remote sensing

MDPI

Article

A Combined Strategy for Detecting Cotton Seedlings in UAV RGB Images Based on GLI Automatic Annotation and Hybrid Architecture Deep Learning

Jinghe Wang, Qingfu Liang, Zaihong Yang, Xinjun Wang *, Haoran Yang, Panfeng Wang, Shuhan Huang, Hanyu Cui, Chenyu Li and Pingan Jiang

Xinjiang Key Laboratory of Soil and Plant Ecological Processes, Xinjiang Engineering Technology Research Center of Soil Big Data, College of Resources and Environment, Xinjiang Agricultural University, Urumqi 830052, China
* Correspondence: wangxj@xjau.edu.cn

Highlights

What are the main findings?

- GLI automatic annotation achieved an F1 score of 96.18% and a mean IoU of 89.78%.
- In source-region testing (means across three random seeds), Mamba-YOLO-B and RT-DETR-L achieved the highest mAP@0.5:0.95 values (0.8020 and 0.7928, respectively). Mamba-YOLO-B + GLI performed best in cross-region testing.

What are the implications of the main findings?

- GLI reduces automatic annotation costs, but manual review remains necessary to ensure final accuracy.
- Compared with the other joint strategies, Mamba-YOLO-B + GLI achieved the highest performance in seedling emergence detection in source-region and cross-region tests.

Abstract

Accurate detection of cotton seedlings is essential for monitoring crop establishment. However, manual annotation is time-consuming, and the quality of conventional automatic annotation methods is affected by illumination and background conditions. This study proposes a joint strategy combining green leaf index (GLI) automatic annotation with deep learning detection. Three annotation methods were considered: GLI, excess green minus excess red (ExG–ExR), and hue–saturation–value (HSV). Their nine combinations with three detectors (YOLOv10n, RT-DETR-L, and Mamba-YOLO-B) were evaluated using spatial data partitioning and independent cross-region testing. Under identical post-processing, GLI achieved the highest annotation quality, with an F1 score of 96.18% and a mean intersection over union (IoU) of 89.78% for matched boxes. For initial box drawing only, bounding boxes for 200 targets were generated in 0.399 s, compared with approximately 600 s for manual drawing. Detection accuracy was assessed using mean average precision (mAP) at IoU thresholds from 0.50 to 0.95 (mAP@0.5:0.95). Across three random seeds, Mamba-YOLO + GLI achieved the highest mean source-region mAP@0.5:0.95, at 0.8020. In cross-region testing, Mamba-YOLO + GLI achieved the best detection and counting performance, with an mAP@0.5:0.95 of 0.3446. However, its detection accuracy remained substantially lower than in source-region testing. These results demonstrate that GLI automatic annotation can provide effective training labels for cotton seedling detection. Among the evaluated combinations, Mamba-YOLO + GLI achieved the highest mean source-region mAP@0.5:0.95 and the highest cross-region mAP@0.5:0.95.

Check for updates

Academic Editors: Renan Falcioni, Renato Herrig Furlanetto and Luís Guilherme Teixeira Crusiol

Received: 8 September 2026
Revised: 12 November 2026
Accepted: 13 November 2026
Published: 19 November 2026

Copyright: © 2026 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.

Keywords: cotton; seedling detection; automatic annotation; unmanned aerial vehicle (UAV); hybrid architecture

1. Introduction

Cotton is a major strategic cash crop with global significance. Its yield and quality directly affect the economic security of agricultural sectors and the development of the textile industry, which makes cotton cultivation a key part of national economic development [1]. Emergence rate is not only a biological indicator of seed germination but also a core agronomic factor that shapes final yield [2]. Keeping track of emergence rates provides baseline data for deciding whether to replant during the seedling stage, and it also helps guide water and fertilizer management as well as yield predictions [3]. Traditional manual surveys can achieve relatively high detection accuracy in small survey areas. However, for large-scale field surveys, the method becomes inefficient and labor-intensive [4]. Recent advances in UAV technology have made it possible to collect agricultural data efficiently at low altitudes. This approach offers a viable solution for monitoring cotton emergence across large areas [5]. Given the cost of specialized sensors, RGB images captured by UAVs are particularly well suited for detecting crop seedlings [6,7]. For this reason, developing intelligent methods for cotton seedling detection using UAV RGB images is an important step toward digital management in cotton production.

Significant progress has been made in a number of studies that evaluate crop emergence rates using UAV images [8,9]. Based on how features are extracted, these approaches fall into two categories. The first category relies mainly on manually selected features. These methods typically build on empirical knowledge and use multispectral or RGB imagery to manually pick out spectral, color, and texture features of vegetation as indicators for assessing emergence rates. For instance, Li et al. [10] used vegetation index features to improve the accuracy of estimating cotton seedling emergence during the early growth stage. Similarly, Li et al. [11] extracted color and texture features from UAV RGB images to refine emergence rate estimates for potatoes. However, it is difficult to adapt the design of features based only on prior knowledge when the crop types are quite different, the varieties are quite different, or the soil conditions are spatially heterogeneous [12]. It implies that the manual design of data features depends on the prior knowledge of researchers. The second category is deep learning methods, which extract features from data automatically during model training. These are methods that exploit the ability of neural networks to learn from large labeled datasets, without the need to manually specify rules based on prior knowledge. Liu et al. [13] proposed a deep learning approach for quantitative evaluation of maize emergence based on UAV imagery. Lin et al. [14] employed an improved YOLOv5s algorithm to observe and count peanut seedlings. These methods require less expert knowledge and effectively deal with high-dimensional, nonlinear, large-scale data [15]. However, the receptive field of the convolutional kernels of a single architecture is limited, especially in shallow layers. This makes it challenging to capture global contextual relations between distant pixels. Such global information is often needed for detecting small objects. To address the limitations of a single architecture, Xie et al. [16] applied a CNN–Transformer hybrid network to cropland parcel mapping, demonstrating that global context helps delineate complex agricultural landscape boundaries. This integration of local features and global perception provides an important technical reference for cotton seedling detection in this study.

Improving the performance of deep learning models usually requires a substantial amount of labeled sample data [17]. Image annotation is still performed mainly by hand, making it slow and labor-intensive. These constraints limit the scale and quality of la-

beled datasets and consequently affect model performance [18]. To address this issue, conventional image processing methods have been explored for automatic annotation in various crop scenarios. These methods can be broadly divided into two categories. The first is based on RGB vegetation indices. Differences in the responses of green vegetation across the red, green, and blue channels are used to calculate vegetation indices, which are then combined with threshold segmentation. Zhao et al. [19] used indices such as ExG and ExG–ExR to estimate maize seedling numbers. Otsu thresholding and morphological operations were incorporated to improve annotation accuracy. Dhakal et al. [20] found that the ExG–ExR vegetation index worked best for annotating emergence rates in cotton and sorghum samples. These methods require only a low-cost visible-light camera and simple computation, which has supported their widespread use. However, RGB channel values are directly affected by illumination intensity. Difference vegetation indices such as ExG–ExR enhance vegetation signals through linear channel differences, but their responses may still be affected by illumination and camera processing. In shaded areas or bright backgrounds, illumination changes alter RGB values. ExG–ExR may not fully compensate for these effects, resulting in increased false detections.The second type utilizes the HSV color space. Hamuda et al. [21] used HSV with thresholding for annotating cauliflower samples in complex conditions. Bouhadjer et al. [22] combined deep learning-based crop segmentation with HSV-based greenness analysis for weed detection. Color separation between seedlings and background may also vary with growth status and illumination. Here, HSV color space was used to construct a baseline, whose applicability was assessed through box-level evaluation under shared post-processing.

In summary, the accuracy of current crop seedling detection based on UAV remote sensing images is mainly limited by two key issues: the reliability of sample annotation and the model's ability to grasp seedling information. Traditional automatic annotation methods (such as those based on color features or differential vegetation indices) require a high and stable contrast between seedlings and their background environment (e.g., soil). However, in actual farmland, environmental stresses often lead to uneven emergence and weak seedlings. Their green characteristics become less distinct, and spectral responses diminish, significantly reducing the distinguishability between seedlings and the background. This results in numerous missed detections by fixed-threshold-based automatic annotation methods. At the model level, YOLOv10 reduces reliance on hand-crafted features compared with conventional approaches. However, its core remains predominantly convolutional, with self-attention introduced only to some channels through the PSA module; convolution and self-attention jointly model features. In complex agricultural environments, seedling detection requires pixel-level analysis and may also benefit from spatial distribution patterns. Therefore, an automatic annotation method for UAV RGB images, which is adaptable to complex field environments, is of significance. This method should be combined with a joint strategy that integrates hybrid deep learning models (e.g., RT-DETR [23], Mamba-YOLO [24]) that can fuse the local seedling features with global contextual information. It is crucial to address these issues so as to improve the accuracy of seedling detection.

Based on this, the study puts forward a strategy that combines an automatic annotation method using the visible light band ratio vegetation index (GLI) with a hybrid architecture model. The idea is to use GLI-based automatic annotation to address issues and collect a large amount of high-quality labeled data. Then, that data is used to train three detection models with different kinds of architectures. The main objectives are threefold.

(1) Compare the annotation accuracy and efficiency of GLI, manual annotation, ExG–ExR, and HSV.

(2) Compare the accuracy of nine annotation-model joint strategies and the performance of YOLOv10, RT-DETR, and Mamba-YOLO under different labels and controlled perturbations (illumination, resolution, etc.).
(3) Compare cross-region generalization and image-level counting errors to identify the optimal combination.

This study combines GLI automatic annotation with deep learning detectors to compare cotton seedling detection and counting using different label sources and model configurations. Photometric perturbations, label perturbations, and cross-region testing are used to examine applicability. The approach rapidly generates initial labels for manual review and provides a reference for cotton seedling detection and counting in UAV RGB images.

2. Materials and Methods

2.1. Overall Research Framework

Figure 1 presents the study workflow. The workflow comprises study design, data preparation, experiments, and performance evaluation. Source-region imagery was preprocessed and spatially partitioned into training, validation, and test subsets, with buffer-zone images excluded. Automatic training labels were generated using GLI, ExG–ExR, and HSV through feature calculation, thresholding, and filtering. These three label sets were paired with YOLOv10n, RT-DETR-L, and Mamba-YOLO-B to form nine combinations. Manual reference labels were used for validation and testing.

Experiments assessed annotation quality and efficiency, detector performance and resource requirements, training-label omission, and counting performance. Cross-region evaluation used source-trained models on image tiles from the target region without target-domain fine-tuning. Detection, localization, and counting metrics were used to compare the strategies.

2.2. Study Area

The field experiment was conducted at the Huaxing Farm experimental base in Changji City, Changji Hui Autonomous Prefecture, Xinjiang, China (44°12′N, 87°17′E; elevation: 443 m; Figure 2). Regional studies have documented the responses of cotton growth and yield to hydrothermal conditions in Xinjiang [25], and irrigation scheduling in cotton fields at Huaxing Farm has also been investigated [26]. A published study at the same farm reported a mean annual temperature of approximately 6.8 °C, annual precipitation of approximately 190 mm, and annual sunshine duration of approximately 2700 h [27]. These values describe the climatic background rather than meteorological observations during the present image acquisition.

The experimental field covered 3200 m^2 (80 m long and 40 m wide) and was planted with the cultivar Zhongmian 113. Mulched drip irrigation was implemented using one plastic film strip covering three drip tapes and six cotton rows. Plant spacing was approximately 8–10 cm, with wide and narrow row spacings of 66 and 10 cm, respectively (Figure 3). Sowing took place on 28 April 2024, and images were acquired on 15 May, 17 days after sowing, when seedlings were at the 2–4-leaf stage (including cotyledons).

Image acquisition covered field locations with different seedling densities, including low-density areas with substantial gaps, medium-density areas with normal emergence, and high-density areas with densely distributed seedlings. These scenarios describe image variation within the same experimental field and do not represent independent replicates across regions or cultivars.

The cross-region test area was located in Kuqa City, Aksu Prefecture, southern Xinjiang, and was also planted with Zhongmian 113. Sowing and image acquisition occurred on

9 and 26 April 2026, respectively. As in the source region, images were acquired 17 days after sowing at the 2–4-leaf stage (including cotyledons).

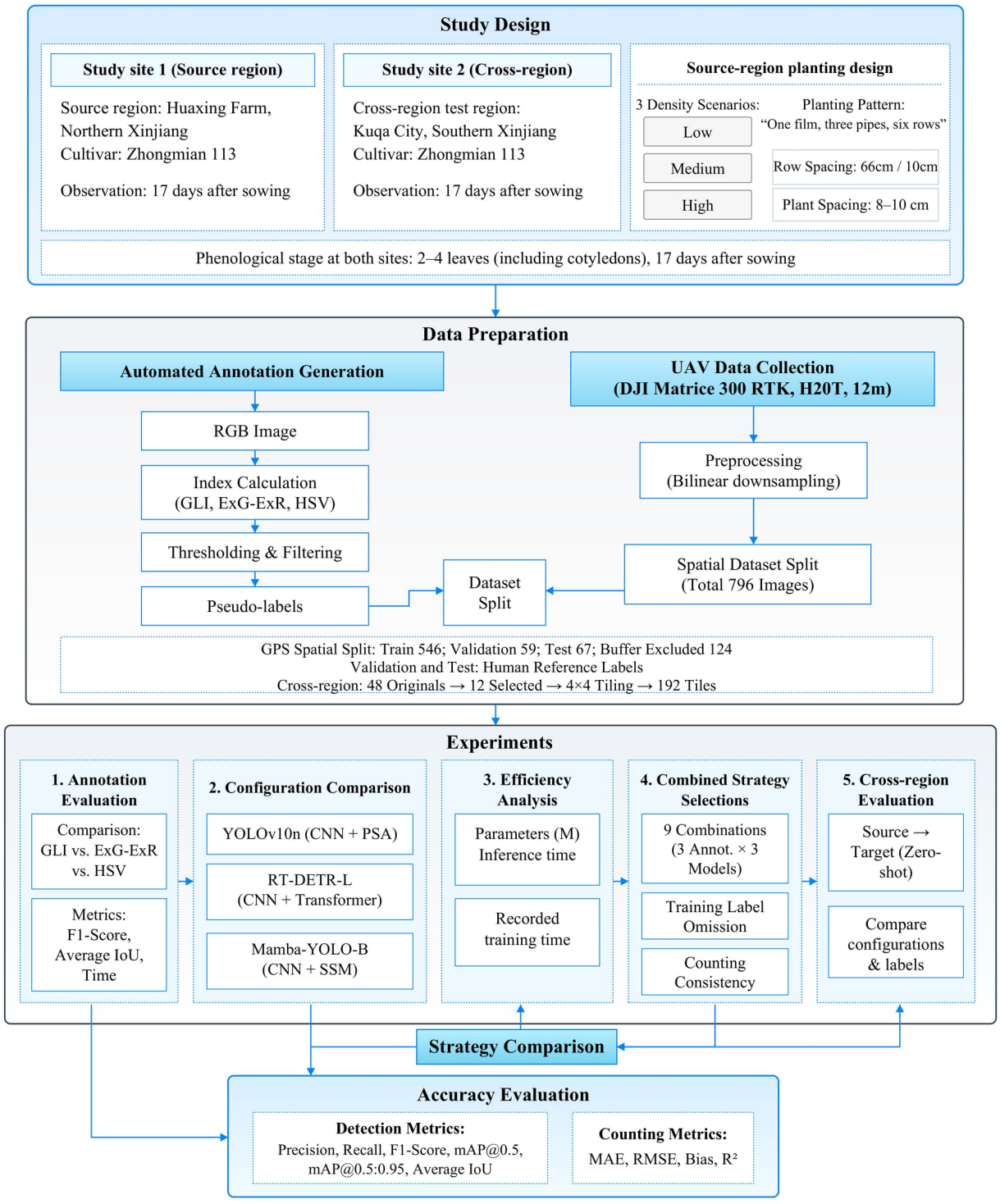


Figure 1. Workflow of the combined strategy for cotton seedling detection and counting using UAV RGB imagery. Abbreviations: UAV, unmanned aerial vehicle; RGB, red–green–blue; GLI, green leaf index; ExG–ExR, excess green minus excess red; HSV, hue–saturation–value; GPS, Global Positioning System; CNN, convolutional neural network; PSA, partial self-attention; SSM, state-space model; mAP, mean average precision; IoU, intersection over union; MAE, mean absolute error; RMSE, root mean square error; Bias, mean signed counting error; R^2, coefficient of determination.

