated in this paper is highly robust, capable of extracting the feature information of weeds even in complex images and distinguishing them from the intricate background features.

In this paper, the ByteTrack tracking algorithm is combined with both the pre-improved and further-enhanced detection algorithms. The tracking performance is evaluated across five captured video datasets, with the results presented in Table 4. Compared to the original model, YOLO-Riny-ByteTrack shows an improvement of three percentage points in *MOTA* and a 3.4-percentage-point increase in *MOTP*. The complexity of the *Dictamnus dasycarpus* environment and the accuracy of video capture often lead to frequent target ID switching. However, employing the YOLO-Riny-ByteTrack model for tracking in this environment significantly reduces ID switching, with occurrences decreasing by a factor of 10. Consequently, the tracking results underscore that YOLO-Riny-ByteTrack exhibits outstanding stability and robustness for weed tracking in *Dictamnus dasycarpus* fields, effectively addressing practical challenges such as varying image speeds, target occlusion, and transient loss.

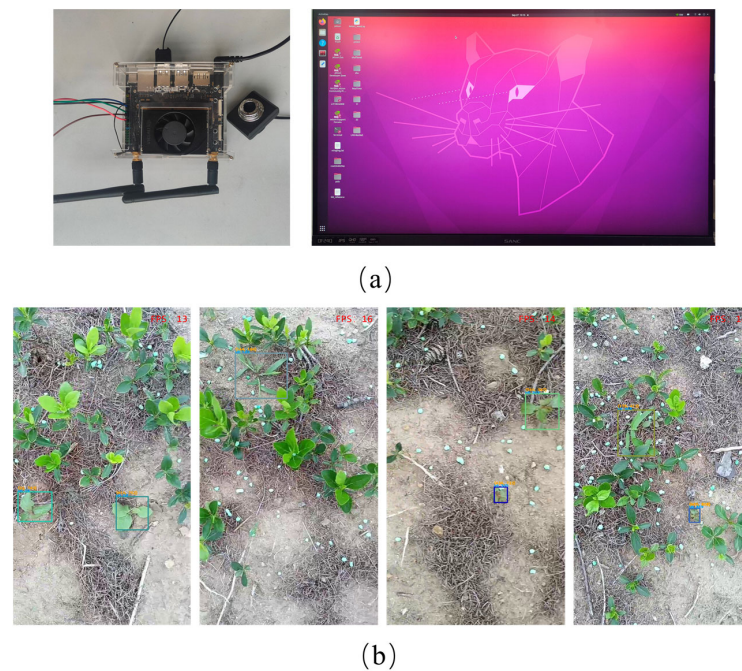
**Table 4.** Comparative analysis of tracking performance.

Proposed Algorithm	Multiple Object Tracking Accuracy%	ID Switch Rate /Times	Multiple Object Tracking Precision%
YOLOv7-tiny ByteTrack	81.4	24	74.2
YOLO-Riny-ByteTrack	84.4	10	77.6

### 3.2.6. Embedded Systems Experiment

To assess the feasibility and efficiency of deploying the proposed method on an edge device, this research involves modeling the YOLO-Riny-ByteTrack algorithm for deployment on the Jetson Orin Nano device, as illustrated in Figure 16a. The Jetson Orin Nano operates on Ubuntu 20.04.5 LTS, with the software environment configured for Python 3.8 and PyTorch 1.8.0. It is equipped with an Arm Cortex-A78AE CPU and NVIDIA Ampere architecture with 32 Tensor Cores GPU. The experiment evaluates the real-time tracking speed of YOLO-Riny-ByteTrack on the edge device, measured in frames per second (FPS), to assess the viability of the method for weed detection and tracking in *Dictamnus dasycarpus* fields. As shown in Figure 16b, YOLO-Riny-ByteTrack deployed on the Jetson Orin Nano demonstrates accurate positioning, stable tracking, and an FPS fluctuating between 13 and 16, indicating robust real-time performance.

Therefore, it can be concluded that the proposed algorithm performs stably on resource-constrained edge devices. The algorithm’s response delay is within an acceptable range, the utilization of CPU and GPU resources is reasonable, and the power consumption meets the requirements for edge devices. This provides a solid foundation and technical support for further deployment and optimization of edge-device algorithms in laser weeding equipment.



