


Article

Advanced Cotton Boll Segmentation, Detection, and Counting Using Multi-Level Thresholding Optimized with an Anchor-Free Compact Central Attention Network Model

Arathi Bairi ^{1,*} and Uma N. Dulhare ² 

¹ Department of Computer Science and Engineering, Kamala Institute of Technology and Science, Jawaharlal Nehru Technological University, Huzurabad, Karimnagar 505468, India

² Department of Computer Science & Artificial Intelligence, Muffakham Jah College of Engineering & Technology, Hyderabad 500034, India; prof.umadulhare@gmail.com

* Correspondence: artibairi@gmail.com

Abstract: Nowadays, cotton boll detection techniques are becoming essential for weaving and textile industries based on the production of cotton. There are limited techniques developed to segment, detect, and count cotton bolls precisely. This analysis identified several limitations and issues with these techniques, including their complex structure, low performance, time complexity, poor quality data, and so on. A proposed technique was developed to overcome these issues and enhance the performance of the detection and counting of cotton bolls. Initially, data were gathered from the dataset, and a pre-processing stage was performed to enhance image quality. An adaptive Gaussian-Wiener filter (AGWF) was utilized to remove noise from the acquired images. Then, an improved Harris Hawks arithmetic optimization algorithm (IH²AOA) was used for segmentation. Finally, an anchor-free compact central attention cotton boll detection network (A-frC²AcbdN) was utilized for cotton boll detection and counting. The proposed technique utilized an annotated dataset extracted from weakly supervised cotton boll detection and counting, aiming to enhance the accuracy and efficiency in identifying and quantifying cotton bolls in the agricultural domain. The accuracy of the proposed technique was 94%, which is higher than that of other related techniques. Similarly, the precision, recall, F1-score, and specificity of the proposed technique were 93.8%, 92.99%, 93.48%, and 92.99%, respectively.

Keywords: cotton boll detection; segmentation; Wiener filter; Gaussian filter; Harris Hawks optimization; anchor-free compact attention



Citation: Bairi, A.; Dulhare, U.N. Advanced Cotton Boll Segmentation, Detection, and Counting Using Multi-Level Thresholding Optimized with an Anchor-Free Compact Central Attention Network Model. *Eng* 2024, 5, 2839–2861. <https://doi.org/10.3390/eng5040148>

Academic Editors: Juvenal Rodriguez-Resendiz, Akos Odry, José Manuel Álvarez-Alvarado and Marco Antonio Aceves-Fernandez

Received: 12 August 2024
Revised: 27 October 2024
Accepted: 28 October 2024
Published: 1 November 2024



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

1. Introduction

Nowadays, cotton, often called “white gold”, ranks among the most widely grown and important cash crops globally [1]. Its fiber value and quality are crucial to the textile industry [2]. It accounts for nearly 35% of the world’s total fiber production, with China being one of the largest harvesters [3]. From 2022 to 2023, India was the second largest cotton producer, contributing 5.84 million metric tons, which is approximately 23.83% of the world’s total cotton production [4]. Measuring cotton traits accurately is essential to improve fiber yield and quality, and cotton bolls have a significant impact on yield and quality [5]. Agrofield operations, such as fertilization and insect control, directly affect yield and quality, and accurate cotton boll counting is vital for smart agriculture, informing management decisions to improve quality and productivity [6,7].

Book counting is the earliest conventional cotton boll counting method. However, manually counting cotton bolls in the field is a time-consuming and complex task that necessitates a large investment of labor, time, and equipment [8]. This technique is time-consuming and prone to human error, making precise results practically unattainable, particularly when dealing with a large number of facilities [9]. Furthermore, manual

calculations are typically limited to small-scale studies and are inferior in large-scale commercial agriculture. Inefficient and inaccurate boll counts impede the ability to make informed decisions, improve crop management, and boost yields [10]. To address this issue, new methodologies are required to expand the possibility of cotton boll phenotyping.