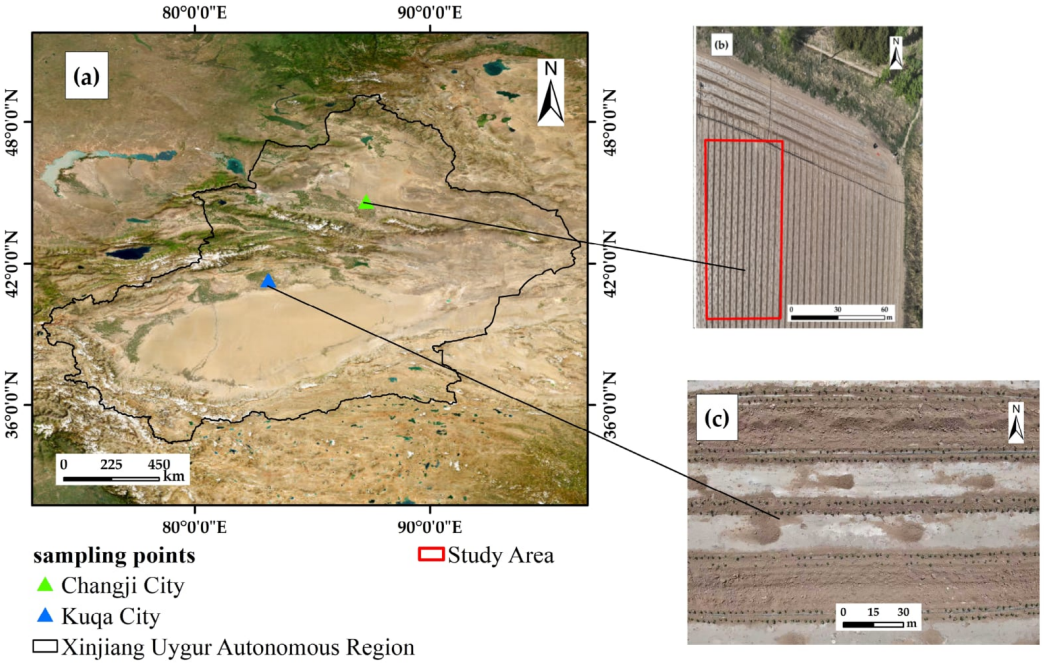


Figure 2. (**a**) Study area location; administrative boundaries were obtained from the Standard Map Service of the Ministry of Natural Resources of China (Tianditu), map approval number GS(2024)0605. (**b**) Source region experimental field in Huaxing. (**c**) Cross-region experimental field in Kuqa City.

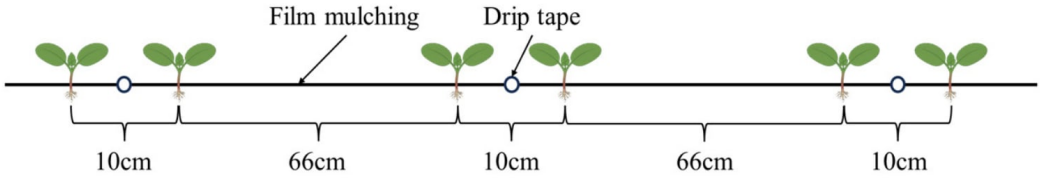


Figure 3. Cotton planting layout with one plastic film strip, three drip tapes, and six rows.

2.3. Data Acquisition

Source-region images were acquired using a DJI Matrice 300 RTK UAV equipped with a DJI Zenmuse H20T visible-light zoom camera (DJI, Shenzhen, China). Camera specifications are provided in Table 1, and acquisition conditions for both regions are summarized in Table 2. The flight altitude was 12 m, with 3.7× optical zoom (117.0 mm equivalent focal length; 25.4 mm physical focal length) and 80% forward and side overlap. A total of 796 RGB images of 5184 × 3888 pixels were acquired. Because of computational resource constraints, source-region images were downsampled to 2592 × 1944 pixels using bilinear interpolation, retaining the 4:3 aspect ratio.

Table 1. Specifications of the DJI Zenmuse H20T visible-light zoom camera.

Parameter	Value
Sensor Type	1/1.7″ CMOS
Physical Focal Length	6.83–119.94 mm
Equivalent Focal Length	31.7–556.2 mm
Digital Field of View	66.6°–4°
Aperture	f/2.8–f/11
Effective Sensor Pixels	20 million
Maximum Photo Size	5184 × 3888 pixels

Table 2. Acquisition and image parameters for the source and cross-region datasets.

Parameter	Source Region: Huaxing Farm	Cross-Region Test Area: Kuqa City
UAV/sensor	DJI Matrice 300 RTK (DJI, Shenzhen, China)/Zenmuse H20T visible-light zoom camera	DJI Mavic 3T visible-light camera (DJI, Shenzhen, China)
Flight altitude	12 m	Approximately 3 m
Zoom/aperture	3.7× optical zoom; focal length 25.4 mm (117.0 mm equivalent)	1× zoom; f/2.8
Forward/side overlap	80%/80%	Not recorded
Original image dimensions (columns × rows)	5184 × 3888	8000 × 6000
Analysis image dimensions (columns × rows)	2592 × 1944 (bilinear downsampling)	2000 × 1500 (non-overlapping tiling)
Original/analysis GSD (cm/pixel)	Approximately 0.068/0.135	Approximately 0.055/0.055
Number of images	796 original images	12 original images, 192 tiles

Cross-region test images were acquired in Kuqa City on 26 April 2026 using a DJI Mavic 3T at approximately 3 m altitude, 1× zoom, and f/2.8. Based on GPS coordinates and spatial coverage, overlap assessment of the 48 original images left 12 RGB images with mutually non-overlapping spatial coverage; all 12 were used for cross-region testing. Each image was divided into a non-overlapping 4 × 4 grid, yielding 16 tiles of 2000 × 1500 pixels and 192 tiles in total. Cropping did not alter the ground sampling distance.

The estimated source-region coverage per image was approximately 3.48–3.52 m × 2.61–2.64 m. The ground sampling distances (GSDs) of the original and twofold-downsampled source-region images were approximately 0.068 and 0.135 cm/pixel, respectively. The original cross-region GSD was approximately 0.055 cm/pixel, and tiling preserved the native GSD.

The source-region imagery covered low-, medium-, and high-density seedling scenarios, as illustrated in Figure 4.



Figure 4. UAV RGB images showing different seedling densities 17 days after sowing: (**a**) high; (**b**) medium; (**c**) low. The upper panels show the original image views. The lower panels provide enlarged views of the seedlings within the red boxes in the corresponding upper panels.

2.4. Automatic Annotation Methods

2.4.1. Manual Annotation

Cotton seedlings were manually annotated with bounding boxes using LabelImg, providing a reference for evaluating automatic annotation quality and model performance. Manual labels were used in the relevant experiments involving S_val, S_test, E_test, and A_ref; the datasets and their uses are described in Section 2.6.1.

To compare annotation quality, three automatic methods (GLI, ExG–ExR, and HSV) were evaluated on the 98 images in A_ref. All methods used identical post-processing (Table 3), differing only in their initial feature representations: GLI and ExG–ExR used their respective index maps, whereas HSV used the H, S, and V channels. The comparison therefore applies only to the HSV implementation used in this shared pipeline. Manual annotations served as ground truth, and evaluation metrics included F1, mean IoU, and the time required to annotate 200 seedlings.

Table 3. Automatic annotation methods and control settings.

Category	Method	Description
Manual Annotation	Manual Annotation	Reference for evaluating automatic annotation accuracy
Automatic annotation	ExG–ExR	Difference Vegetation Index Method
	HSV	Color space transformation method
	GLI	Ratio-based vegetation index

2.4.2. Conventional Automatic Annotation

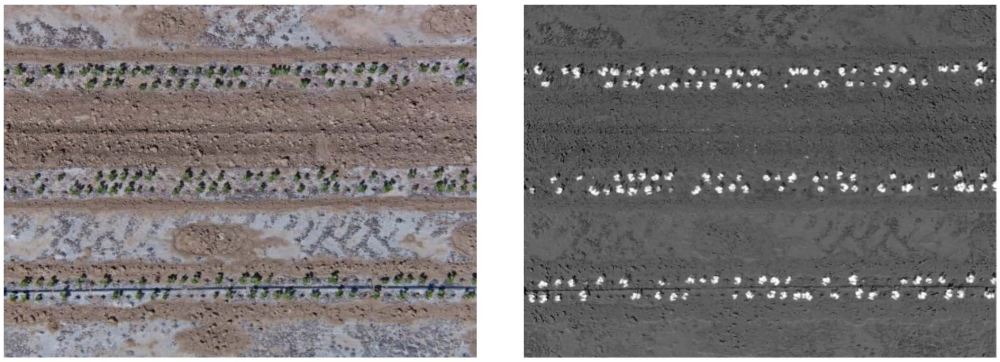
(1) ExG–ExR feature construction

ExG–ExR enhances the separation between green vegetation and the background through a weighted difference in visible-light channels, as follows:

$$\mathrm{ExG} - \mathrm{ExR} = \frac{3 \times G - 2.4 \times R - B}{G + R + B + 10^{-7}} \quad (1)$$

Here, R, G, and B denote pixel values in the red, green, and blue channels, respectively. The denominator $R + G + B$ normalizes the weighted difference against overall brightness variations, and a constant of 10^{-7} is added to avoid division by zero. The calculated index was transformed using $((\mathrm{ExG–ExR} + 1)/2) \times 255$ and then converted to uint8 for subsequent image-wise Otsu thresholding. Because the index requires only visible-light imagery and has a low computational cost, it can be used for crop extraction.

Figure 5 illustrates the feature responses of the three methods.

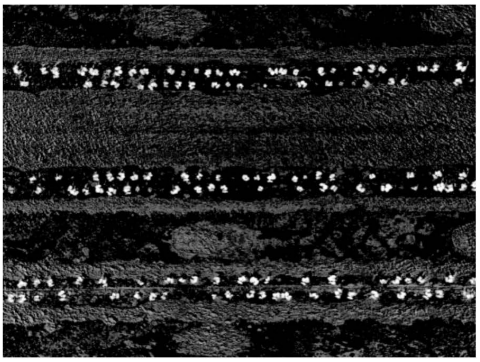


(**a**) Original RGB Image (**b**) GLI Features

Figure 5. *Cont.*



(**c**) ExG − ExR Features



(**d**) HSV-derived Feature

Figure 5. Original RGB image and corresponding feature representations of cotton seedlings: (**a**) original RGB image; (**b**) GLI feature map; (**c**) ExG–ExR feature map; and (**d**) continuous HSV-derived feature map. Grayscale intensities represent the mapped feature values, with brighter pixels indicating higher values. The HSV-derived map is a composite feature calculated based on hue, saturation, and value, rather than a single HSV channel or a binary segmentation mask.

(2) HSV feature construction

The HSV color space represents image color and brightness through hue (H), saturation (S), and value (V). Hamuda et al. [21] combined HSV-based processing with morphological operations for crop extraction. For visualization in Figure 5, a continuous HSV-derived feature map was constructed by combining a hue-based greenness score with normalized saturation and value. Here, H, S, and V are the channel values obtained using OpenCV. The resulting feature was then converted to an unsigned 8-bit image for display.

2.4.3. GLI Automatic Annotation

GLI is a ratio-based vegetation index derived from RGB channels and is calculated as follows:

$$\mathrm{GLI} = \frac{2 \times G - R - B}{2 \times G + R + B + 10^{-7}} \tag{2}$$

Here, R, G, and B denote the red, green, and blue pixel values, respectively. RGB values in the range 0–255 were converted to float32 without chromatic normalization. A constant of 10^{-7} was added to the denominator to avoid division by zero. GLI was mapped using $(\mathrm{GLI} + 1)/2 \times 255$, retaining negative values in the mapping. The mapped values were converted to uint8 before image-wise Otsu thresholding. Although GLI has been used to estimate vegetation cover at broader scales [28], a common multiplicative RGB factor cancels pointwise in the ideal GLI ratio, even when the factor varies spatially. Responses to controlled photometric perturbations are reported in Section 3.3.2. Figure 6 provides a workflow example only.

2.4.4. Shared Post-Processing and Candidate Box Generation

After feature construction and binarization, all three methods used identical post-processing: erosion with a disk(1) structuring element, extraction of 8-connected components, retention of regions with an area ≥15 pixels and at least 0.5 times the mean region area, and bounding box generation. Otsu thresholding and bounding box generation are described below.

(1) Adaptive image binarization

Otsu's method [29] determines the segmentation threshold by maximizing between-class variance, separating the feature map into foreground and background. Adaptive

thresholding reduces manual parameter tuning but does not guarantee invariant segmentation under different illumination conditions. The binary image $B(x, y)$ is defined as follows:

$$B(x,y) = \begin{cases} 1 & \text{if I}(x,y) > T \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Here, $I(x, y)$ is the pixel value of the input feature map. Figure 6b shows the binary mask, in which white pixels represent candidate seedling foreground.

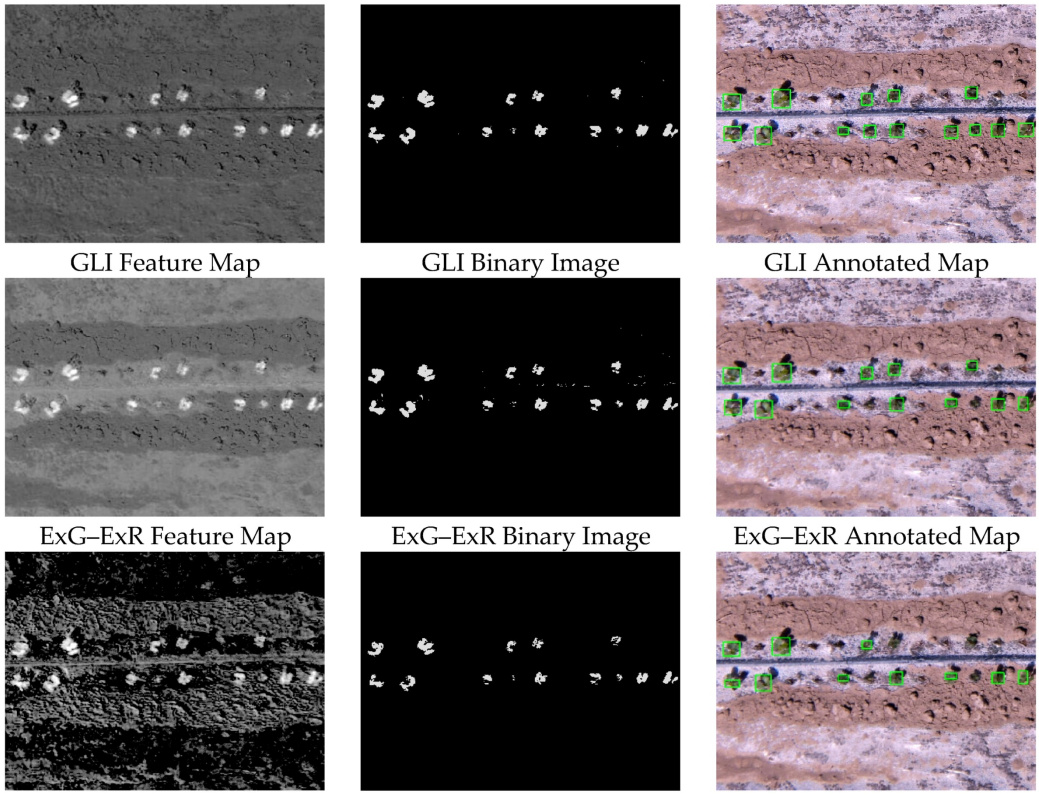


Figure 6. Automatic annotation workflow for GLI, ExG–ExR, and HSV. Rows from top to bottom correspond to GLI, ExG–ExR, and HSV, respectively. Columns show (**a**) feature maps, (**b**) binary masks, and (**c**) automatically generated candidate annotations. In the binary masks, white pixels indicate candidate seedling foreground, whereas black pixels indicate background. Green boxes in column (**c**) represent candidate bounding boxes generated after morphological processing and connected-component filtering.

In Equation (3), x and y are pixel coordinates, T is the Otsu threshold, and $B(x, y)$ is a 0/1 mask, distinct from the blue channel B in Equations (1) and (2). In Equation (4), cx and cy are the centroid coordinates of the current connected component; summation is restricted to that component, and the denominator is the foreground pixel count.

(2) Centroid calculation and bounding box generation

An 8-connected component analysis was used to extract separate candidate regions and calculate the geometric centroid $C(x, y)$ of each component:

$$c_x = \frac{\sum_x \sum_y x \cdot B(x,y)}{\sum_x \sum_y B(x,y)}, \; c_y = \frac{\sum_x \sum_y y \cdot B(x,y)}{\sum_x \sum_y B(x,y)} \quad (4)$$

The minimum enclosing rectangle was generated from the foreground pixel extent of each component. The center coordinates, width, and height were recorded and exported

as normalized label files. Figure 6 shows the feature maps, binary images, and candidate boxes of the three methods.

2.5. Detector Configurations

2.5.1. YOLOv10n

YOLOv10n (hereafter referred to as YOLOv10), with approximately 2.7 M parameters, was used in this study. This predominantly convolutional model integrates multiscale information through feature fusion [30] and incorporates a partial self-attention (PSA) module [31]. It served as a lower-capacity control. Only its performance under the specified data and training conditions was compared; its structure is shown in Figure 7.