**Figure 16.** Model deployment experiments. (a) Jetson Orin Nano device; (b) Experimentation of models on embedded devices.

#### 4. Conclusions

In order to achieve weed detection and tracking in a *Dictamnus dasycarpus* field environment, a detection algorithm for seven species of weeds, namely, *Chenopodium album*, *Digitaria sanguinalis*, *Poa annua*, *Acalypha australis*, *Bidens pilosa*, *Commelina communis*, and *Capsella bursa-pastoris*, was proposed in this research. The detection algorithm consists of two main components. First, YOLO-Riny is introduced to detect weeds in *Dictamnus dasycarpus* fields. YOLO-Riny extracts features of the seven species of weeds and accurately identifies weeds even in complex environments with *Dictamnus dasycarpus* interference. Second, the ByteTrack tracking algorithm is incorporated. The YOLO-Riny-ByteTrack algorithm can detect the position and movement trajectory of weed targets in real time, enabling the identification of the same species of weeds and reducing resource waste from repeated weeding. This paper provides a preliminary investigation into algorithms for automated laser weeding in agricultural fields.

The main conclusions of this paper are as follows:

1. This research utilizes a self-constructed weed dataset specific to the *Dictamnus dasycarpus* field environment, facilitating the model to learn key weed features and enhance its robustness against the complex background of these fields. This approach mitigates the interference of crop features on weed detection.
2. By enhancing the backbone structure of the model, introducing a lightweight up-sampling operator, and applying structural adjustments such as detector head decoupling to YOLOv7-tiny through structural reparameterization, detection accuracy for *Digitaria sanguinalis* and *Acalypha australis* ultimately amounted to 5.4% and 10%, respectively. Overall precision and recall increased and ultimately amounted to 2.9% and 2.1%, respectively, with average precision ultimately amounting to 1.9%. These improvements demonstrate that the proposed method significantly enhances the model's feature perception for various species of weeds, boosts overall model generalization, and reduces issues of omission and misdetection in the tracking algorithm.
3. The YOLO-Riny detection model reduces the original network size by 2 MB and the number of floating-point operations by 2GFLOPs, leading to a 10 ms improvement in inference time for an entire batch of images. For complex images, inference time is

notably reduced, with a 2–3 ms faster performance on multi-weed category images in contrast to the original network. These improvements indicate that the proposed algorithm requires fewer computational resources, making it more suitable for deployment on resource-constrained devices. Consequently, it facilitates timely updates of target positions and bounding box information, while minimizing ID transformations in complex scenes.

4. To minimize the cost associated with repeatedly removing the same weed during laser weeding, this paper combines the improved YOLO-Riny detection algorithm with the parameterized ByteTrack algorithm to track and mark seven species of weeds in *Dictamnus dasycarpus* fields. The YOLO-Riny-ByteTrack model achieved a 3-percentage-point increase in *MOTA* and a 3.4-percentage-point increase in *MOTP*, while reducing the number of ID switches by a factor of 10 compared to the original model. This combination enhances target consistency across consecutive frames, improves tracking accuracy and stability, and consequently reduces resource and time waste in laser weeding.

The YOLO-Riny-ByteTrack model proposed in this paper still possesses deficiencies as well as room for enhancement. The detection algorithm part of YOLO-Riny-ByteTrack can be adapted accordingly in the future based on the performance of YOLO after iteration and the stability of combining with the tracking algorithm. In the future, the YOLO-Riny-ByteTrack model will be further optimized to solve the problem of false detection caused by targets entering and leaving the camera field of view. In addition, the model's laser weeding system as well as the matching and control of the laser equipment will be implemented one by one in future work and research and development.

**Author Contributions:** Conceptualization, Y.X. and Z.L.; Methodology, Y.X. and Z.L.; Validation, Z.L., J.L. and Y.Z.; Formal analysis, Z.L. and Y.Z.; Data curation, D.H. and Y.C.; Writing—original draft preparation, Z.L.; Writing—review and editing, Y.X., J.L. and Y.Z.; Supervision, Y.Z., D.H., and Y.C.; Project administration, Y.X. and Y.C.; Funding acquisition, Y.Z. and Y.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the Jilin Provincial Department of Education-Science and Technology Project (JJKH20230382KJ), the Jilin Provincial Department of Science and Technology-Free Exploration Basic Research (YDZJ202301ZYTS408), and the Jilin Provincial Department of Science and Technology-Key Research and Development (20230202035NC).