Furthermore, computer vision is being used extensively in agriculture due to growing interest in the automated intelligent assessment of agriculture. The rising energy consumption to address complicated problems and falling equipment costs are the main drivers of this growth. Finding information on cotton growth has historically been a labor-intensive, hand-crafted procedure requiring skill and attention to detail [11]. While cotton can be effectively managed with computer vision, integrating artificial intelligence (AI) and image processing for boll detection can enhance efficiency and reduce errors associated with manual inspection [12]. Improving crop forecasts and disease detection using computer vision is a promising approach. However, it is crucial to address the limitations of computer vision when analyzing large amounts of data: while it can provide high accuracy with limited data, its performance may degrade when processing large datasets [13].

Cotton boll detection uses traditional machine learning approaches such as support vector machine (SVM), random forest (RF), naïve Bayes (NB), decision tree (DT), and k-nearest neighbors (KNN) [14]. For example, open boll candidates were identified via binary classification using automatic and morphological filters based on the features of each development zone [15]. A segmentation technique to identify cotton in the YCbCr color space image was also derived [16]. Other methods include creating regions with noise applications based on the Wasserstein distance using simple linear iterative clustering (SLIC) and density-based spatial clustering of applications with noise (DBSCAN) [17]. Each region is analyzed to extract histogram-based color and texture features, which are then used for output prediction through a random forest model [18]. It is generally designed for simple images that typically have a single purpose and uniform background, making them easier to analyze. In contrast, real-world scenes contain a variety of objects, occlusions, lighting effects, and reflections, which introduce significant complexity [19]. These conditions can reduce the accuracy and power of cotton boll detection when using conventional imaging techniques.

Nowadays, the application of deep learning models has revolutionized cotton boll detection [20], with notable architectures such as multi-column convolutional neural networks (MCNNs) [21], dilated convolutional neural networks (CSRNNs) [22], scale aggregation networks (SANets) [23], VGG16, InceptionV3, ResNet34 [24], and Faster R-CNN Inception-V2 [25] demonstrating promising results. These models use a variety of techniques to enhance detection accuracy, including capturing features at different sizes, enhancing contextual understanding, and better identifying complex shapes under changing environmental conditions. However, they also face several limitations, including data variability, high computational demands, challenges in generalization, and difficulties with the precise localization of bolls, particularly in crowded and uneven fields. Furthermore, their black-box design causes latency issues and interpretability concerns, which hinder their real-time application. To overcome these limitations, a new model must be developed that improves the robustness, efficiency, and applicability of optimal cotton boll detection solutions. To motivate, this study proposes creating cotton boll segmentation, detection, and computing methods that combine multi-level thresholding and optimization with an anchor-free compact central attention network model, resulting in accurate and effective instant cotton boll detection. The goal is to provide accurate and effective instant cotton boll detection, thereby improving the overall efficiency of the detection process. This research contributes to the field of cotton boll segmentation, detection, and counting in several key ways:

- (i) To identify cotton boll segmentation, detection, and counting, multi-level thresholding optimized with an anchor-free compact central attention network model is introduced;
- (ii) To remove noise and improve data quality using an adaptive Gaussian–Wiener filter (AGWF);

- (iii) To segment cotton boll regions using a multi-thresholding-based hybrid Harris Hawks arithmetic optimization (Mul-TH³AO);
- (iv) To propose a novel model for cotton boll detecting and counting using an anchor-free compact central attention cotton boll detection network (A-frC²AcbdN), which distinguishes cotton bolls from non-cotton objects and accurately counts the bolls.

The remaining content of this paper is organized in the following manner. Section 2 includes a related survey of the various techniques of cotton boll segmentation, detection, and counting. Section 3 describes the details and process of the overall proposed method with images. The results and discussion of the proposed methodology are briefly explained with graphs in Section 4. Finally, Section 5 describes the overall conclusion of the paper.