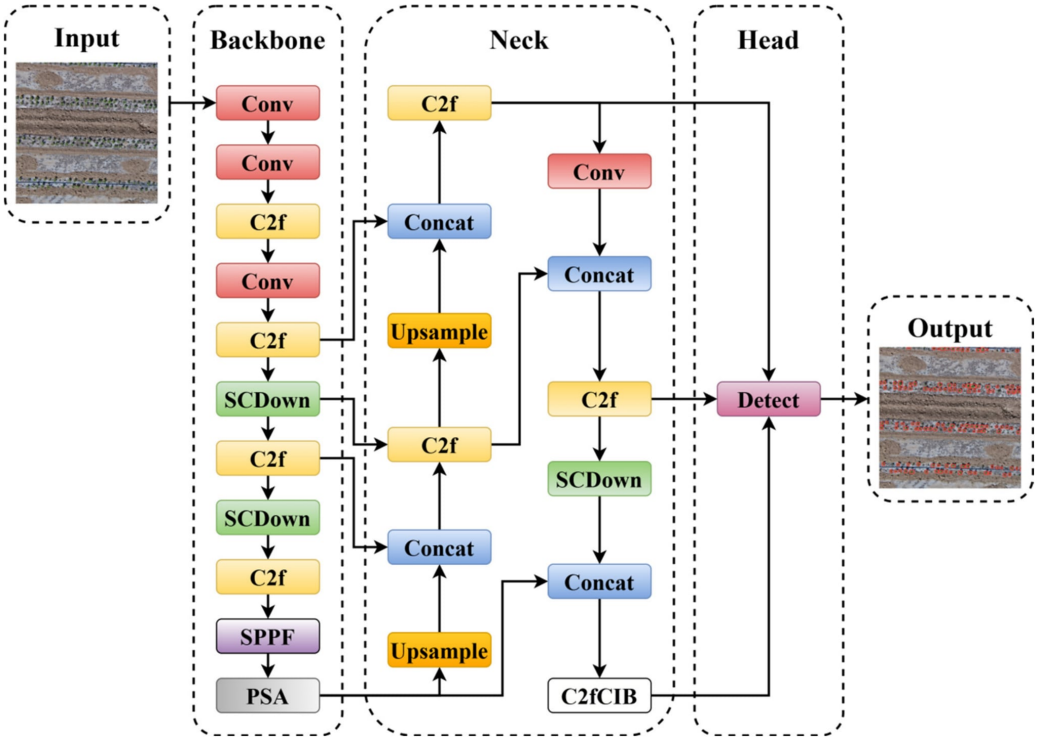


Figure 7. Architecture of the YOLOv10n model used in this study. The network consists of three main parts: Backbone, Neck, and Head. The abbreviations in the diagram are defined as follows: Conv (Convolutional layer), C2f (Cross Stage Partial network with two convolutions), SCDown (Spatial-Channel Downsampling), SPPF (Spatial Pyramid Pooling-Fast), PSA (Partial Self-Attention), Concat (Concatenation), Upsample (Upsampling layer), and C2fCIB (Cross Stage Partial with Compact Inverted Block). Arrows indicate the direction of data and feature flow through the network. Red bounding boxes in the output image indicate detected cotton seedlings.

2.5.2. RT-DETR-L

RT-DETR-L (hereafter referred to as RT-DETR), with approximately 32.8 M parameters, was used in this study. It combines a convolutional backbone with a Transformer and performs multiscale feature interaction through an efficient hybrid encoder [23]. It was evaluated as a specific detector configuration; its structure is shown in Figure 8.

2.5.3. Mamba-YOLO-B

We used Mamba-YOLO-B (hereafter referred to as Mamba-YOLO) with around 21.8 M parameters in this study. It combines convolutional and state-space modules with multiscale feature fusion. ODSSBlock utilizes the LS Block for enhancing local features and SS2D for state-space feature modeling [24]. This configuration was tested under different labeling

conditions, and its structure is shown in Figure 9. Its state-space module is built upon Mamba, proposed by Gu and Dao, which enables linear-time sequence modeling [32].

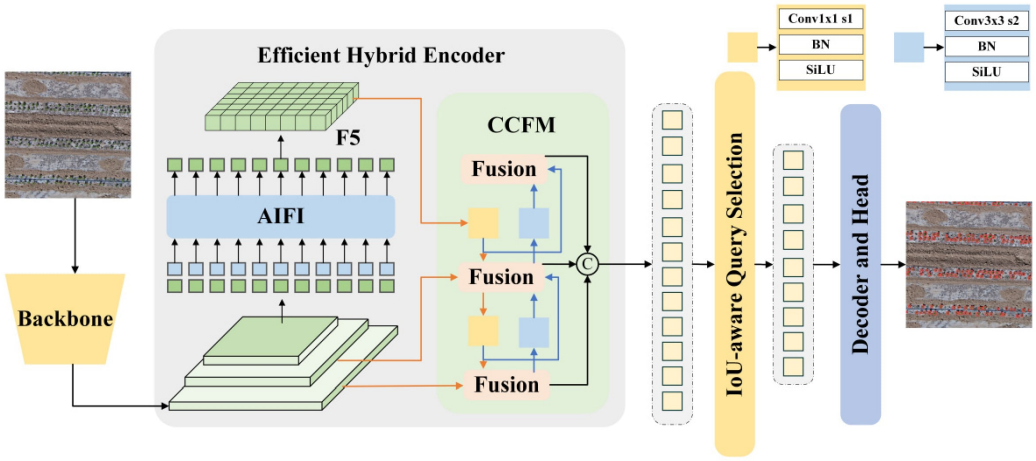


Figure 8. Architecture of the RT-DETR-L model used in this study. The model consists of a Backbone, an Efficient Hybrid Encoder, and a Decoder and Head. The abbreviations in the diagram are defined as follows: AIFI (Attention-based Intra-scale Feature Interaction), CCFM (Cross-scale Feature Fusion Module), IoU (Intersection over Union), BN (Batch Normalization), SiLU (Sigmoid Linear Unit), and Conv (Convolutional layer).Arrows indicate the direction of data and feature flow. Orange and blue arrows indicate the top-down and bottom-up feature fusion paths, respectively. Yellow and light-blue square blocks denote 1 × 1 convolutions with stride 1 and 3 × 3 convolutions with stride 2, respectively, each followed by batch normalization and SiLU activation. "C" denotes feature concatenation. Small squares represent feature tokens and selected decoder queries. Other background colors distinguish functional modules. Red bounding boxes in the output image indicate detected cotton seedlings.

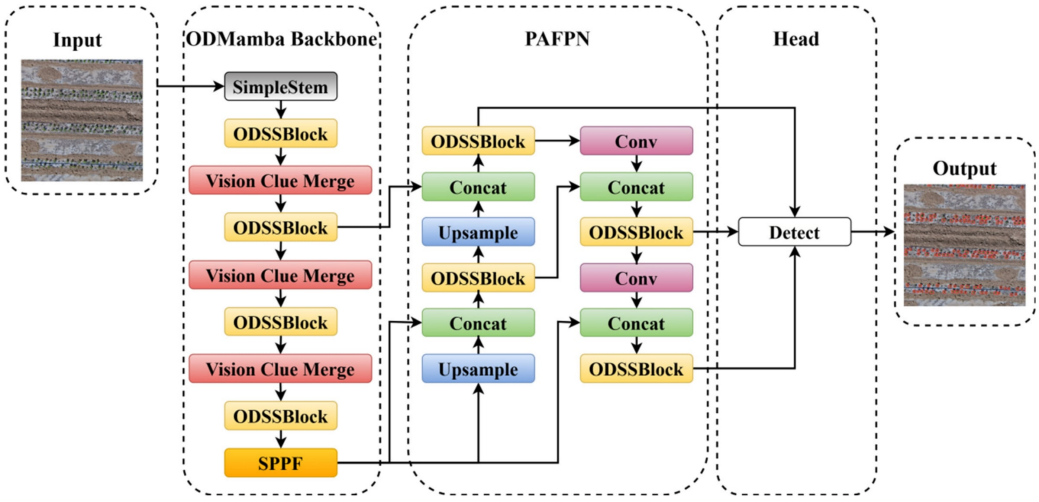


Figure 9. Architecture of the Mamba-YOLO-B model used in this study. The model mainly consists of the ODMamba Backbone, PAFPN, and Head. The abbreviations in the diagram are defined as follows: PAFPN (Path Aggregation Feature Pyramid Network), SimpleStem (a lightweight stem module for initial feature extraction), ODSSBlock (Object Detection State Space Block), Vision Clue Merge (a downsampling module for visual clue integration), SPPF (Spatial Pyramid Pooling-Fast), Concat (Concatenation), Upsample (Upsampling layer), and Conv (Convolutional layer). Arrows indicate the direction of data and feature flow through the network. Red bounding boxes in the output image indicate detected cotton seedlings.

2.6. Experimental Design

2.6.1. Spatial Data Partitioning

To reduce spatial information leakage caused by overlapping adjacent UAV images, the 796 source-region images were spatially grouped using GPS positions. S_train, S_val, and S_test contained 546, 59, and 67 images, respectively; 124 buffer-zone images were excluded from the main experiment. Image centers from different subsets were separated by at least 10 m, exceeding the approximately 4.4 m diagonal ground coverage of a single image, to reduce the risk of the same ground target appearing in both training and test sets. Figure 10 shows the partition.

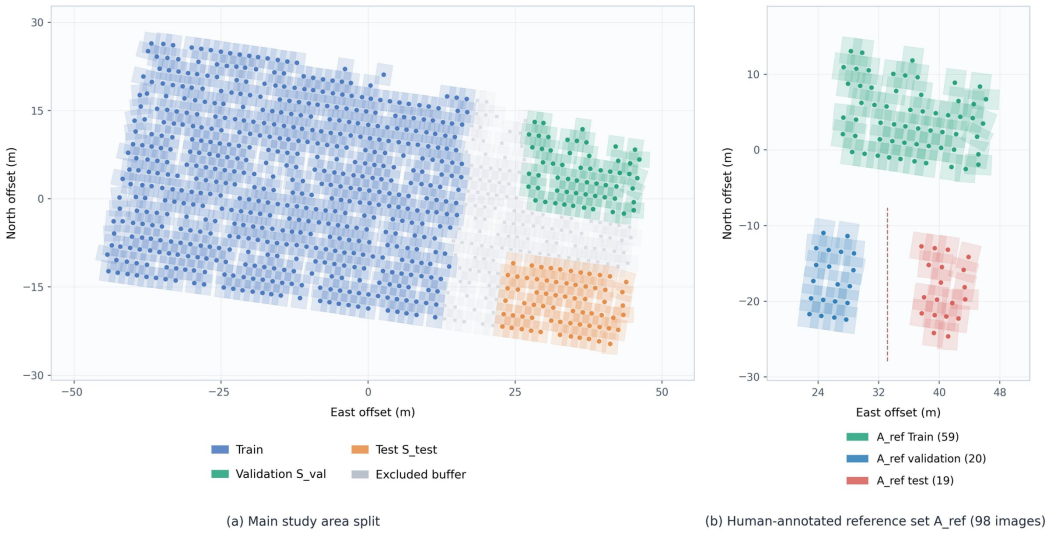


Figure 10. Spatial partitioning and ground coverage of source-region images at Huaxing Farm, Changji City. (**a**) Blue, green, and orange indicate the 546 training, 59 validation, and 67 test images, respectively; gray indicates 124 excluded buffer-zone images. (**b**) Green, blue, and red indicate the 59 training, 20 validation, and 19 test images in the A_ref matched experiment. Translucent rectangles represent estimated image ground coverage, and points indicate image centers.The red dashed line marks the spatial division between the A_ref validation and test subsets.

The 98-image manual reference set A_ref comprised the 59 source-region S_val images and two subsets of 20 and 19 images repartitioned from the S_test region with 10 m spatial separation. A_ref was used to assess automatic annotation quality and conduct auxiliary matched experiments, with 59 training, 20 validation, and 19 test images (Table 4).

Table 4. Data partitioning, label sources, and experimental uses.

Dataset	Spatial Region/Description	Number of Images	Label Source	Experimental Use
S_train	Source region (Huaxing Farm training area)	546 images	GLI/ExG–ExR/HSV automatic labels	Model training
S_val	Source region (Huaxing Farm validation area)	59 images	Manual annotation	Trained model validation
S_test	Source region (Huaxing Farm test area)	67 images	Manual annotation	Model testing and evaluation
E_test	Cross-region test area (Kuqa City, southern Xinjiang)	192 tiles	Manual annotation	Cross-region model generalization evaluation
A_ref	Source region: 59 S_val images and repartitioned test-area subsets of 20/19 images	98 images (59/20/19)	Manual reference labels; GLI/ExG–ExR/HSV labels	Comparison of manual and automatic label quality

Note: S_train, S_val, and S_test denote the main training, validation, and test sets. S_val was used for checkpoint selection and S_test for final evaluation. A_ref was used for automatic annotation quality assessment, matched label training, and scenario subset analysis; the matched experiment used 59/20/19 training/validation/test images. E_test was used exclusively for cross-region testing.

Overlap assessment was performed on the 48 original UAV RGB images using GPS coordinates and spatial coverage. Twelve images had mutually non-overlapping spatial coverage. All 12 images were included in the cross-region test set. Each image was divided into a 4 × 4 grid of non-overlapping tiles. This yielded 192 tiles for E_test. Figure 11 illustrates the tiling procedure.

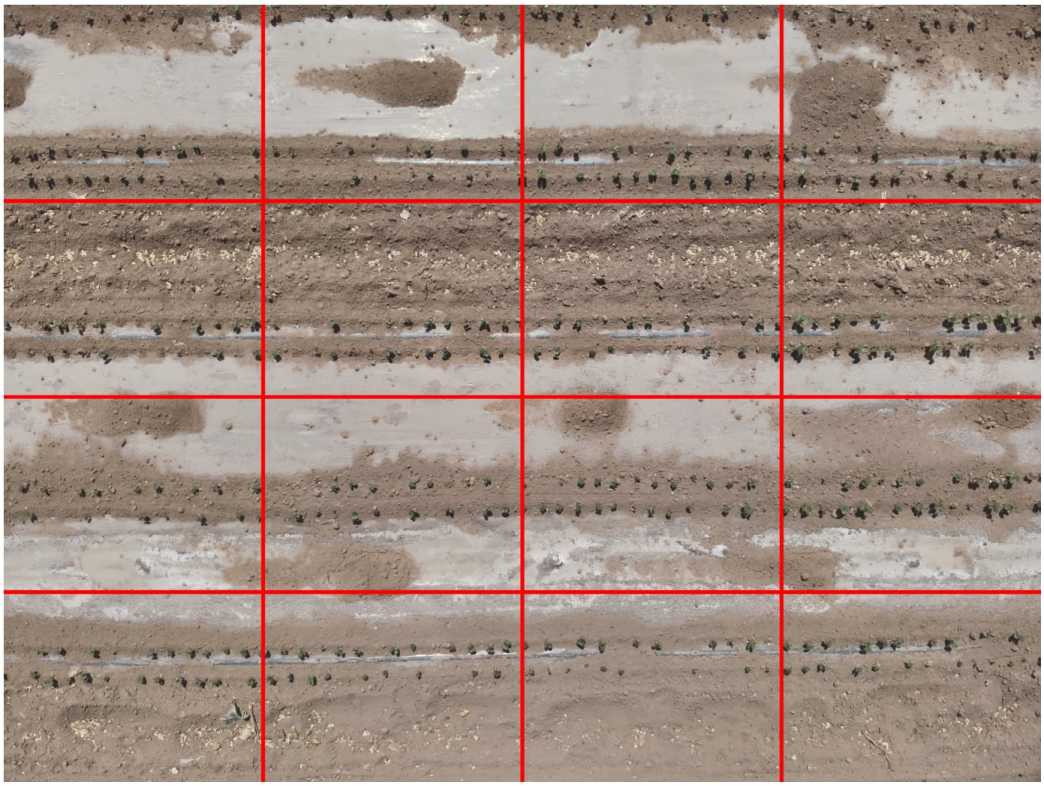


Figure 11. Tiling of a representative cross-region RGB image. The original image (8000 × 6000 pixels) was divided into a 4 × 4 grid, producing 16 non-overlapping tiles of 2000 × 1500 pixels. Red lines indicate tile boundaries.

2.6.2. Label Source Controls and Matched Design

To assess joint strategies combining automatic annotation methods and detector architectures, three training datasets were constructed using GLI, ExG–ExR, and HSV. Three detectors were selected: YOLOv10 (a predominantly convolutional model incorporating partial self-attention), RT-DETR (a Transformer-based detector), and Mamba-YOLO (a CNN–SSM hybrid architecture). Pairing the three datasets with the three models yielded nine specific detector instances for subsequent evaluation and comparison. Table 5 lists the combinations.

Table 5. Nine combinations of detector configurations and training labels.

Combined Strategy Name	Annotation Method	Detection Model	Architecture Type
Option 1	GLI	RT-DETR	CNN + Transformer
Option 2	GLI	Mamba-YOLO	CNN + SSM
Option 3	GLI	YOLOv10	CNN + PSA
Option 4	ExG–ExR	RT-DETR	CNN + Transformer
Option 5	ExG–ExR	Mamba-YOLO	CNN + SSM
Option 6	ExG–ExR	YOLOv10	CNN + PSA
Option 7	HSV	RT-DETR	CNN + Transformer
Option 8	HSV	Mamba-YOLO	CNN + SSM
Option 9	HSV	YOLOv10	CNN + PSA

Images were annotated in batches using the three automatic algorithms. GLI, ExG–ExR, and HSV were applied to the same 546 training images, producing 80,315, 80,418, and 81,961 candidate boxes, respectively. On the 59 validation images, the corresponding numbers were 8146, 7929, and 8330, describing generation scale only. Model selection used manual validation labels, and the 67 test images were also manually annotated. Table 4 summarizes dataset uses.

Manual reference labels were prepared for the matched experiment using a fixed subset of 59 training, 20 validation and 19 test images from A_ref. Each detector was trained individually using manual and GLI labels, with identical images, hyperparameters, and augmentation, and tested against full manual labels on the same 19 test images. The three GLI-trained detectors were additionally trained on the main 546-image training set, using seeds 0, 1, and 2, and the results are summarized on the same S_test set. Paired *t*-tests matched runs by seed and were only used as exploratory evidence due to the limited number of repetitions.

All three detectors used common training settings. YOLOv10n, RT-DETR-L, and Mamba-YOLO-B had approximately 2.7 M, 32.8 M, and 21.8 M parameters, respectively, and were not capacity-matched. The comparison therefore concerns these specific configurations under the specified conditions and does not establish absolute architectural superiority.