**Data Availability Statement:** The original contributions presented in the research are included in the article, further inquiries can be directed to the corresponding author.

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

## References

1. Hu, X.; Zhang, W.; Zhu, Q. *Zhonghua Bencao*; Shanghai Science and Technology Publications: Shanghai, China, 1998; pp. 225–226.
2. Liang, J.R.; Shi, J.J.; Yuan, W.H.; Zhang, Y.; Ding, C.H. First Report of Chinese Medicinal Plant *Dictamnus dasycarpus* Leaf Spot Disease Caused by *Fusarium scirpi* in China. *Plant Dis.* **2024**, *108*, 2232. [[CrossRef](#)]
3. Wang, Z.; Xu, F.; An, S. Chemical constituents from the root bark of *Dictamnus dasycarpus* Turcz. *China J. Chin. Mater. Medica* **1992**, *17*, 551–576.
4. Deng, Z.; Wang, T.; Zheng, Y.; Zhang, W.; Yun, Y.-H. Deep learning in food authenticity: Recent advances and future trends. *Trends Food Sci. Technol.* **2024**, *144*, 104344. [[CrossRef](#)]
5. Mesnage, R.; Székács, A.; Zaller, J.G. Herbicides: Brief History, Agricultural Use, and Potential Alternatives for Weed Control. In *Herbicides*; Mesnage, R., Zaller, J.G., Eds.; Elsevier: Amsterdam, The Netherlands, 2021; pp. 1–20.
6. Jin, X.; Liu, T.; Yang, Z.; Xie, J.; Bagavathiannan, M.; Hong, X.; Xu, Z.; Chen, X.; Yu, J.; Chen, Y. Precision weed control using a smart sprayer in dormant bermudagrass turf. *Crop Prot.* **2023**, *172*, 106302. [[CrossRef](#)]
7. Sharma, V.; Tripathi, A.K.; Mittal, H. Technological revolutions in smart farming: Current trends, challenges & future directions. *Comput. Electron. Agric.* **2022**, *201*, 107217. [[CrossRef](#)]
8. Mwitwa, C.; Rains, G.C.; Prostko, E. Evaluation of Diode Laser Treatments to Manage Weeds in Row Crops. *Agronomy* **2022**, *12*, 2681. [[CrossRef](#)]
9. Yu, K.; Ren, J.; Zhao, Y. Principles, developments and applications of laser-induced breakdown spectroscopy in agriculture: A review. *Artif. Intell. Agric.* **2020**, *4*, 127–139. [[CrossRef](#)]