2. Related Works

This section reviews and analyzes the various techniques related to cotton boll segmentation, detection, and counting, which are described as follows.

Yu Jiang et al. [26] developed a DL-based approach named DeepFlower to detect and count individual emerging blooms on plants using a dataset. This approach processes each image to count the blooms of individual plants over a flowering period. The root mean square error (RMSE) of this technique was found to be 0.79 for detecting and counting emerging blooms on cotton plants. It is important to note that this technique is specifically designed to detect blooms on cotton plants.

Nasseb Singh et al. [27] developed an image-processing algorithm for in-field cotton boll detection under natural lighting conditions. The image quality was enhanced using color difference, the YCbCr method, band ratio, and chromatic algorithms to enable real-time cotton boll segmentation under natural outdoor light conditions. This technique achieved a higher identification rate of 91.05%, with false-positive and false-negative rates of 6.99% and 4.88%, respectively.

Mengli Zhang et al. [28] developed a method based on the YOLO v8 framework for transfer learning called YOLO small-scale pyramid depth-aware detection (SSPD). This method combines space-to-depth and non-strided convolution (SPD-Conv), and a small target detector head integrates a parameter-free attentional mechanism (SimAM) to enhance target boll detection accuracy. This technique achieves a 12.38 RMSE, and the performance of this technique resulted in a higher loss rate.

RuiXu et al. [29] developed a CNN that is trained to detect cotton blooms in raw images. Here in this work, a dense point cloud contrasted from an aerial image with the structure from motion method was utilized to evaluate the 3D location of a cotton bloom. A constrained clustering algorithm was developed to register the same bloom detected from various images based on the 3D location of the bloom. The precision of this technique was 90%, and limited datasets were utilized in this model. The survey of existing techniques is described in Table 1.

Table 1. Survey of related techniques with their performance and limitations for cotton boll detection.

Author Name and Reference	Technique Used	Performance	Limitation
Yu Jiang et al. [26]	DeepFlower	RMSE—0.79	This technique is only utilized to detect the blooms on cotton plants.
Nasseb Singh et al. [27]	YCbCr method, band ratio, and chromatic algorithm	Identification rate—91.05%	Unable to separate overlapped cotton bolls.
Mengli Zhang et al. [28]	YOLO SSPD	12.38 RMSE	Higher loss rate.
RuiXu et al. [29]	CNN	90% Precision	Limited datasets were utilized in this model.

The existing methods for cotton boll segmentation, detection, and counting, as described in [26–29], have several limitations that hinder their accuracy and effectiveness.

These limitations include a limited applicability to detect blooms on cotton plants, insufficient features considered, inability to separate overlapped cotton bolls, higher loss rate, and limited datasets. To overcome these limitations, a new model is required that can improve applicability, incorporate more features, separate overlapped cotton bolls, reduce loss rate, and utilize a larger and more diverse dataset. By addressing these limitations, a new model can be developed that provides more accurate and reliable cotton boll segmentation, detection, and counting.

3. Materials and Methods

Nowadays, cotton boll segmentation, detection, and counting are important for estimating crop yield, monitoring crop health, and making informed decisions in precision farming and automation. The proposed approach for cotton boll counting and segmentation involves the integration of advanced multi-thresholding and an anchor-free compact central attention network. This combination of techniques aims to overcome the limitations of the existing methods [26–29]. The proposed model enhances the quality and yield of cotton production, ultimately benefiting the industry and meeting the growing market demand. Figure 1 displays the graphical representation of the proposed method.