2.6.3. Perturbation and Parameter Response Experiments

We performed four controlled experiments to evaluate the effects of label quality, illumination, input resolution and confidence thresholds on the accuracy of the model. In the label perturbation experiments, the training labels were perturbed and the models retrained, with training images and validation and test labels held constant. Illumination, resolution, and confidence-threshold experiments used fixed trained weights. The corresponding variables were controlled separately in each experiment, and Table 6 shows the settings and evaluation metrics. In addition, a fixed-threshold sensitivity scan was performed on S_val to examine how the GLI segmentation threshold affects annotation quality; this scan was diagnostic only and was not used for threshold or method selection, and S_test was not involved.

Table 6. Auxiliary perturbation experiment parameters and controlled conditions.

Experiment	Simulated Scenario	Data and Variable Settings (Single Variable)	Fixed Conditions	Evaluation Metrics
Training label omission	Detection performance after training label omissions	Randomly omit 10% of target boxes from GLI labels in the A_ref training set; evaluate the trained model against complete manual labels on the same test set.	Training images and validation/test labels	mAP@0.5, mAP@0.5:0.95
Illumination perturbation	Cloud shadows and nonuniform illumination	S_test; shadow s = 0.2/0.4/0.6; brightness k = 0.5–1.5	Model weights and manual reference boxes	Annotation F1; detection mAP and P/R/F1
Resolution sensitivity	Limited edge-device computation	S_test; input 320 × 320, 416 × 416, 512 × 512, 640 × 640	Model weights, original images, and labels	Two mAP metrics and per-image inference time
Threshold sensitivity	Deployment trade-off between missed and false detections	S_val (scan 0.10–0.90); S_test/E_test (fixed 0.25)	Model weights and 640 input	P, R, F1, MAE, Bias

(1) Training-label omission: To evaluate missing training labels, 10% of target boxes were randomly omitted from the GLI labels of the A_ref training set. Models were retrained under this condition and evaluated against complete manual references on the same test set. Training images and validation and test labels were unchanged. (2) Illumination perturbation: Gaussian-smoothed random fields generated synthetic shadows (s = 0.2, 0.4, 0.6) superimposed on S_test images to simulate field cloud shadows. Detection was

evaluated using fixed model weights and unchanged manual reference boxes. Uniform brightness changes (k = 0.5–1.5) were also examined for automatic annotation quality. (3) Resolution sensitivity: model weights, original images, and labels were fixed, while input resolutions were set to 320 × 320, 416 × 416, 512 × 512, and 640 × 640. The two mAP metrics and per-image inference time were evaluated on S_test to assess accuracy–time trade-offs under computational constraints. (4) Confidence threshold sensitivity: with fixed model weights and 640 × 640 inputs, conf = 0.10–0.90 was scanned on S_val. Main counting evaluations on S_test and E_test used conf = 0.25 to avoid test-set threshold selection.

For scenario stratification, contrast was defined as the pixel standard deviation of the grayscale image, and target density as the number of manual boxes per image. The same A_ref images were divided at the median of each dimension: low/high contrast groups contained 48/50 images, and low/high density groups contained 47/51 images. Images and labels were unchanged.

For photometric perturbations, uniform brightness transformation multiplied each pixel by k and clipped the result to 0–255:

$$I_k(x, y) = \text{clip}(kI(x, y),\ 0,\ 255) \tag{5}$$

Synthetic shadows were generated from a low-resolution random field of size $\lfloor h/32 \rfloor \times \lfloor w/32 \rfloor$, enlarged to the original image size using bilinear interpolation, Gaussian-smoothed with $\sigma = 200/32$ pixels, and normalized to $N(x, y) \in [0, 1]$. The random field used seed = 42, and all models received identical perturbed images. The illumination field and perturbed image were defined as follows:

$$F(x, y) = 1 - sN(x, y), I_{\text{shadow}}(x, y) = I(x, y)F(x, y) \tag{6}$$

Photometric experiments evaluated automatic annotation pipelines and trained detectors separately. For automatic annotation, the unperturbed baseline and all perturbation conditions, including $s = 0$, used the same feature calculation, image-wise Otsu thresholding, and post-processing. Perturbed RGB values were clipped to 0–255 and converted to uint8 before feature calculation. For detection, model weights were fixed and inference used identical perturbed images. Manual reference boxes remained unchanged in both experiments. Confidence threshold sensitivity was assessed on S_val by scanning conf = 0.10–0.90 in increments of 0.05, recording Precision, Recall, F1, Bias, MAE, and RMSE, comparing configuration responses, and reporting the peak F1 and minimum counting error within the scan. The scan did not alter the common main counting threshold: S_test and E_test counting used conf = 0.25. Detection AP used conf = 0.001 to retain low-confidence predictions for Precision–Recall curves; settings are detailed in Section 2.8.

Spatial grouping for source-region uncertainty estimation was performed within S_test without changing the training, validation, or test partitions. Pairwise distances between the GPS coordinates of the 67 S_test images were calculated using the Haversine formula. Images separated by no more than 2 m were connected. Directly or indirectly connected images were assigned to the same spatial group, yielding 14 groups. These groups were used as the bootstrap resampling units to reduce pseudoreplication caused by overlapping images.

2.6.4. Cross-Region Applicability Evaluation Design

E_test was used for a preliminary evaluation of source-region models in an independently acquired cross-region scenario. All nine models used source-region weights and common inference settings without target-domain fine-tuning or threshold selection. Detection mAP and counting errors were compared between regions, and relative AP changes were calculated. Camera, ground scale, and background changed simultaneously. Therefore, the experiment evaluated combined domain shift. Cross-region 95% confidence intervals for

detection AP and counting MAE were obtained from 20,000 bootstrap resamples grouped by the 12 selected non-overlapping original images.

All tiles from each sampled original image were retained. Identical resampling lists were used across models to preserve pairing. The 67 S_test images were assigned to 14 spatial groups using a 2 m GPS-distance threshold. Source-region counting intervals were obtained from 20,000 bootstrap resamples of these spatial groups, retaining all images within each sampled group. Cross-region results apply only to this Kuqa acquisition scenario and do not isolate the effects of geography, year, sensor, or scale.

2.7. Evaluation Metrics

2.7.1. Detection and Localization Metrics

Predicted boxes were evaluated using Precision, Recall, F1, and mAP for detection accuracy and mean IoU of successfully matched boxes for localization quality.

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{7}$$

At an IoU threshold of 0.5, predictions were classified as true positives (TP), false positives (FP), or false negatives (FN), and Precision, Recall, and F1 were calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

AP summarizes the area under the Precision–Recall curve across confidence thresholds. This study involved a single cotton seedling class, so mAP equals the AP of that class. N denotes the number of recall sampling points:

$$\text{AP} = \sum_{k=1}^{N} (\text{Recall}_k - \text{Recall}_{k-1}) \times \text{Precision}_k \tag{11}$$

Both mAP@0.5 and mAP@0.5:0.95 are reported. The latter averages AP at IoU thresholds from 0.50 to 0.95 in increments of 0.05 to evaluate stricter localization performance. Mean IoU for automatic annotation includes only successfully matched boxes and excludes missed targets.

Predicted and manual boxes were matched one-to-one, and duplicate predictions were not counted repeatedly as TP. Mean IoU included only successfully matched boxes, excluding omissions. In Equations (8)–(10), TP, FP, and FN denote matched boxes, unmatched predictions, and unmatched manual targets, respectively. In Equation (11), N is the number of recall sampling points, k is the sample index, and Recall_k and Precision_k are the corresponding recall and precision.

Predictions from the deep learning detectors were sorted by descending confidence. A prediction was considered a TP if its IoU with an unmatched GT box was ≥0.5. Truncated or overlapping seedlings were retained and evaluated as ordinary targets. If a prediction overlapped multiple GT boxes, the GT box with the highest IoU was matched.

2.7.2. Counting Metrics

The image- or tile-level counting error was defined as $e_i = \hat{y}_i - y_i$. MAE is the mean of $|e_i|$, Bias is the mean of e_i, and RMSE is calculated from the differences between predicted and manual counts. Positive and negative Bias indicate overcounting and undercounting,

respectively. Equations (12) and (13) define R^2 for raw predicted counts and RMSE, respectively. Counting was evaluated primarily using MAE, RMSE, and Bias. Field-level unique plant counts or emergence rates after cross-view deduplication were not reported. Units are plants/image for source-region testing and plants/tile for cross-region testing. Because coverage differs, MAE values are not compared directly across regions.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{12}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{13}$$

Here, n is the number of evaluated images or tiles; y_i and $\hat{y}_i$ denote the manual and predicted counts for image or tile i, respectively; and $\bar{y}$ denotes the mean manual count.

Connected-component merging was defined as one automatic box overlapping at least two manual boxes (IoU > 0). Splitting was defined as one manual box overlapping at least two automatic boxes. Merging and splitting cases were counted separately.

2.7.3. Model Size and Resource Metrics

Resource requirements were characterized using parameter counts, floating-point operations, inference time, and reported training time.

Parameter counts are expressed in millions (M) to describe model size. Model file size also depends on numerical precision and storage format.

FLOPs denotes the floating-point operation count for one forward pass, generally expressed in billions (G), whereas FLOP/s denotes operations per second. Inference time also depends on hardware, numerical precision, and implementation and cannot be determined from FLOPs alone.

Training time refers to the time required for 500 epochs on fixed hardware. Only recorded training times were described; missing timing or memory records were not treated as evidence for a complete resource comparison.

2.8. Experimental Platform

Training was performed on Linux using a single NVIDIA Tesla V100-SXM2-32GB GPU(NVIDIA Corporation, Santa Clara, CA, USA), Python 3.10.10, PyTorch 2.0.1, CUDA 11.7, and Ultralytics 8.2.29. All models used 640 × 640 inputs, batch = 16, 500 epochs, Adam, an initial learning rate of 0.001, a final learning rate ratio of 0.01, and workers = 1.

Under a common training protocol without pretrained weights, all models were trained from scratch with patience = 0, mosaic = 1.0, fliplr = 0.5, hsv_h = 0.015, hsv_s = 0.7, and hsv_v = 0.4. The main experiment used seed = 0, with seeds 1 and 2 for repeated GLI training. Training times for YOLOv10, RT-DETR, and Mamba-YOLO were approximately 3.4, 6.9, and 5.2 h, respectively.

Models were tested using the best.pt checkpoints selected during training based on S_val performance. Counting used conf = 0.25 and max_det = 300, with the number of retained boxes as the predicted image- or tile-level count. Each model retained its implementation-specific output post-processing. Detection AP used conf = 0.001 to retain low-confidence predictions for Precision–Recall curves, serving a different purpose from conf = 0.25 for counting. A common counting threshold establishes a shared confidence operating point but does not ensure equivalent confidence calibration; counting results should therefore be interpreted alongside S_val threshold sensitivity.

For automatic annotation efficiency, automatic processing time covered image reading through label file generation. Manual annotation time was estimated from the mean duration of three operators experienced in cotton seedling annotation working independently

on identical images. The comparison represents initial box drawing only and excludes subsequent review, supplementation, and correction.

Source-region images and cross-region tiles were letterboxed to 640 × 640 while preserving aspect ratios. Median target box sizes were approximately 40.5 × 37.5 pixels in source-region 2592 × 1944 images and 51.0 × 48.7 pixels in cross-region 2000 × 1500 tiles, corresponding to approximately 10.00 × 9.26 and 16.32 × 15.58 pixels at the 640 input size. Input-size sensitivity settings are described in Section 2.6.3.

A separate learning-rate sensitivity analysis is provided in Supplementary Section S1 and Table S1.

3. Results

3.1. Automatic Annotation Quality, the GLI-Alone Baseline, and Automatic Annotation Efficiency

The partitions and manual references described in Section 2 were used: A_ref for automatic annotation quality, S_test for main source-region testing, and E_test for cross-region detection and counting.

3.1.1. Automatic Annotation Quality Against Manual References

GLI achieved the best automatic annotation quality, followed by ExG–ExR and HSV (Table 7). Its Precision, Recall, F1, and mean IoU were 0.9912, 0.9341, 0.9618, and 0.8978, respectively, the highest across all four metrics. This indicates fewer incorrect boxes and omissions and closer agreement between matched boxes and manual references.

Table 7. Candidate box quality of three automatic annotation methods, using manual annotation as the gold standard.

Method	Precision	Recall	F1-Score	Mean IoU	Type
GLI	99.12%	93.41%	96.18%	89.78%	Ratio-based vegetation index
ExG–ExR	95.7%	86.9%	91.09%	87.6%	Difference vegetation index
HSV	92.4%	86.4%	89.30%	81.9%	Color space method

Note: All methods were matched against the same manual references on the 98 A_ref images, with manual annotation as the gold standard. Automatic and manual boxes used the same IoU matching threshold: TP denotes matched boxes, FP unmatched automatic boxes, and FN unmatched manual boxes. Precision = TP/(TP + FP), Recall = TP/(TP + FN), and F1 is their harmonic mean. Mean IoU is the mean IoU between TP boxes and their manual matches; unmatched targets are excluded. All metrics were calculated relative to manual annotations.

Because GLI recall remained below 1, initial automatic boxes require manual review, particularly to add missed seedlings and correct inaccurate box locations.

GLI labels were therefore prioritized for subsequent training, while ExG–ExR and HSV were retained as controls.

3.1.2. Comparison of the GLI-Alone Detection Baseline and GLI + Deep Learning Models

Table 8 compares GLI-alone with three GLI-trained detectors in detection, localization, and counting. GLI-alone outputs boxes without confidence ranking, so its mAP is not reported. On S_test, GLI-alone achieved the highest F1 (0.9446) and matched-box mean IoU (0.8980), but its counting MAE (9.85 plants/image) exceeded those of YOLOv10 + GLI (6.493) and Mamba-YOLO + GLI (8.403). Among the detectors, Mamba-YOLO + GLI achieved the highest values for both mAP metrics, whereas YOLOv10 + GLI had the lowest counting MAE. RT-DETR + GLI had the highest MAE (15.731 plants/image) and a Bias of +15.731 plants/image, indicating substantial average overcounting. Detection and localization rankings therefore differed from counting MAE rankings. Per-image counting error equals false positives minus false negatives. Detection and localization metrics do not directly report this difference and cannot replace counting evaluation.

Table 8. Detection, localization, and counting results for GLI-alone and GLI-trained models.

Method/Model	Evaluation Set	mAP@0.5/mAP@0.5:0.95	Precision	Recall	F1	Mean IoU of TP Boxes	Counting MAE	Bias	RMSE
GLI-alone	S_test (67)	—/—	0.9608	0.9289	0.9446	0.8980	9.85	−5.28	13.012
YOLOv10 + GLI	S_test (67)	0.9554/0.7336	0.9407	0.9089	0.9245	0.8707	6.493	−5.179	8.719
RT-DETR + GLI	S_test (67)	0.9651/0.7989	0.8744	0.9558	0.9133	0.8916	15.731	+15.731	16.973
Mamba-YOLO + GLI	S_test (67)	0.9682/0.7994	0.9644	0.9164	0.9398	0.8883	8.403	−7.896	10.193
GLI-alone	E_test (192)	—/—	0.9122	0.8851	0.8975	0.7336	5.880	+2.036	8.423

Note: Precision, Recall, and F1 were calculated per image and then averaged. Mean IoU includes only successfully matched boxes. Counting MAE, Bias, and RMSE are expressed in plants/image for source-region testing and plants/tile for cross-region testing. Bias is the predicted count minus the manual count; positive values indicate overcounting and negative values indicate undercounting. GLI-alone mAP is not reported because the pipeline does not assign ranking confidence scores.

Table 9 compares the counting MAE of each detector with that of GLI-alone on the 192 E_test tiles. GLI-alone achieved an MAE of 5.880 plants/tile. ΔMAE is defined as detector MAE minus GLI-alone MAE, with negative values indicating reduced counting error. YOLOv10 + GLI, RT-DETR + GLI, and Mamba-YOLO + GLI achieved MAEs of 3.833, 4.969, and 2.344 plants/tile, respectively. Their corresponding ΔMAE values were −2.047, −0.911, and −3.536 plants/tile. The paired 95% confidence intervals, calculated by grouping tiles according to their 12 original images, were below zero for all three comparisons. However, the RT-DETR + GLI interval [−2.469, −0.015] had an upper bound close to zero, providing more limited evidence of reduced counting MAE than the YOLOv10 + GLI and Mamba-YOLO + GLI comparisons. Mamba-YOLO + GLI achieved the lowest MAE.

Table 9. Counting performance and paired MAE differences between GLI-alone and GLI-trained detectors on E_test.

Method/Model	Evaluation Set	Counting MAE	ΔMAE	Paired 95% CI for ΔMAE
GLI-alone	E_test (192)	5.880	—	—
YOLOv10 + GLI	E_test (192)	3.833	−2.047	[−2.703, −0.422]
RT-DETR + GLI	E_test (192)	4.969	−0.911	[−2.469, −0.015]
Mamba-YOLO + GLI	E_test (192)	2.344	−3.536	[−4.651, −2.630]

Note: All methods were evaluated on the same 192 E_test tiles. MAE and ΔMAE are expressed in plants/tile. ΔMAE is defined as detector MAE minus GLI-alone MAE; negative values indicate reduced counting error. Paired 95% confidence intervals were obtained from 20,000 bootstrap resamples grouped by the 12 original images. Each resample included all tiles from each sampled original image. The same sampled groups were used for both methods, and the difference between their MAEs was calculated within each resample. The intervals were defined by the 2.5th and 97.5th percentiles of the resulting difference distribution.