10. Upadhyay, A.; Zhang, Y.; Koparan, C.; Rai, N.; Howatt, K.; Bajwa, S.; Sun, X. Advances in ground robotic technologies for site-specific weed management in precision agriculture: A review. *Comput. Electron. Agric.* **2024**, *225*, 109363. [[CrossRef](#)]
11. Quan, L.; Jiang, W.; Li, H.; Li, H.; Wang, Q.; Chen, L. Intelligent intra-row robotic weeding system combining deep learning technology with a targeted weeding mode. *Biosyst. Eng.* **2022**, *216*, 13–31. [[CrossRef](#)]
12. Zhu, H.; Zhang, Y.; Mu, D.; Bai, L.; Wu, X.; Zhuang, H.; Li, H. Research on improved YOLOx weed detection based on lightweight attention module. *Crop Prot.* **2024**, *177*, 106563. [[CrossRef](#)]
13. Guo, W.; Qiao, S.; Zhao, C.; Zhang, T. Defect detection for industrial neutron radiographic images based on modified YOLO network. *Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrometers Detect. Assoc. Equip.* **2023**, *1056*, 168694. [[CrossRef](#)]
14. Zhu, H.B.; Zhang, Y.Y.; Mu, D.L.; Bai, L.Z.; Zhuang, H.; Li, H. YOLOX-based blue laser weeding robot in corn field. *Front. Plant Sci.* **2022**, *13*, 1017803. [[CrossRef](#)] [[PubMed](#)]
15. Mwitwa, C.J. Development of the Autonomous Diode Laser Weeding Robot. Ph.D. Thesis, University of Georgia, Athens, GA, USA, 2023.
16. Liu, S.; Jin, Y.; Ruan, Z.; Ma, Z.; Gao, R.; Su, Z. Real-Time Detection of Seedling Maize Weeds in Sustainable Agriculture. *Sustainability* **2022**, *14*, 15088. [[CrossRef](#)]
17. Shao, Y.; Guan, X.; Xuan, G.; Gao, F.; Feng, W.; Gao, G.; Wang, Q.; Huang, X.; Li, J. GTCBS-YOLOv5s: A lightweight model for weed species identification in paddy fields. *Comput. Electron. Agric.* **2023**, *215*, 108461. [[CrossRef](#)]
18. Peng, H.; Li, Z.; Zhou, Z.; Shao, Y. Weed detection in paddy field using an improved RetinaNet network. *Comput. Electron. Agric.* **2022**, *199*, 107179. [[CrossRef](#)]
19. Toliás, A.; Papanicolaou, G.C.; Alexandropoulos, D. Fabrication of glass to PLA joints with an intermediate aluminum layer by using low-cost industrial nanosecond IR fiber lasers. *Opt. Laser Technol.* **2024**, *175*, 110811. [[CrossRef](#)]
20. Kuantama, E.; Zhang, Y.; Rahman, F.; Han, R.; Dawes, J.; Mildren, R.; Abir, T.A.; Nguyen, P. Laser-based drone vision disruption with a real-time tracking system for privacy preservation. *Expert Syst. Appl.* **2024**, *255*, 124626. [[CrossRef](#)]
21. Liu, Y.; Li, Y. Positioning accuracy improvement for target point tracking of robots based on Extended Kalman Filter with an optical tracking system. *Robot. Auton. Syst.* **2024**, *179*, 104751. [[CrossRef](#)]
22. Chai, J.; He, S.; Shin, H.-S.; Tsourdos, A. Domain-knowledge-aided airborne ground moving targets tracking. *Aerosp. Sci. Technol.* **2024**, *144*, 108807. [[CrossRef](#)]
23. Su, Y.; Cheng, T.; He, Z. Collaborative trajectory planning and transmit resource scheduling for multiple target tracking in distributed radar network system with GTAR. *Signal Process.* **2024**, *223*, 109550. [[CrossRef](#)]
24. Chen, G.; Xu, Y.; Yang, X.; Hu, H.; Cheng, H.; Zhu, L.; Zhang, J.; Shi, J.; Chai, X. Target tracking control of a bionic mantis shrimp robot with closed-loop central pattern generators. *Ocean. Eng.* **2024**, *297*, 116963. [[CrossRef](#)]
25. Wang, Z.; Walsh, K.; Koirala, A. Mango Fruit Load Estimation Using a Video Based MangoYOLO—Kalman Filter—Hungarian Algorithm Method. *Sensors* **2019**, *19*, 2742. [[CrossRef](#)] [[PubMed](#)]
26. Li, X.; Wang, X.; Ong, P.; Yi, Z.; Ding, L.; Han, C. Fast Recognition and Counting Method of Dragon Fruit Flowers and Fruits Based on Video Stream. *Sensors* **2023**, *23*, 8444. [[CrossRef](#)]
27. Özlüoymak, Ö. Design and development of a servo-controlled target-oriented robotic micro-dose spraying system in precision weed control. *Semin.-Cienc. Agrar.* **2021**, *42*, 635–656. [[CrossRef](#)]
28. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. ByteTrack: Multi-Object Tracking by Associating Every Detection Box. In Proceedings of the European Conference on Computer Vision, Montreal, QC, Canada, 11 October 2021.
29. Li, X.; Orchard, M.T. New edge-directed interpolation. *IEEE Trans. Image Process.* **2001**, *10*, 1521–1527. [[CrossRef](#)]
30. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
31. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. CARAFE: Content-Aware ReAssembly of FEatures. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3007–3016.
32. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-Style ConvNets Great Again. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13728–13737.
33. Chen, J.; Kao, S.-h.; He, H.; Zhuo, W.; Wen, S.; Lee, C.-H.; Chan, S.-H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
34. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.
35. Wang, Y.; Jia, Y.; Tian, Y.; Xiao, J. Deep reinforcement learning with the confusion-matrix-based dynamic reward function for customer credit scoring. *Expert Syst. Appl.* **2022**, *200*, 117013. [[CrossRef](#)]

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