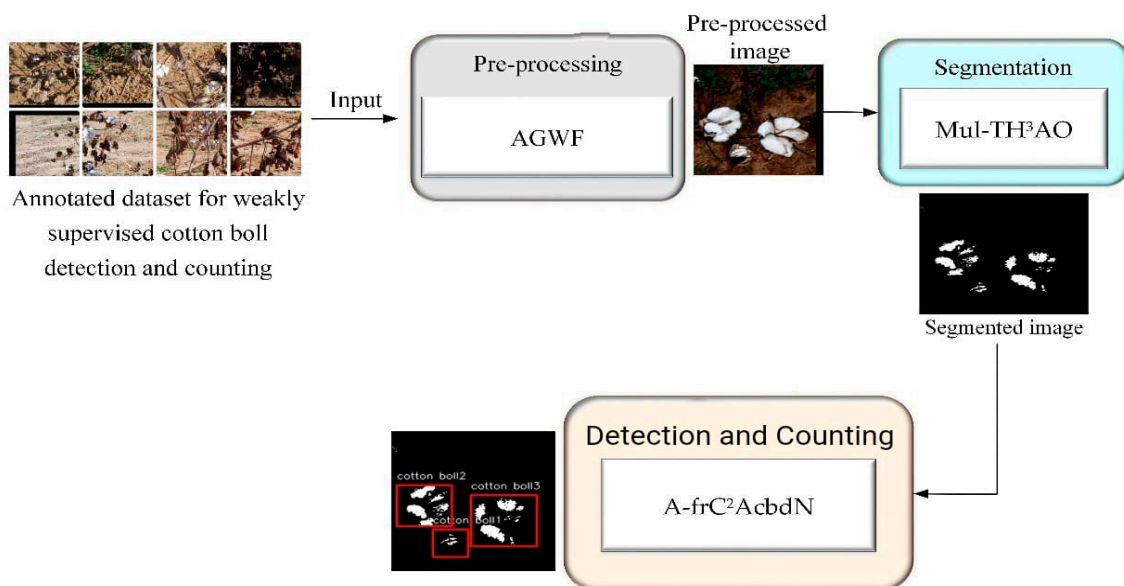


Figure 1. Graphical representation of the proposed method.

Here, the proposed method involves several key phases, such as image acquisition, pre-processing, segmentation, detection, and counting. Initially, the cotton plant images are collected from publicly available annotated datasets mainly designed for cotton boll detection and counting. Then, the collected images are fed into the pre-processing step using an adaptive Gaussian–Wiener filter (AGWF) to remove noise and improve the input image quality. Next, the pre-processed images are fed into the segmentation process using multi-thresholding-based hybrid Harris Hawks arithmetic optimization (Mul-TH³AO) to segment the cotton boll regions. Finally, the segmented images are input into the anchor-free compact central attention cotton boll detection network (A-frC²AcbdN) model to effectively detect and count the cotton bolls. The above-mentioned methods are described in detail in the following sections.

3.1. Image Acquisition

This study utilized an annotated dataset for weakly supervised cotton boll detection and counting, which is publicly available for download at Figshare (https://figshare.com/articles/dataset/Annotated_dataset_for_weakly_supervisedCotton_boll_detection_and_counting/19665096?file=34923654 (13 April 2024)). The dataset comprises RGB images of cot-

ton plants captured in various environments, including both potted (indoor and outdoor) and in-field settings, showcasing a range of plant conditions and backgrounds. The images were taken using a hand-held consumer-grade camera.

To ensure quality, a pre-screening process was implemented to select high-quality images, resulting in a total of 350 images, each with a resolution of 500×500 pixels. These images were subsequently resized to 256×256 pixels for processing. The dataset was divided into two subsets: 300 images for training (total size: 33.0 MB) and 50 images for testing (total size: 4.45 MB).

For the task of cotton boll detection and counting, Python was employed as the programming language, utilizing prominent libraries such as os, cv2, numpy, tensorflow, matplotlib, scipy, and pandas. These libraries were selected for their established reliability, widespread use, and efficiency in handling specific tasks, facilitating the development and testing of the model.