Together, Tables 8 and 9 show that training detectors with GLI-generated labels reduced counting MAE in this cross-region test, whereas the source-region counting benefit depended on the detector. Considering detection accuracy and cross-region counting performance together, Mamba-YOLO + GLI was the best-performing combination evaluated in this study.

3.1.3. Comparison of Automatic and Manual Annotation Efficiency

The initial manual box drawing for 200 targets took about 600 s, while GLI generated the initial boxes in 0.399 s. But this comparison is just for box generation. The subsequent manual review, supplementation and correction also need time. The HSV, GLI and ExG–ExR took 0.166, 0.399 and 0.429 s respectively. These results and Table 7 demonstrate that GLI achieved reasonably good box-level quality and fast generation time. Table 10 summarizes the automatic box generation time.

Table 10. Initial box drawing time for 200 targets.

Method	Time for 200 Targets (s)	Time Saved Relative to Manual Annotation (s)	Manual/Automatic Time Ratio
Manual annotation	Approximately 600	0	1×
ExG–ExR	0.429	599.571	Approximately 1399×
HSV	0.166	599.834	Approximately 3614×
GLI	0.399	599.601	Approximately 1504×

Note: Timing covered 200 targets. Automatic processing excludes manual review, missed-label supplementation, and incorrect-box correction. The time ratio is manual annotation time divided by automatic generation time and represents acceleration of initial box drawing only, not the labor savings of the complete annotation workflow.

3.2. Effects of Training Label Sources on Detector Performance

3.2.1. Accuracy Comparison of Joint Strategies

Among the nine combinations, Mamba-YOLO + GLI performed best on the source-region test set, achieving mAP@0.5 = 0.9682, mAP@0.5:0.95 = 0.7994, and Precision = 0.9644. It ranked first in both mAP metrics and Precision, and third in Recall among the nine combinations (Table 11).

Table 11. Accuracy comparison of nine joint strategies on the source-region test set.

Model	Training Labels	mAP@0.5	mAP@0.5:0.95	Precision	Recall
YOLOv10n	GLI	0.9554	0.7336	0.9407	0.9089
YOLOv10n	ExG–ExR	0.9502	0.7184	0.9487	0.8865
YOLOv10n	HSV	0.9413	0.6229	0.9297	0.8602
RT-DETR-L	GLI	0.9651	0.7989	0.8744	0.9558
RT-DETR-L	ExG–ExR	0.9516	0.7564	0.9280	0.9040
RT-DETR-L	HSV	0.9358	0.6984	0.9187	0.8833
Mamba-YOLO-B	GLI	0.9682	0.7994	0.9644	0.9164
Mamba-YOLO-B	ExG–ExR	0.9597	0.7635	0.9604	0.9075
Mamba-YOLO-B	HSV	0.9617	0.7162	0.9385	0.9201

Note: All nine combinations used the same 67 S_test images and manual references, with results from seed0 weights. Training labels indicate automatic label sources. mAP@0.5 was evaluated at IoU = 0.5, whereas mAP@0.5:0.95 averages results over IoU = 0.50–0.95 in increments of 0.05. Precision and Recall denote their respective detection metrics.

GLI labels outperformed ExG–ExR and HSV in localization accuracy and target coverage. GLI achieved the highest mean mAP@0.5 (0.9629), mAP@0.5:0.95 (0.7773), and Recall (0.9270), with the largest advantage over HSV in mAP@0.5:0.95 (+0.0981; Figure 12A).

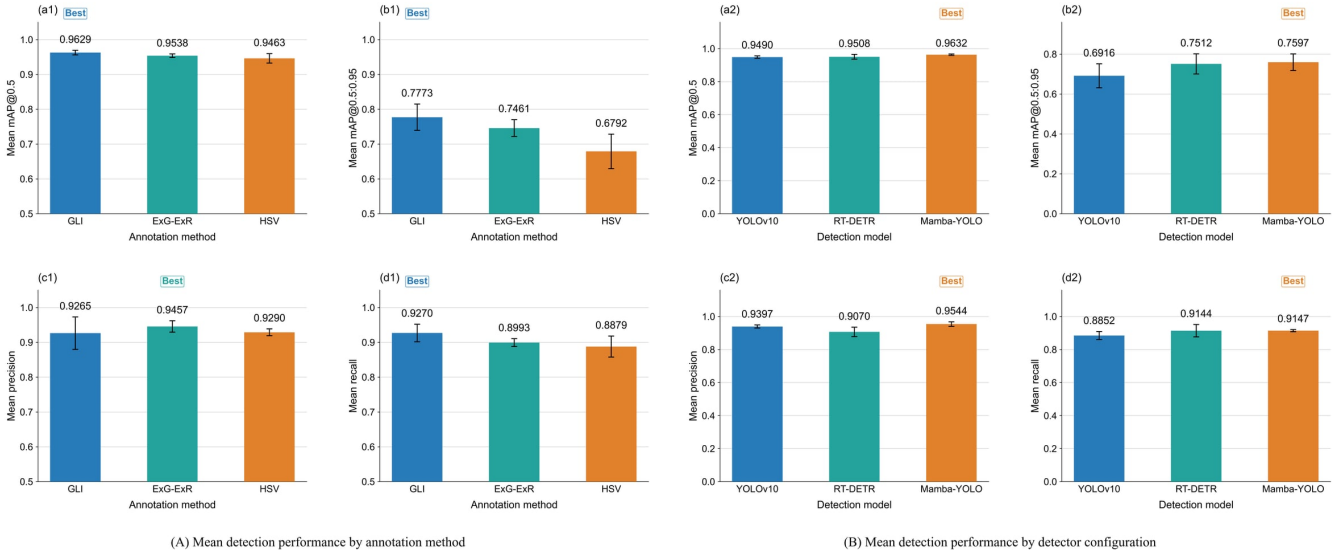
Mamba-YOLO performed better on average among the three configurations. Averaged across the three automatic label conditions, it achieved the highest mAP@0.5 (0.9632), mAP@0.5:0.95 (0.7597), Precision (0.9544), and Recall (0.9147). Its mAP@0.5:0.95 exceeded YOLOv10 and RT-DETR by 0.0681 and 0.0085, respectively, while Recall was close to RT-DETR (0.9144; Figure 12B).

3.2.2. Model Accuracy with Manual and GLI Labels

GLI automatic labels effectively supported training, with the accuracy of all three models largely unaffected.

With other variables controlled, the responses to GLI labels are reported in Table 12. Switching to GLI labels improved YOLOv10's two metrics by 0.66% and 3.76%. Mamba-YOLO's mAP@0.5 increased by 0.19%, whereas its mAP@0.5:0.95 decreased by 2.51%. RT-DETR's two metrics decreased by 1.67% and 2.28%, respectively.

The omissions and localization errors in GLI labels did not substantially impair overall model accuracy (the maximum absolute difference in the two mAP metrics was only 0.021). YOLOv10 and Mamba-YOLO were highly tolerant of these errors. Overall, models trained with GLI labels achieved detection accuracy close to that obtained with manual labels, supporting the use of GLI labels for initial model training.



(A) Mean detection performance by annotation method

(B) Mean detection performance by detector configuration

Figure 12. Mean detection performance across annotation methods and detector configurations. The left group shows the performance of each annotation method averaged across three detector configurations: (**a1**) mAP@0.5; (**b1**) mAP@0.5:0.95; (**c1**) Precision; and (**d1**) Recall. The right group shows the performance of each detector configuration averaged across three annotation conditions: (**a2**) mAP@0.5; (**b2**) mAP@0.5:0.95; (**c2**) Precision; and (**d2**) Recall. Bars indicate means. Error bars indicate sample standard deviations across detector configurations in the left group and annotation conditions in the right group, rather than across independent training runs. Blue, teal, and orange represent GLI, ExG–ExR, and HSV in the left group, and YOLOv10, RT-DETR, and Mamba-YOLO in the right group. "Best" marks the highest mean in each panel and does not indicate statistical significance.

Table 12. Model accuracy using manual and GLI labels.

Model	Architecture	Manual Labels mAP@0.5/mAP@0.5:0.95	GLI Labels mAP@0.5/mAP@0.5:0.95	GLI Labels − Manual Labels	GLI − Manual (%)
YOLOv10	CNN + PSA	0.9280/0.5670	0.9341/0.5883	+0.006/+0.021	+0.66/+3.76
RT-DETR	Transformer	0.8942/0.5208	0.8793/0.5089	−0.015/−0.012	−1.67/−2.28
Mamba-YOLO	CNN–SSM	0.9513/0.6817	0.9531/0.6646	+0.002/−0.017	+0.19/−2.51

Note: This experiment used A_ref, with training, validation, and test subsets all drawn from A_ref. Values represent accuracy on the A_ref test subset.

3.2.3. Detection Performance Across Training Seeds

Across three training runs, Mamba-YOLO + GLI achieved the highest mean detection accuracy. Its mean mAP@0.5:0.95 was 0.8020, compared with 0.7928 for RT-DETR + GLI and 0.7331 for YOLOv10 + GLI (Table 13). Mamba-YOLO + GLI also achieved the highest mean mAP@0.5 (0.9692) and F1 (0.9402). However, rankings varied across runs. At seed 2, RT-DETR achieved a slightly higher mAP@0.5:0.95 than Mamba-YOLO, at 0.8048 versus 0.8038. RT-DETR achieved the highest mean Recall (0.9527).

For mAP@0.5:0.95, two-sided paired t-tests with Holm correction showed a significant difference between Mamba-YOLO and YOLOv10 (adjusted $p = 0.0036$). The difference between Mamba-YOLO and RT-DETR was not statistically significant (adjusted $p = 0.4331$; Table 14). Across the three runs, Mamba-YOLO showed less variation in this metric, with a sample standard deviation of 0.0023, compared with 0.0160 for RT-DETR. Overall, Mamba-YOLO + GLI achieved the highest mean mAP and F1 under the tested conditions, but it did not lead in every run or metric. These findings are based on only three independent training repetitions ($n = 3$) and the tested configurations. Additional repetitions are needed to establish broader deployment advantages.

Table 13. Per-seed detection results and summary statistics for three GLI-trained models on S_test.

Model	Seed	mAP@0.5	mAP@0.5:0.95	Precision	Recall	F1
Mamba-YOLO	0	0.9682	0.7994	0.9644	0.9164	0.9398
	1	0.9708	0.8027	0.9694	0.9170	0.9425
	2	0.9685	0.8038	0.9575	0.9203	0.9385
	Mean ± SD	0.9692 ± 0.0014	0.8020 ± 0.0023	0.9637 ± 0.0060	0.9179 ± 0.0021	0.9402 ± 0.0020
RT-DETR	0	0.9651	0.7989	0.8744	0.9558	0.9133
	1	0.9623	0.7746	0.9146	0.9505	0.9322
	2	0.9662	0.8048	0.9111	0.9517	0.9310
	Mean ± SD	0.9645 ± 0.0020	0.7928 ± 0.0160	0.9000 ± 0.0223	0.9527 ± 0.0028	0.9255 ± 0.0106
YOLOv10	0	0.9554	0.7336	0.9407	0.9089	0.9245
	1	0.9578	0.7311	0.9412	0.9104	0.9255
	2	0.9583	0.7348	0.9497	0.9119	0.9304
	Mean ± SD	0.9572 ± 0.0015	0.7331 ± 0.0019	0.9439 ± 0.0051	0.9104 ± 0.0015	0.9268 ± 0.0032

Note: All models were trained with GLI labels and evaluated on the same S_test set. Rows labeled 0, 1, and 2 identify the corresponding training seeds. Mean ± SD denotes the mean and sample standard deviation across the three training runs.

Table 14. Two-sided paired *t*-tests of test-set detection performance across training seeds.

Comparison	Metric	Holm-Corrected *p*-Value	Two-Sided Paired t-Statistic
Mamba-YOLO vs. YOLOv10	mAP@0.5:0.95	0.0036	40.97
Mamba-YOLO vs. YOLOv10	mAP@0.5	0.0263	13.75
RT-DETR vs. YOLOv10	mAP@0.5:0.95	0.0725	7.33
RT-DETR vs. YOLOv10	mAP@0.5	0.1214	4.82
Mamba-YOLO vs. RT-DETR	mAP@0.5:0.95	0.4331	0.97
Mamba-YOLO vs. RT-DETR	mAP@0.5	0.2793	2.39

Note: Runs were paired by training seed for two-sided paired *t*-tests. Each comparison included three paired observations with two degrees of freedom. Holm correction was applied across all six comparisons. With only three training repetitions, the tests provide exploratory evidence. A nonsignificant difference does not establish equivalent performance.

3.2.4. Model Complexity and Training Resource Consumption

YOLOv10n required the fewest resources, RT-DETR-L had the highest computational cost, and Mamba-YOLO-B occupied an intermediate position while retaining the highest detection accuracy at substantially reduced computational cost.

Table 15 summarizes parameter counts, FLOPs, training time, and GPU memory utilization. YOLOv10n had the lowest values for all metrics (2.7 M, 8.4 G, 3.4 h, and 38.2%). RT-DETR-L had the highest parameter count (32.8 M) and theoretical computational cost (107.9 GFLOPs), with training time (6.9 h) and GPU memory utilization (99.3%) also approaching their limits.

Table 15. Model size and training resource consumption of three detection architectures.

Model	Parameters (M)	FLOPs (G)	Mean Training Time (h)	Training GPU Memory Utilization (%)
YOLOv10n	2.7	8.4	3.4	38.2
Mamba-YOLO-B	21.8	49.6	5.2	89.5
RT-DETR-L	32.8	107.9	6.9	99.3

Note: Results were obtained using a single NVIDIA Tesla V100-SXM2-32GB GPU and 500 training epochs on S_train. GPU memory utilization denotes peak training memory consumption, not inference memory requirements on UAV edge devices.

Mamba-YOLO-B had substantially fewer parameters (21.8 M) and FLOPs (49.6 G) than RT-DETR-L, requiring less than half its theoretical computation. Training time decreased to 5.2 h, and GPU memory utilization decreased to 89.5%.

Together with Table 13, Mamba-YOLO-B achieved higher mAP@0.5:0.95 (0.8020 versus 0.7928) with less than half the computation of RT-DETR, indicating an excellent balance between computational efficiency and detection accuracy.

3.3. Accuracy of Detection Models Under Perturbations

3.3.1. Effects of Training-Label Perturbation on Detection Accuracy

Removing 10% of training target labels reduced both mAP metrics for all three models. Under this condition, Mamba-YOLO achieved the highest detection accuracy, followed by YOLOv10 and RT-DETR.

As shown in Table 16, YOLOv10 mAP@0.5 and mAP@0.5:0.95 decreased by 0.0211 and 0.0821, respectively. The corresponding decreases were 0.0116 and 0.0398 for RT-DETR and 0.0103 and 0.0307 for Mamba-YOLO. Mamba-YOLO showed the smallest decreases in both metrics under the tested omission condition. These results describe configuration-specific responses and do not establish a general ranking of architectural noise tolerance.

Table 16. Detection results before and after omitting 10% of training target labels.

Model	Complete Labels (Matched) (mAP@0.5/mAP@0.5:0.95)	10% Label Omission (Omit10) (mAP@0.5/mAP@0.5:0.95)	Difference (Omit10 − Matched) (mAP@0.5/mAP@0.5:0.95)
YOLOv10	0.9341/0.5883	0.9130/0.5062	−0.0211/−0.0821
RT-DETR	0.8793/0.5089	0.8677/0.4691	−0.0116/−0.0398
Mamba-YOLO	0.9531/0.6646	0.9428/0.6339	−0.0103/−0.0307

Note: All experiments used A_ref. Each cell reports mAP@0.5/mAP@0.5:0.95. The matched condition used complete GLI training labels, whereas omit10 omitted approximately 10% of target boxes from the corresponding training labels. Training images and complete manual validation and test references were unchanged. Differences are omit10 minus matched; negative values indicate lower accuracy.

3.3.2. Effects of Illumination Perturbations on Model Accuracy

Trained combinations were quite resistant to smooth synthetic shadows. GLI maintained relatively stable F1 scores under the tested brightness changes and achieved a higher F1 than ExG–ExR under strong brightening ($k = 1.5$).

The maximum absolute change in mAP@0.5 among the trained models was only 0.0016 as synthetic shadow intensity increased from 0 to 0.6 (Figure 13). This shows high stability to this type of smooth illumination disturbance with negligible effects on final detection results.

With a unified pipeline, the maximum absolute change in GLI F1 relative to the unshadowed baseline was 0.0013 (Table 17), indicating low sensitivity to the tested synthetic shadows. GLI F1 remained above 0.94 under brightness changes, whereas ExG–ExR F1 decreased to 0.8629 at $k = 1.5$.

Furthermore, the effect of the segmentation threshold on annotation quality should not be overlooked (Table 18). On S_val, as the fixed threshold increased from 0.05 to 0.10, the F1 score of GLI dropped sharply from 0.9636 to 0.7148; it further decreased to 0.0533 at 0.15 and approached zero at 0.20–0.40. An excessively high threshold excludes a large amount of seedling foreground and increases missed annotations. For reference, image-wise Otsu on the same S_val images achieved F1 = 0.9641, essentially matching the best fixed threshold (0.05; F1 = 0.9636) without requiring a post hoc single threshold. This scan was used only as a diagnostic, not for threshold or method selection, and S_test was not involved. Note that Table 18 compares fixed thresholds, whereas the conventional label generation pipeline utilizes Otsu adaptive thresholding.