The dataset accommodates diverse data collection methods, making it adaptable for various applications. It includes in-field, indoor, proximal, aerial, RGB, and multispectral collection methods, which capture images in different settings and resolutions. These methodologies allow researchers to ensure the reproducibility and comparability of their findings, resulting in more accurate machine learning models for cotton boll detection and counting. Figure 2 depicts some sample images of the annotated dataset for weakly supervised cotton boll detection and counting.

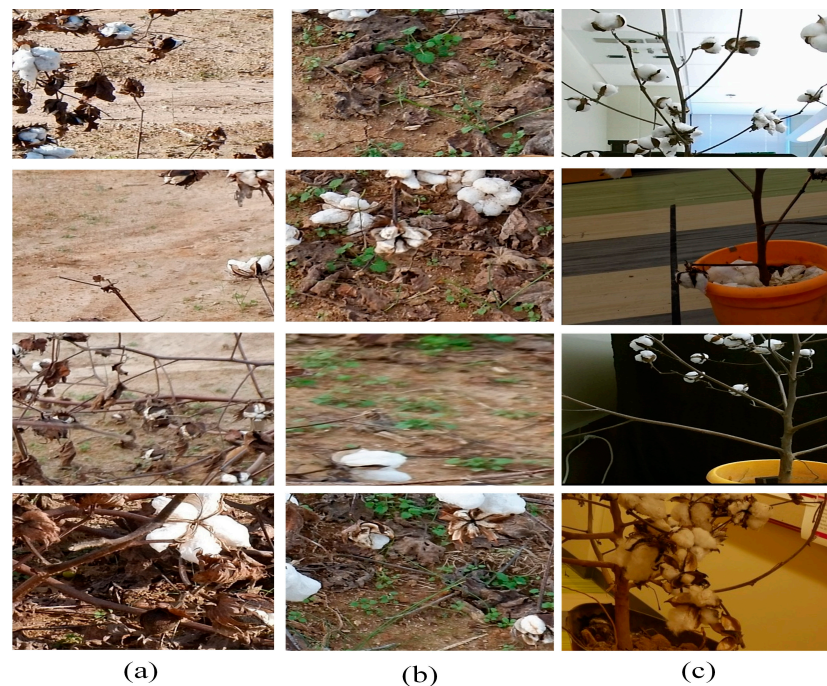


Figure 2. Sample images of the annotated dataset for cotton boll detection and counting: (a) in-field plants; (b) outdoor potted plants; and (c) indoor potted plants.

3.2. Image Pre-Processing

This study proposed an AGWF for filtering noise within the input cotton plant images. Specifically, the combination of Wiener and Gaussian filters was explored.

The Gaussian filter is employed to enhance the efficiency of image smoothing [30]. The initial stage of the Gaussian filter involves the detection of noise, which is not particularly successful in removing salt-and-pepper noise. The analysis was conducted using the

Gaussian distribution. The Gaussian distribution’s probability density function ($G(a)$) is described in the below equation:

$$G(a) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(a-\mu)^2}{2\sigma^2}} \quad (1)$$

Here, the gray-level image is denoted as a , the mean value is represented as μ , and the standard deviation is represented as σ . The amount of smoothing is determined by the standard deviation (σ). Then, the Gaussian filter output is fed to the Wiener filter.

The Wiener filter [31] is a linear filter designed to reduce the mean square error between the original and filtered signal. It is intended to improve photos that have been corrupted by additive noise. The Wiener filter balances noise reduction and image detail retention through parameter adjustments. The Wiener filter equation is illustrated in the following equations:

$$K(x, y) = G(x, y)A(x, y) \quad (2)$$

$$G(x, y) = \frac{S^*(x, y)F_m(x, y)}{|S^*(x, y)|^2 F_m(x, y) + F_n(x, y)} \quad (3)$$

$$G(x, y) = \frac{S^*(x, y)}{|S^*(x, y)|^2 + \frac{F_n(x, y)}{F_m(x, y)}} \quad (4)$$

Here, $F_m(x, y)$ represents the power spectrum of the signal process, and $F_n(x, y)$ denotes the spectrum of the noise process. Equation (4) can be obtained by dividing through F_m in Equation (3). Finally, the AGWF method reduces noise and improves the quality of the original cotton plant image. Figure 3 displays the flow chart of the AGWF method.

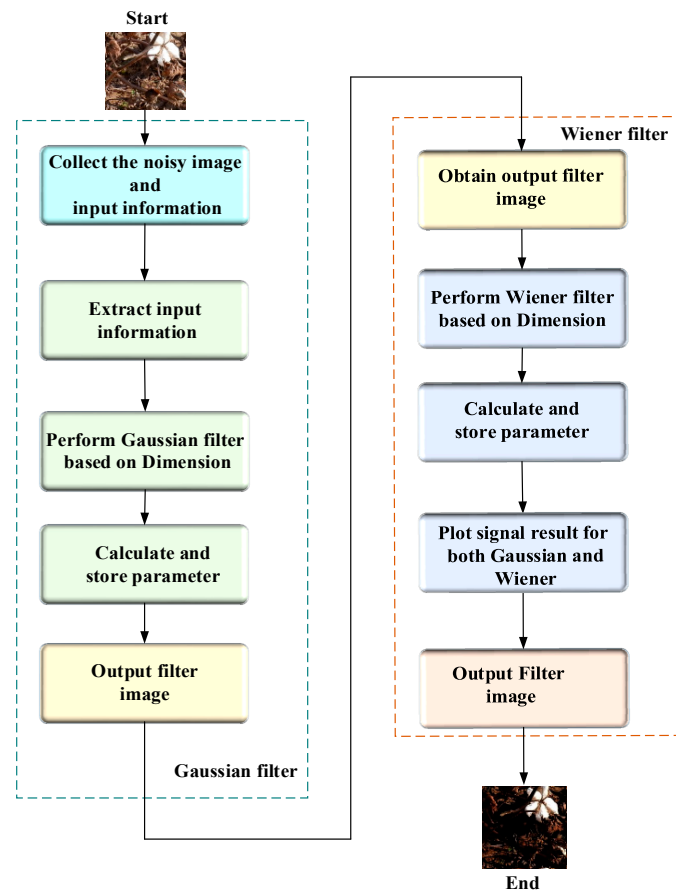


Figure 3. Flowchart of the adaptive Gaussian–Wiener filter method.

3.3. Segmentation Using the Mul-TH³AO Algorithm

The Mul-TH³AO algorithm segments cotton boll regions using a combination of a fitness function and Harris Hawks optimization (H²O) [32] and arithmetic optimization (AO) [33] algorithms. These algorithms start with a set of random solutions, which are iteratively evaluated and improved using a fitness function.

3.3.1. Fitness Function

The fitness function helps determine the optimal threshold vector for classifying image pixels, similar to the Otsu thresholding [34] method. The method uses the image’s histogram to accurately define its boundaries. The fitness function is based on the following equation:

$$T = \sqrt{\left| \sum_{a=0}^Y SUM_a \left((Y + 1)\mu_a - \frac{\mu_1}{Y + 1} \right) + (\mu_a - \mu_1)^3 \right|} \tag{5}$$

The average image density of an image μ_1 is represented as X in $R_1 = 0$ and $R_2 = A$ (X in $R_1 = 0$ and $R_2 = A$). The number of searched thresholds is represented as Y . μ_a denotes the average density of class F_a in R_a and $R_{a+1} - 1$, and the probabilities are denoted as SUM_a :

$$SUM_a = \sum_{b=R_a}^{R_{a+1}-A} G_b \tag{6}$$

$$\mu_a = \sum_{b=R_a}^{R_{a+1}-A} a \frac{G_b}{SUM_a} \quad \text{where } G_b = z(b) / NUM_G \tag{7}$$

Here, the gray-level b probability is denoted as G_b , the gray-level b frequency is denoted as $z(b)$, and the total number of pixels in an image is denoted as NUM_G . This ensures a high chance of obtaining the best optimal global solution for cotton boll segmentation.