In addition to illumination, field contrast and target density are important environmental influences on detection. A_ref images were stratified at the median contrast and density for further verification (Table 19). F1 scores were 0.9627 and 0.9611 for low and high contrast, and 0.9639 and 0.9587 for high and low density, respectively. These small differences indicate that GLI automatic annotation maintained high quality even in complex low-contrast and high-density scenarios.

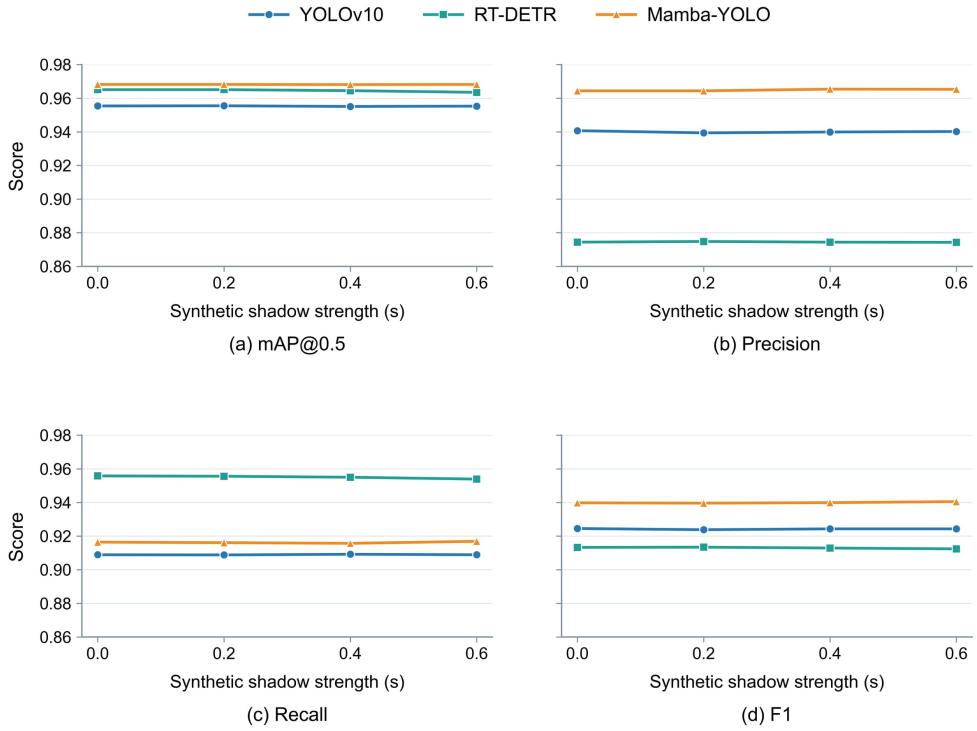


Figure 13. Effects of synthetic shadows on the detection accuracy of three GLI-trained models. (**a**) mAP@0.5; (**b**) Precision; (**c**) Recall; (**d**) F1. The horizontal axis denotes synthetic shadow intensity s: s = 0 is the unshadowed baseline, and s = 0.2, 0.4, and 0.6 indicate progressively stronger shadows. The vertical axis shows the corresponding metric. Blue, teal, and orange represent YOLOv10, RT-DETR, and Mamba-YOLO.

Table 17. Automatic annotation F1 of GLI and ExG–ExR under brightness and synthetic-shadow perturbations.

Perturbation	GLI	ExG–ExR
k = 0.5	0.9434	0.9384
k = 0.75	0.9435	0.9397
k = 1.0	0.9446	0.9424
k = 1.25	0.9436	0.9413
k = 1.5	0.9441	0.8629
shadow = 0	0.9446	0.9424
shadow = 0.2	0.9433	0.9402
shadow = 0.4	0.9442	0.9403
shadow = 0.6	0.9445	0.9394

Note: The same 67 S_test images and unchanged manual reference boxes were used. The factor k denotes uniform brightness scaling, with k = 1.0 as the baseline. The parameter s denotes synthetic-shadow intensity, with s = 0 as the unshadowed baseline. All baseline and perturbed images were processed using the same automatic annotation pipeline. Brightness and shadow perturbations are listed separately.

Table 18. GLI F1 at fixed segmentation thresholds and with Otsu on S_val.

Threshold	F1
0.05	0.9636
0.10	0.7148
0.15	0.0533
0.20	0.0008
0.25–0.40	0
Otsu (S_val)	0.9641

Note: Values are GLI F1 scores on S_val at the specified fixed segmentation thresholds; this diagnostic was not used for threshold or method selection, and S_test was not involved. The Otsu row was obtained with the main pipeline (image-wise Otsu on the mapped uint8 GLI image) on the same S_val images. The dimensionless thresholds were applied directly to the raw float32 GLI, ranging from −1 to 1, before mapping to 0–255 and uint8 quantization. Pixels with GLI greater than the threshold were classified as foreground. Because the fixed thresholds and Otsu operate on different value spaces, the threshold magnitudes are not directly comparable, although both F1 values were obtained on S_val.

Table 19. Evaluation of contrast and target-density subsets of A_ref.

Subset	Number of Images	P	R	F1	IoU
Low contrast	48	0.9903	0.9367	0.9627	0.8971
High contrast	50	0.9920	0.9321	0.9611	0.8984
High density	51	0.9903	0.9389	0.9639	0.8991
Low density	47	0.9926	0.9271	0.9587	0.8958

Note: Evaluation used the matches between GLI automatic boxes and manual references on the 98 A_ref images. Contrast groups contained 48 low-contrast and 50 high-contrast images; density groups contained 51 high-density and 47 low-density images.

Overall, GLI maintained high annotation quality under the tested photometric perturbations and scenario strata.

3.3.3. Effects of Image Resolution on Model Accuracy

Lower input resolution substantially reduced detection accuracy without a marked reduction in inference time. The architectures differed considerably in their sensitivity to resolution changes.

As shown in Table 20, reducing input resolution from 640 × 640 to 512 × 512 decreased mAP@0.5:0.95 by 0.0590, 0.1444, and 0.0716 for YOLOv10, RT-DETR, and Mamba-YOLO, respectively. At 320 × 320, the mAP@0.5:0.95 of RT-DETR decreased sharply to 0.0997.

Table 20. Detection accuracy and per-image inference time at different input image resolutions.

Model (GLI)	320 × 320 Pixels; mAP@0.5/mAP@0.5:0.95; Inference Time (ms)	416 × 416 Pixels; mAP@0.5/mAP@0.5:0.95; Inference Time (ms)	512 × 512 Pixels; mAP@0.5/mAP@0.5:0.95; Inference Time (ms)	640 × 640 Pixels; mAP@0.5/mAP@0.5:0.95; Inference Time (ms)
YOLOv10	0.7640/0.3881; 12.45	0.8852/0.5704; 12.85	0.9323/0.6746; 12.82	0.9554/0.7336; 14.45
RT-DETR	0.3837/0.0997; 86.91	0.9150/0.4176; 86.73	0.9515/0.6545; 87.89	0.9651/0.7989; 90.88
Mamba-YOLO	0.7632/0.4215; 26.27	0.8887/0.6083; 32.35	0.9456/0.7278; 26.76	0.9682/0.7994; 26.65

Note: The first two values in each cell are mAP@0.5/mAP@0.5:0.95; the value after the semicolon is per-image inference time (ms). Fixed weights and the same 67 S_test images and manual references were used. Only input pixel resolution changed; original images and their GSD remained unchanged. Inference times were recorded as single-run measurements on a single NVIDIA Tesla V100-SXM2-32GB GPU with batch size = 1 and are therefore indicative.

These results reveal two patterns. First, resolution reduction substantially degrades local target details, with Transformer-based RT-DETR being most sensitive. Second, inference speed improvements from smaller inputs are constrained by the current hardware (e.g., YOLOv10 decreased only from 14.45 to 12.45 ms). Mamba-YOLO retained the highest mAP@0.5:0.95 at all tested resolutions. Smaller inputs provided limited inference-time gains under the tested conditions.

3.3.4. Effects of Confidence Thresholds on Model Accuracy

Detection F1 remained relatively stable around conf = 0.25 for all three models, whereas counting errors were more sensitive to threshold changes. To maintain consistent evaluation settings, all three models used a confidence threshold of 0.25 for counting. Counting results were interpreted alongside threshold sensitivity analysis on the validation set.

As shown in Figure 14, F1 curves were relatively flat around 0.25 and declined markedly only as the threshold approached 0.90 because of rapidly decreasing recall. Thus, using 0.25 as a common baseline allows all models to maintain normal detection performance.

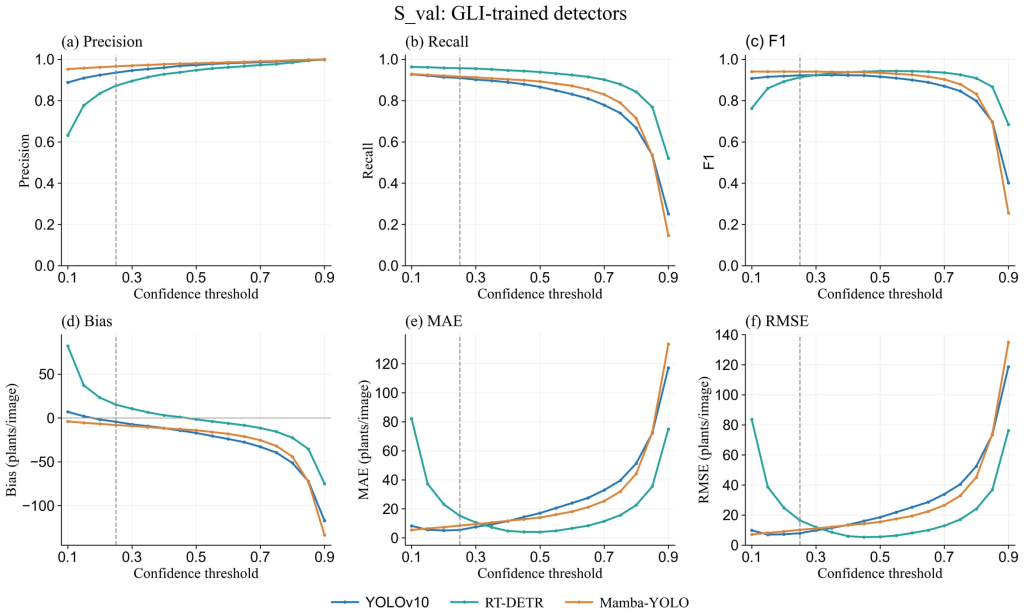


Figure 14. Detection and counting metrics versus confidence threshold for three GLI-trained configurations on S_val. (**a**) Precision; (**b**) Recall; (**c**) F1 score; (**d**) Bias; (**e**) mean absolute error (MAE); (**f**) root mean square error (RMSE).The horizontal axis is conf (0.10–0.90, step 0.05); Bias, MAE, and RMSE are expressed in plants/image. Blue, teal, and orange represent YOLOv10, RT-DETR, and Mamba-YOLO. The vertical dashed line indicates the common main counting threshold, conf = 0.25. All configurations used fixed weights and identical S_val manual references; these curves were not used to alter the main S_test or E_test counting threshold.

Counting errors changed more noticeably with confidence thresholds than detection F1. RT-DETR showed substantial overcounting at low thresholds, whereas increasing the threshold to approximately 0.45–0.50 reduced its counting errors. YOLOv10 achieved low counting errors around conf = 0.15–0.20, while Mamba-YOLO achieved its lowest errors at conf = 0.10 within the scanned range. Higher thresholds progressively increased undercounting in YOLOv10 and Mamba-YOLO.

Stable detection F1 does not guarantee stable counting performance. For the main experiment, we used conf = 0.25 as a common setting for the counting evaluation. This does not ensure identical calibration of confidence across models and count results should be interpreted alongside threshold sensitivity on the validation set. For practical counting applications, the MAE and Bias responses need to be calibrated for the specific model.

3.4. Cross-Region Generalization, Counting, and Statistical Uncertainty

3.4.1. Joint Strategy Detection Performance in Source-Region and Cross-Region Tests

Cross-region application reduced detection accuracy for all combinations to varying degrees. GLI labels still outperformed the other label sources in the cross-region scenario, and Mamba-YOLO + GLI achieved the highest accuracy.

For each model, GLI training labels produced higher values of both cross-region mAP metrics than ExG–ExR and HSV. For Mamba-YOLO, cross-region mAP@0.5 values were 0.8823, 0.2647, and 0.4708 with GLI, ExG–ExR, and HSV, respectively. Thus, despite substantial domain shift, GLI labels provided the best training performance in the cross-region target domain.

Among the three GLI-trained combinations, Mamba-YOLO + GLI achieved the highest mAP@0.5 (0.8823) and mAP@0.5:0.95 (0.3446), followed by YOLOv10 + GLI (0.8505/0.3030) and RT-DETR + GLI (0.7309/0.2234; Table 21).

Table 21. Detection results of nine combinations on source-region and cross-region test sets.

Combination	Source-Region mAP@0.5	Source-Region mAP@0.5:0.95	Cross-Region mAP@0.5	95% CI of Cross-Region mAP@0.5	Cross-Region mAP@0.5:0.95	95% CI of Cross-Region mAP@0.5:0.95
YOLOv10_GLI	0.9554	0.7336	0.8505	[0.8450, 0.8555]	0.3030	[0.2971, 0.3267]
YOLOv10_ExG–ExR	0.9502	0.7184	0.6528	[0.5786, 0.6680]	0.2660	[0.2131, 0.2773]
YOLOv10_HSV	0.9413	0.6229	0.8187	[0.8096, 0.8695]	0.2930	[0.2893, 0.3195]
RT-DETR_GLI	0.9651	0.7989	0.7309	[0.7143, 0.8033]	0.2234	[0.2150, 0.2497]
RT-DETR_ExG–ExR	0.9516	0.7564	0.5956	[0.5209, 0.6144]	0.2140	[0.1894, 0.2311]
RT-DETR_HSV	0.9358	0.6984	0.6153	[0.5944, 0.6406]	0.1909	[0.1398, 0.2002]
Mamba-YOLO_GLI	0.9682	0.7994	0.8823	[0.8784, 0.8876]	0.3446	[0.3327, 0.3477]
Mamba-YOLO_ExG–ExR	0.9597	0.7635	0.2647	[0.2577, 0.2911]	0.0853	[0.0783, 0.1093]
Mamba-YOLO_HSV	0.9617	0.7162	0.4708	[0.4002, 0.4900]	0.1823	[0.1481, 0.2041]

Note: Source-region results correspond to S_test, and cross-region results correspond to E_test. The 95% confidence intervals apply only to cross-region results. They were obtained from 20,000 bootstrap resamples grouped by the 12 non-overlapping original images, retaining all tiles from each sampled original image. The bounds are the 2.5th and 97.5th percentiles of the bootstrap distribution.

3.4.2. Transfer Performance Changes and Uncertainty of Joint Strategies

All combinations showed lower cross-region detection accuracy. The larger decreases in mAP@0.5:0.95 than in mAP@0.5 suggest greater sensitivity under stricter localization requirements.

As shown in Figure 15, cross-region values of the mAP@0.5:0.95 were substantially below source-region values for all combinations. Across the nine combinations, mAP@0.5 decreased by 8.9–72.4%, whereas mAP@0.5:0.95 decreased by 53.0–88.8%. With GLI labels, mAP@0.5 decreased by only 8.9–24.3%, but the mAP@0.5:0.95 decreased by 56.9–72.0%.

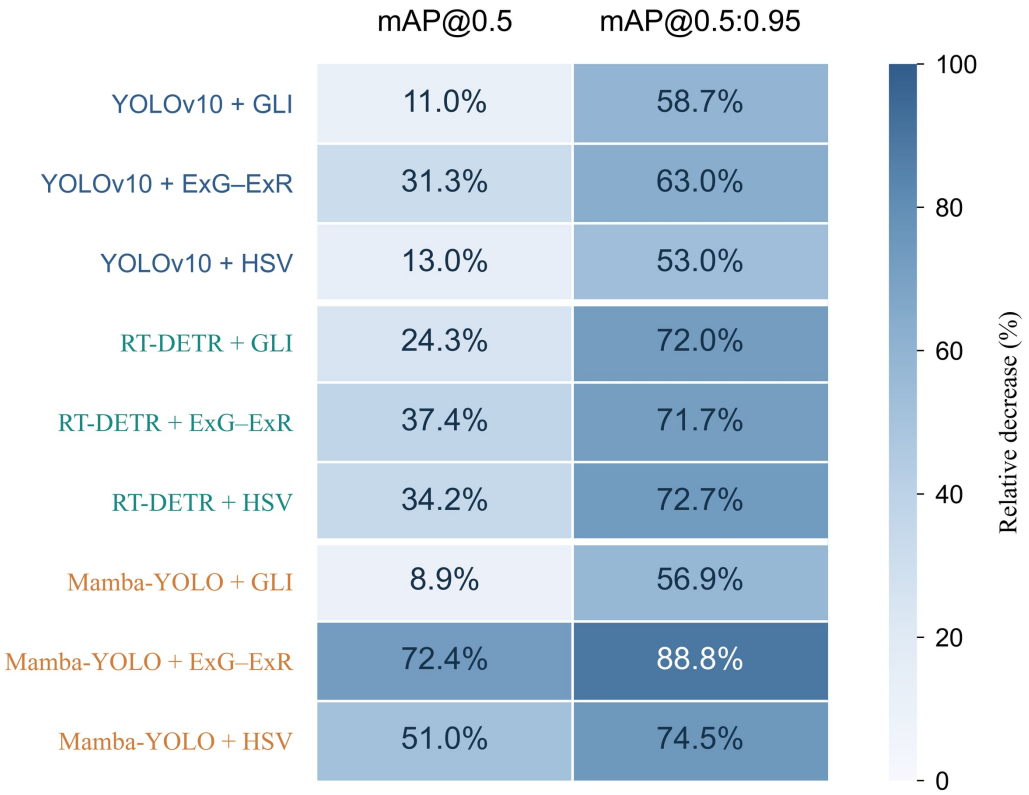


Figure 15. Relative decreases in detection mAP for nine combinations in cross-region testing. Rows indicate detector and training label combinations; columns indicate mAP@0.5 and mAP@0.5:0.95. Cell values are relative decreases, with darker colors indicating larger decreases. Relative decrease = (source-region value − cross-region value)/source-region value × 100%. Blue, teal, and orange row labels indicate YOLOv10, RT-DETR, and Mamba-YOLO, respectively.

Overall, combinations trained with GLI labels generally showed smaller decreases in detection accuracy than those trained with HSV or ExG–ExR labels. Among the GLI-trained combinations, Mamba-YOLO showed the smallest cross-region decreases (−8.9%/−56.9%), whereas RT-DETR showed the largest (−24.3%/−72.0%).

The different decreases in AP suggest greater sensitivity to cross-region changes under stricter localization requirements. The effects of background, scale, and illumination were not isolated in this study. Although Mamba-YOLO + GLI showed the smallest decrease and the highest accuracy, its mAP@0.5:0.95 still declined by more than half. Improving cross-region detection, particularly localization accuracy, therefore remains a priority for future research.

3.4.3. Image-Level Counting Performance of Joint Strategies Under Independent Spatial Grouping

All detector counting results in this section were obtained at the common confidence threshold of 0.25.

Considering detection accuracy and counting performance together, Mamba-YOLO + GLI showed the strongest overall performance among the evaluated combinations, with the highest detection AP among the GLI-trained models on S_test and the lowest MAE on E_test. YOLOv10 + GLI achieved the lowest MAE on S_test.

On the source-region test set, YOLOv10 + GLI achieved the lowest MAE (6.493 plants/image), followed by Mamba-YOLO + GLI (8.403 plants/image); RT-DETR + HSV had the highest MAE (32.701 plants/image). All detectors achieved their lowest MAE with GLI labels. All YOLOv10 and Mamba-YOLO label combinations undercounted on average, whereas all RT-DETR combinations overcounted. For YOLOv10 + GLI, RT-DETR + GLI, and Mamba-YOLO + GLI, Bias was −5.179, +15.731, and −7.896 plants/image, respectively. Table 22 provides detailed results.

Table 22. Image-level counting metrics of nine combinations on source-region S_test.

Detector	Training Labels	Sample Size	MAE	MAE 95% CI	Bias	RMSE
YOLOv10	GLI	67	6.493	[5.164, 7.940]	−5.179	8.719
YOLOv10	ExG–ExR	67	8.612	[5.567, 12.925]	−5.418	17.653
YOLOv10	HSV	67	9.791	[7.522, 12.373]	−8.209	14.062
RT-DETR	GLI	67	15.731	[14.224, 17.284]	+15.731	16.973
RT-DETR	ExG–ExR	67	23.925	[21.283, 26.597]	+23.866	26.378
RT-DETR	HSV	67	32.701	[30.284, 35.149]	+32.701	34.215
Mamba-YOLO	GLI	67	8.403	[7.090, 9.836]	−7.896	10.193
Mamba-YOLO	ExG–ExR	67	13.925	[11.686, 16.493]	−13.866	17.145
Mamba-YOLO	HSV	67	10.164	[8.090, 12.507]	−9.836	13.715

Note: All results use the same 67 S_test images. MAE, RMSE, and Bias are expressed in plants/image. Bias is predicted minus manual count; positive values indicate overcounting. The images were assigned to 14 spatial groups using GPS proximity and a 2 m distance threshold. The 95% MAE confidence intervals were defined by the 2.5th and 97.5th percentiles of 20,000 bootstrap resamples of these groups, retaining all images within each sampled group. Image-level results do not represent field-level unique plant counts after cross-view deduplication.

On the cross-region test set, Mamba-YOLO + GLI achieved the lowest MAE (2.344 plants/tile), followed by YOLOv10 + HSV (3.104) and YOLOv10 + GLI (3.833). The best training labels varied by detector: HSV for YOLOv10, ExG–ExR for RT-DETR, and GLI for Mamba-YOLO.Counting bias also varied across combinations. For example, YOLOv10 + GLI overcounted on average, whereas RT-DETR + ExG–ExR undercounted. Detailed results are provided in Table 23.

Figure 16 separately ranks the two sets of counting results, showing changes in optimal combinations and rankings between regions. Evaluation units were whole images in the source region and tiles in cross-region testing. The comparison therefore concerns combi-

nation rankings; lower cross-region MAE alone does not demonstrate improved counting accuracy after transfer.

Table 23. Tile-level counting metrics of nine combinations on cross-region E_test.

Detector	Training Labels	Sample Size	MAE	MAE 95% CI	Bias	RMSE
YOLOv10	GLI	192	3.833	[3.073, 4.569]	+3.708	4.876
YOLOv10	ExG–ExR	192	8.260	[6.468, 10.461]	−7.740	10.063
YOLOv10	HSV	192	3.104	[2.750, 3.475]	+0.375	3.862
RT-DETR	GLI	192	4.969	[4.146, 5.813]	+4.802	5.963
RT-DETR	ExG–ExR	192	4.292	[3.149, 5.465]	−2.500	5.661
RT-DETR	HSV	192	25.521	[23.024, 28.021]	+25.479	28.847
Mamba-YOLO	GLI	192	2.344	[1.826, 2.835]	−1.031	3.114
Mamba-YOLO	ExG–ExR	192	21.458	[18.779, 24.306]	−21.458	22.964
Mamba-YOLO	HSV	192	5.375	[4.532, 6.112]	+1.000	6.909

Note: This table uses 192 E_test tiles cropped from 12 original images with non-overlapping ground footprints. MAE, RMSE, and Bias are expressed in plants/tile. Bias is predicted minus manual count; positive values indicate overcounting. The 95% MAE confidence intervals were defined by the 2.5th and 97.5th percentiles of 20,000 bootstrap resamples grouped by original image, retaining all tiles from each sampled original image. Tile-level counts do not represent field-level unique plant counts after cross-view deduplication.

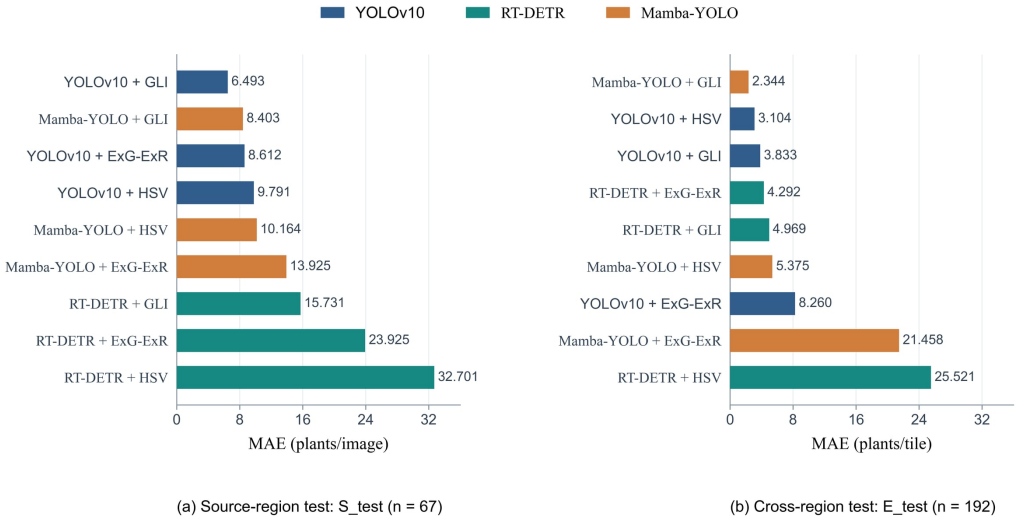


Figure 16. Counting MAE rankings of nine detector–training label combinations on source-region and cross-region test sets. (**a**) Source-region S_test; (**b**) cross-region E_test. Both panels rank combinations by ascending MAE. Bar lengths and endpoint values indicate MAE; lower values indicate smaller counting errors. Blue, teal, and orange represent YOLOv10, RT-DETR, and Mamba-YOLO.

Among the source-region GLI-trained models, Mamba-YOLO achieved the highest detection AP but a higher counting MAE than YOLOv10, indicating that high detection accuracy does not necessarily guarantee low counting errors. Figure 17 illustrates the joint effects of false positives and false negatives: in an image containing 146 seedlings, RT-DETR + GLI missed 2 and produced 16 false positives, yielding a net overcount of 14; Mamba-YOLO + GLI missed 3 and produced 4 false positives, yielding a net overcount of 1.

YOLOv10 + GLI had a Bias of −5.179 plants/image, slightly smaller in absolute magnitude than the 5.418 plants/image of the ExG–ExR combination, with lower MAE and RMSE. YOLOv10 + ExG–ExR had an RMSE of 17.653 plants/image, exceeding 14.062 plants/image for HSV; Figure 18 shows large negative error observations for ExG–ExR. RT-DETR Bias values increased from +15.731 to +23.866 and +32.701 plants/image under GLI, ExG–ExR, and HSV, respectively, indicating increasing overcounting. Mamba-YOLO Bias values were −7.896, −13.866, and −9.836 plants/image, with greater undercounting under ExG–ExR.

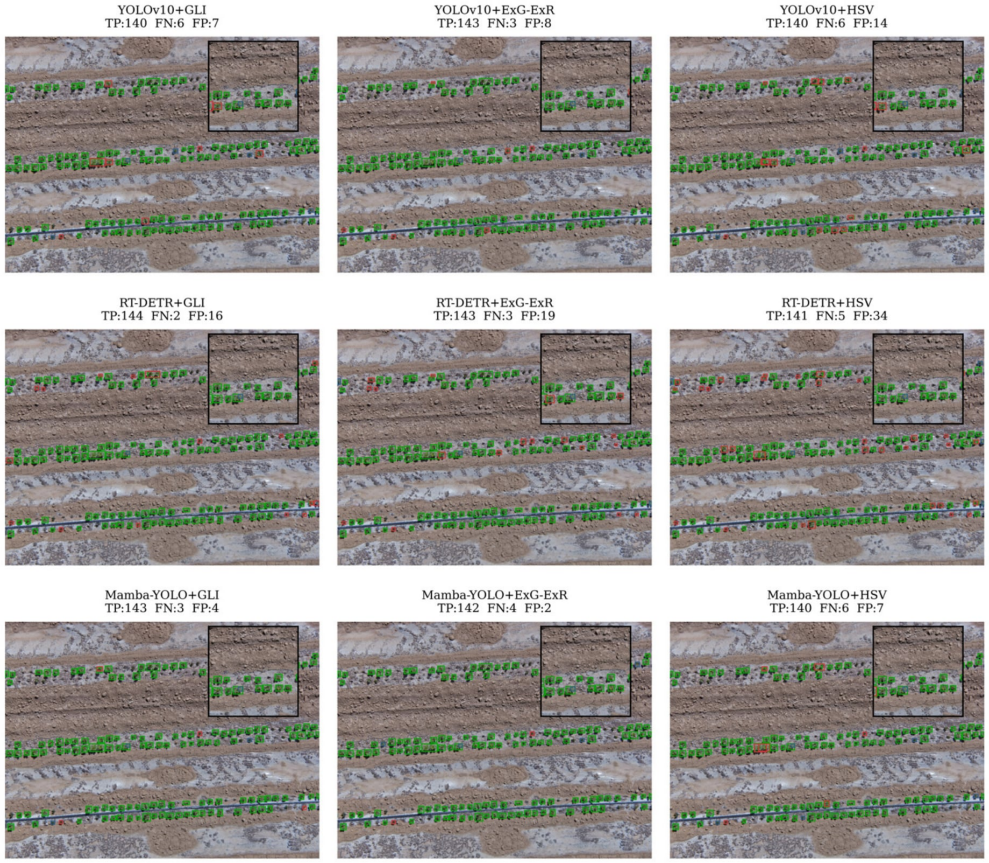


Figure 17. Detection results of nine detector–training label combinations on the same source-region test image. Rows show YOLOv10, RT-DETR, and Mamba-YOLO; columns show GLI, ExG–ExR, and HSV training labels. Green, blue, and red boxes indicate true positives, false negatives, and false positives, respectively; counts are shown above each panel. Upper-right insets magnify the same region. The image contains 146 manually annotated seedlings. RT-DETR + GLI missed 2 and produced 16 false positives, yielding 14 excess counts; Mamba-YOLO + GLI missed 3 and produced 4 false positives, yielding 1 excess count. This example illustrates how false positives and false negatives jointly determine net counting error and does not replace error decomposition across the complete test set.

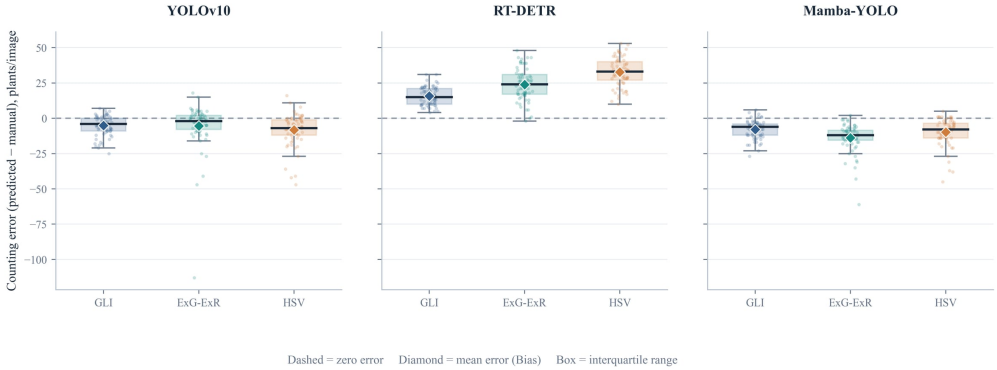


Figure 18. Image-level counting error distributions of nine combinations on S_test. Panels show YOLOv10, RT-DETR, and Mamba-YOLO; blue, teal, and orange represent GLI, ExG–ExR, and HSV. The horizontal axis indicates training label strategies, and the vertical axis is predicted minus manual count (plants/image). The gray dashed line indicates zero error; positive and negative values denote overcounting and undercounting. Box edges represent the first and third quartiles, with the median shown inside. Whiskers extend to the most distant observations within 1.5 interquartile ranges of the box edges. Diamonds indicate mean Bias, and translucent points show image-level errors, including observations beyond the whiskers. All panels share the same vertical axis scale as the leftmost panel.The data are the same as in Table 22.

3.4.4. Connected-Component Merging and Splitting Errors

GLI automatic boxes on S_test were examined. A box enclosing two or more seedlings was classified as a merged component: 112 cases. One seedling represented by two or more boxes was classified as splitting: 22 cases. Merging was much more common, indicating that adjacent seedlings were more likely to be joined and undercounted, whereas splitting could cause overcounting. The two types of cases were counted separately. Post-processing should prioritize separating connected seedlings.

4. Discussion

4.1. The Impact of Illumination Variations on Automatic Annotation Quality

The GLI produced better-quality annotations than standard ExG–ExR and HSV in this study. Mechanistically, training label quality is highly dependent on how spectral indices respond to nonuniform field illumination [33]. ExG–ExR is a normalized channel-difference index; however, in real UAV imagery, channel nonlinearity, saturation, and shadows can still shift its feature distribution and Otsu threshold. Single vegetation indices are sensitive to soil background and illumination, leading to reduced model stability and generalizability [34]. The controlled photometric perturbations (Table 17) revealed that for k = 1.5, ExG–ExR F1 decreased to 0.8629, while GLI remained stable.

Huang et al. [35] incorporated saliency-based region selection and visual quality assessment into a spatial pyramid model for aerial image classification, providing a complementary perspective on accounting for image quality in remote sensing image analysis.

GLI represents relative RGB ratios. In the ideal calculation, a common multiplicative illumination factor cancels pointwise. The small F1 changes observed under synthetic shadows may be associated with quantization and subsequent segmentation. Robustness to real field shadows requires further evaluation.

HSV describes color in images by hue, saturation and value but separation of color and brightness is limited, especially under low illumination [36]. Here, HSV features were processed by shared Otsu thresholding and post-processing to generate candidate boxes and the annotation quality is reported in Table 7. GLI maintained relatively stable F1 scores under the tested brightness and synthetic shadow conditions. The photometric comparison used GLI and ExG–ExR.

4.2. Conditional Comparison and Model Capacity Confound

Detection accuracy decreased for all three models under incomplete training labels, but the magnitude of the decrease varied. On S_test, RT-DETR and Mamba-YOLO trained with GLI labels achieved higher detection accuracy than YOLOv10n. Structural optimization can improve detection performance. For example, Yao et al. [37] improved convolutional modules and increased YOLOv8s mAP@0.5 to 93.28%. Zhang et al. [38] showed that deep neural networks can fit even random labels, while Song et al. [39] reported that label noise can severely impair generalization. However, parameter counts differed substantially: 2.7 M for YOLOv10n, 21.8 M for Mamba-YOLO, and 32.8 M for RT-DETR. This comparison therefore reflects three specific configurations. The observed differences may partly reflect model capacity and cannot be attributed entirely to architectural superiority.

The accuracy decreases in Table 16 are consistent with unannotated targets being treated as background during training, as discussed by Suri et al. [40]. This mechanism may explain the decreases observed in all three models. Previous studies [41–43] have shown that combining local details with global context can improve detection performance. In the present label omission experiment, YOLOv10 showed the largest accuracy decrease, whereas RT-DETR and Mamba-YOLO showed smaller decreases. These results suggest that global contextual modeling may partly mitigate the effects of missing labels. However,

no dedicated ablation experiment was conducted. This explanation therefore remains a hypothesis requiring further validation.

RT-DETR had the highest theoretical computational cost (107.9 GFLOPs) and peak training GPU memory utilization (99.3%), with a training time of approximately 6.9 h. Mamba-YOLO required 49.6 GFLOPs, with peak training GPU memory utilization of 89.5% and a training time of approximately 5.2 h. Using less than half the theoretical computation of RT-DETR, Mamba-YOLO achieved a higher mean mAP@0.5:0.95 (0.8020 versus 0.7928), although the difference was not statistically significant. Considered alongside the work of Pan et al. [44] on resource-constrained settings, these findings may inform model selection. The reported memory values represent peak training consumption and do not directly indicate inference memory requirements. Under the tested conditions, Mamba-YOLO offered advantages over RT-DETR in training time and resource requirements for applications requiring frequent retraining.

4.3. Research Limitations and Future Directions

The data were collected from the source region at Huaxing Farm in 2024 and the cross-region test area in Kuqa City in 2026. Training used only source-region data. The analysis was restricted to visible-light imagery and focused primarily on the seedling stage. Deployment on embedded UAV devices was not evaluated. These conditions limit the scope of the conclusions.

Compared with source-region testing, cross-region mAP@0.5:0.95 decreased by 56.9%, 58.7%, and 72.0% for Mamba-YOLO, YOLOv10, and RT-DETR trained with GLI labels, respectively. Region, year, imaging device, and input target scale changed simultaneously. The results therefore reflect a combined domain shift, and the effect of each factor cannot be isolated. The marked decreases at higher IoU thresholds indicate that cross-region generalization remains a major limitation. Su et al. [45] proposed a continuous UAV localization method based on cross-view perception and particle filtering, showing that multiview information can improve adaptation to environmental changes. Although their task concerned localization rather than crop detection, the multiview fusion approach may inform future research.

Future work could address data, sensors, and deployment. Datasets could be expanded to additional regions, years, and cotton cultivars to assess robustness under broader conditions. Multispectral and other sensor data could be investigated to improve detection in complex backgrounds or at later growth stages. Testing on resource-constrained platforms and model compression could help assess the feasibility of real-time field monitoring. In addition, target-region fine-tuning and other domain adaptation strategies could help reduce cross-region detection accuracy losses.

5. Conclusions

This study evaluated a joint strategy combining GLI automatic annotation, based on visible-band ratios, with three specific deep learning detector configurations (YOLOv10n, RT-DETR-L, and Mamba-YOLO) for cotton seedling detection in UAV RGB images. The main findings are summarized below.

(1) GLI achieved F1 = 96.18% and mean IoU = 89.78%. Averaged across the three detector configurations, GLI labels yielded the highest source-region mAP@0.5:0.95 (0.7773), compared with 0.7461 for ExG–ExR and 0.6792 for HSV.Across three training runs, Mamba-YOLO + GLI achieved the highest mean source-region mAP@0.5:0.95 (0.8020), followed by RT-DETR + GLI (0.7928). However, their paired difference was not statistically significant. In cross-region testing, Mamba-YOLO + GLI achieved the highest mAP@0.5:0.95 (0.3446),

representing a 56.9% decrease from its source-region value (0.7994). Both values were obtained using the same model checkpoint.

(2) First, the joint strategy of GLI automatic annotation and deep learning was tested, demonstrating a substantial reduction in initial annotation workload. Second, a comprehensive comparison restricted to three specific configurations showed that detection and counting performance vary with domain and task. Complexity analysis indicated that Mamba-YOLO achieved a numerically higher source-region mAP@0.5:0.95, although not statistically significant, with less than half the computation of RT-DETR (49.6 G versus 107.9 G) and shorter training time. Due to the large differences in the number of parameters, conclusions are strictly limited to the tested configurations and do not imply absolute architectural superiority.

(3) GLI is a practical tool to obtain high-quality automatically generated annotations. However, the automatically generated annotations need to be manually reviewed to fix omissions and localization errors. Model selection must be task- and region-dependent. Mamba-YOLO + GLI achieved good cross-region detection performance with moderate computational cost; YOLOv10 + GLI had the lowest computational cost and was more suitable for source-region counting. RT-DETR had the highest computational cost and greater deployment requirements.

(4) Future work should aim at evaluating domain adaptation strategies such as target-region fine-tuning and feature alignment to improve cross-region localization, as well as evaluating the strategy across additional years, cotton cultivars and regions to define its generalization limits.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/rs18223984/s1, Section S1: Learning Rate Check; Table S1: Validation metrics at epoch 500 for YOLOv10 on the A_ref validation subset at different initial learning rates.

Author Contributions: Conceptualization, X.W., Q.L. and P.J.; methodology, X.W. and P.J.; software, Q.L. and H.Y.; validation, Q.L., P.W. and S.H.; formal analysis, P.J.; investigation, C.L. and J.W., Z.Y.; resources, X.W.; data curation, C.L. and H.C.; writing—original draft preparation, Q.L.; writing—review and editing, X.W., J.W. and Z.Y.; visualization, Q.L. and H.C.; supervision, X.W.; project administration, X.W.; funding acquisition, P.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Xinjiang Uygur Autonomous Region Key Research and Development Project (2024B03023-2) and the Xinjiang Uygur Autonomous Region Major Science and Technology Special Project (2022A02011-1).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The source code and trained model weights are also available from the corresponding author upon reasonable request.

Acknowledgments: We want to thank the editor and anonymous reviewers for their valuable comments and suggestions for this paper.

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

References

1. Khan, M.A.; Wahid, A.; Ahmad, M.; Tahir, M.T.; Ahmed, M.; Ahmad, S.; Hasanuzzaman, M. World Cotton Production and Consumption: An Overview. In *Cotton Production and Uses: Agronomy, Crop Protection, and Postharvest Technologies*; Ahmad, S., Hasanuzzaman, M., Eds.; Springer: Singapore, 2020; pp. 1–7. [CrossRef]
2. McMaster, G.S.; Palic, D.B.; Dunn, G.H. Soil Management Alters Seedling Emergence and Subsequent Autumn Growth and Yield in Dryland Winter Wheat-Fallow Systems in the Central Great Plains on a Clay Loam Soil. *Soil Tillage Res.* **2002**, *65*, 193–206. [CrossRef]

3. Reddy, K.R.; Brand, D.; Wijewardana, C.; Gao, W. Temperature Effects on Cotton Seedling Emergence, Growth, and Development. *Agron. J.* **2017**, *109*, 1379–1387. [CrossRef]
4. Huang, Y.; Qian, Y.; Wei, H.; Lu, Y.; Ling, B.; Qin, Y. A Survey of Deep Learning-Based Object Detection Methods in Crop Counting. *Comput. Electron. Agric.* **2023**, *215*, 108425. [CrossRef]
5. Song, C.; Ma, W.; Li, J.; Qi, B.; Liu, B. Development Trends in Precision Agriculture and Its Management in China Based on Data Visualization. *Agronomy* **2022**, *12*, 2905. [CrossRef]
6. Tahir, M.N.; Lan, Y.; Zhang, Y.; Huang, W.; Wang, Y.; Naqvi, S.M.Z.A. Application of Unmanned Aerial Vehicles in Precision Agriculture. In *Precision Agriculture*; Zaman, Q., Ed.; Elsevier: Amsterdam, The Netherlands, 2023; pp. 55–70. [CrossRef]
7. Jin, X.; Liu, S.; Baret, F.; Hemerlé, M.; Comar, A. Estimates of Plant Density of Wheat Crops at Emergence from Very Low Altitude UAV Imagery. *Remote Sens. Environ.* **2017**, *198*, 105–114. [CrossRef]
8. Feng, A.; Zhou, J.; Vories, E.; Sudduth, K.A. Evaluation of Cotton Emergence Using UAV-Based Imagery and Deep Learning. *Comput. Electron. Agric.* **2020**, *177*, 105711. [CrossRef]
9. Banerjee, B.P.; Sharma, V.; Spangenberg, G.; Kant, S. Machine Learning Regression Analysis for Estimation of Crop Emergence Using Multispectral UAV Imagery. *Remote Sens.* **2021**, *13*, 2918. [CrossRef]
10. Li, T.; Wang, H.; Cui, J.; Wang, W.; Li, W.; Jiang, M.; Shi, X.; Song, J.; Wang, J.; Lv, X.; et al. Improving the Accuracy of Cotton Seedling Emergence Rate Estimation by Fusing UAV-Based Multispectral Vegetation Indices. *Front. Plant Sci.* **2024**, *15*, 1333089. [CrossRef] [PubMed]
11. Li, B.; Xu, X.; Han, J.; Zhang, L.; Bian, C.; Jin, L.; Liu, J. The Estimation of Crop Emergence in Potatoes by UAV RGB Imagery. *Plant Methods* **2019**, *15*, 15. [CrossRef] [PubMed]
12. Löw, F.; Michel, U.; Dech, S.; Conrad, C. Impact of Feature Selection on the Accuracy and Spatial Uncertainty of Per-Field Crop Classification Using Support Vector Machines. *ISPRS J. Photogramm. Remote Sens.* **2013**, *85*, 102–119. [CrossRef]
13. Liu, M.; Su, W.-H.; Wang, X.-Q. Quantitative Evaluation of Maize Emergence Using UAV Imagery and Deep Learning. *Remote Sens.* **2023**, *15*, 1979. [CrossRef]
14. Lin, Y.; Chen, T.; Liu, S.; Cai, Y.; Shi, H.; Zheng, D.; Lan, Y.; Yue, X.; Zhang, L. Quick and Accurate Monitoring Peanut Seedlings Emergence Rate Through UAV Video and Deep Learning. *Comput. Electron. Agric.* **2022**, *197*, 106938. [CrossRef]
15. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
16. Xie, J.; Wu, H.; Wu, W.; Hong, L.; He, L.; Yu, Q.; Liu, L.; Lin, A.; Som-ard, J. A CNN-Transformer hybrid network with boundary guidance for mapping cropland field parcels from high-resolution remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2026**, *64*, 4401122. [CrossRef]
17. Samiei, S.; Rasti, P.; Ly Vu, J.; Buitink, J.; Rousseau, D. Deep Learning-Based Detection of Seedling Development. *Plant Methods* **2020**, *16*, 103. [CrossRef] [PubMed]
18. Mamat, N.; Othman, M.F.; Abdoulghafor, R.; Belhaouari, S.B.; Mamat, N.; Mohd Hussein, S.F. Advanced Technology in Agriculture Industry by Implementing Image Annotation Technique and Deep Learning Approach: A Review. *Agriculture* **2022**, *12*, 1033. [CrossRef]
19. Zhao, X.; Huang, Y.; Wang, Y.; Chu, D. Estimation of Maize Seedling Number Based on UAV Multispectral Data. *Remote Sens. Nat. Resour.* **2022**, *34*, 106–114.
20. Dhakal, M.; Huang, Y.; Locke, M.A.; Reddy, K.N.; Moore, M.T.; Krutz, L.J.; Gholson, D.; Bajgain, R. Assessment of Cotton and Sorghum Stand Establishment Using UAV-Based Multispectral and DSLR-Based RGB Imagery. *Agrosyst. Geosci. Environ.* **2022**, *5*, e20247. [CrossRef]
21. Hamuda, E.; Mc Ginley, B.; Glavin, M.; Jones, E. Automatic Crop Detection Under Field Conditions Using the HSV Colour Space and Morphological Operations. *Comput. Electron. Agric.* **2017**, *133*, 97–107. [CrossRef]
22. Bouhadjer, M.E.A.; Mazari, S.; Chikr El Mezouar, M. Toward Accurate Weed Detection: Integrating Deep Crop Segmentation and Greenness-Based Analysis. *SSRN* **2025**. [CrossRef]
23. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs Beat YOLOs on Real-time Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 16–22 June 2024*; IEEE: Piscataway, NJ, USA, 2024; pp. 16965–16974. [CrossRef]
24. Wang, Z.; Li, C.; Xu, H.; Zhu, X.; Li, H. Mamba YOLO: A Simple Baseline for Object Detection with State Space Model. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 8205–8213. [CrossRef]
25. Arshad, A.; Raza, M.A.; Zhang, Y.; Zhang, L.; Wang, X.; Ahmed, M.; Habib-ur-Rehman, M. Impact of Climate Warming on Cotton Growth and Yields in China and Pakistan: A Regional Perspective. *Agriculture* **2021**, *11*, 97. [CrossRef]
26. Wang, F.; Fu, Q.; Hong, M.; Tang, W.; Su, L.; Zhu, D.; Wang, Q. Optimization of Cotton Field Irrigation Scheduling Using the AquaCrop Model Assimilated with UAV Remote Sensing and Particle Swarm Optimization. *Agriculture* **2025**, *15*, 1815. [CrossRef]
27. Lu, K.; Fu, W. Analysis of Emergy-Economy Coupling in Maize Farmland Ecosystems Under Nitrogen and Phosphorus Reduction and Optimization of Fertilization Schemes. *Sustainability* **2026**, *18*, 901. [CrossRef]

28. Louhaichi, M.; Borman, M.M.; Johnson, D.E. Spatially Located Platform and Aerial Photography for Documentation of Grazing Impacts on Wheat. *Geocarto Int.* **2001**, *16*, 65–70. [CrossRef]
29. Otsu, N. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
30. Hussain, M. YOLOv5, YOLOv8 and YOLOv10: The Go-to Detectors for Real-Time Vision. *arXiv* **2024**, arXiv:2407.02988. [CrossRef]
31. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011. [CrossRef]
32. Gu, A.; Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *Proceedings of the First Conference on Language Modeling, Philadelphia, PA, USA, 7–9 October 2024*; OpenReview: Amherst, MA, USA, 2024.
33. Li, J.; Wu, W.; Zhao, C.; Bai, X.; Dong, L.; Tan, Y.; Yusup, M.; Akelebai, G.; Dong, H.; Zhi, J. Effects of Solar Elevation Angle on the Visible Light Vegetation Index of a Cotton Field When Extracted from the UAV. *Sci. Rep.* **2025**, *15*, 18497. [CrossRef] [PubMed]
34. Fan, X.; Gao, P.; Zhang, M.; Cang, H.; Zhang, L.; Zhang, Z.; Wang, J.; Lv, X.; Zhang, Q.; Ma, L. The Fusion of Vegetation Indices Increases the Accuracy of Cotton Leaf Area Prediction. *Front. Plant Sci.* **2024**, *15*, 1357193. [CrossRef] [PubMed]
35. Huang, F.; Qi, X.; Li, C.; Hu, W. Aerial Image Classification by Learning Quality-Aware Spatial Pyramid Model. *Future Gener. Comput. Syst.* **2020**, *111*, 271–277. [CrossRef]
36. Yan, Q.; Feng, Y.; Zhang, C.; Wang, P.; Wu, P.; Dong, W.; Sun, J.; Zhang, Y. You Only Need One Color Space: An Efficient Network for Low-Light Image Enhancement. *arXiv* **2024**, arXiv:2402.05809. [CrossRef]
37. Yao, X.; Yang, F.; Yao, J. YOLO-Wheat: A Wheat Disease Detection Algorithm Improved by YOLOv8s. *IEEE Access* **2024**, *12*, 133877–133888. [CrossRef]
38. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding Deep Learning Requires Rethinking Generalization. *arXiv* **2016**, arXiv:1611.03530. [CrossRef]
39. Song, H.; Kim, M.; Park, D.; Shin, Y.; Lee, J.-G. Learning From Noisy Labels With Deep Neural Networks: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 8135–8153. [CrossRef] [PubMed]
40. Suri, S.; Rambhatla, S.; Chellappa, R.; Shrivastava, A. SparseDet: Improving Sparsely Annotated Object Detection with Pseudo-positive Mining. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023*; IEEE: Piscataway, NJ, USA, 2024; pp. 6770–6781. [CrossRef]
41. Wang, C.; Wu, X.; Zhang, Y.; Wang, W. Recognizing Weeds in Maize Fields Using Shifted Window Transformer Network. *Trans. Chin. Soc. Agric. Eng.* **2022**, *38*, 133–142. [CrossRef]
42. Divvala, S.K.; Hoiem, D.; Hays, J.H.; Efros, A.A.; Hebert, M. An Empirical Study of Context in Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009*; IEEE: Piscataway, NJ, USA, 2009; pp. 1271–1278. [CrossRef]
43. Chen, Z.; Zhang, J.; Xu, Y.; Tao, D. Transformer-Based Context Condensation for Boosting Feature Pyramids in Object Detection. *Int. J. Comput. Vis.* **2023**, *131*, 2738–2756. [CrossRef]
44. Pan, J.; Liu, Z.; Li, X.; Luo, X.; Huang, F.; He, M.; Qin, W.; Cai, Y.; Xiao, J.; Li, Y.; et al. An end-to-end dual backbone framework for UAV fish disease detection using optimized GateHCO feature fusion mechanism. *Neural Netw.* **2026**, *203*, 109140. [CrossRef] [PubMed]
45. Su, C.; Yuan, J.; Zheng, E.; Xu, W.; Liu, Z.; Hu, J. CPFL: Resilient continuous UAV localization via cross-view perception and particle filtering. *Drones* **2026**, *10*, 437. [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.