The segmentation process is divided into two main phases, the exploration phase and the exploitation phase, carried out over 100 iterations. The AO algorithm is responsible for the first 50 iterations, focusing on wide-scale exploration of the solution space. Subsequently, the H²O algorithm performs the remaining 50 iterations, refining the solutions by focusing on exploitation to optimize the threshold values for better segmentation accuracy.

3.3.2. Exploration Phase

In the exploration phase, AO [33] is used to explore the image histogram and identify the best threshold vector, which segments the image into relevant regions (cotton bolls and background). The AO algorithm operates by evaluating the image histogram, treating each solution as a vector that corresponds to various pixel brightness levels.

The pixel brightness values from the image histogram are mapped onto solution vectors, where each vector (\vec{K}_a) contains the threshold values representing pixel intensities:

$$\vec{K}_a = (k_{a,1}, k_{a,2}, \dots, k_{a,y}), \quad 0 \leq k_{a,1}, k_{a,2}, \dots, k_{a,y} \leq T \tag{8}$$

Here, $k_{a,y}$ denotes each threshold of the vector, and T denotes the maximum brightness of the pixels in the images. The AO algorithm uses a fitness function (Otsu thresholding) to assess the quality of each solution vector by calculating the difference between foreground (cotton boll) and background pixel intensities. To improve the exploration phase, AO is described in [33]. The AO updates the solution vectors across iterations using the following equations:

$$k_{a,b}(F_{Iter} + 1) = \begin{cases} \text{best}(k_b) \div (Mop + \epsilon) \times ((OD_b - CD_b) \times \mu + CD_b) & Z_2 < 0.5 \\ \text{best}(k_b) \times MOP \times ((OD_b - CD_b) \times \mu + CD_b) & \text{otherwise} \end{cases} \tag{9}$$

Here, the solution of another iteration is denoted by $k_{a,b}(F_{Iter} + 1)$, which corresponds to the threshold vector in the segmentation problem. In a subsequent iteration $k_{a,b}(F_{Iter})$, the solution for the b th candidate in the a th dimension is represented as such, while the best solution ever achieved is denoted as the best b th solution. The upper limit of the b th solution is represented by OD_b , and the lower limit values of the b th solution are represented by CD_b . This process continues until the 50th iteration, at which point the exploration phase ends.

3.3.3. Exploitation Phase

In the exploitation phase, the H²O [32] algorithm runs for the next 50 iterations (from iteration 51 to 100) to further refine the solutions produced by the AO algorithm. The main goal here is to improve the threshold vector for cotton boll segmentation. The H²O algorithm mimics the hunting strategy of Harris Hawks, where the prey represents the optimal threshold vector in the segmentation process.

During this phase, the H²O algorithm fine-tunes the threshold vectors by minimizing the fitness function, which improves the clarity of the cotton bolls in the image. Each solution is a vector of pixel brightness values that distinguishes different levels of cotton boll visibility in the segmentation.

Every solution is denoted as a vector according to the MOA function in the AO algorithm [34]. The solutions are updated as mathematically expressed in Equation (10):

$$\vec{K}(r + 1) = \begin{cases} \vec{K}_{rand(r)} - e_1 \left| \vec{K}_{rand(r)} - 2e_2 \vec{K}(r) \right|, & j \geq 0.5(K_{rabbit}(r) - K_n(r)) - e_3(CD + e_4(OD - CD)), \\ j < 0.5 \end{cases} \quad (10)$$

Here, the $(r + 1)$ th iteration next-solution vector is denoted as $\vec{K}(r + 1)$. The r th iteration bait positions are denoted as $K_{rabbit}(r)$, and the r th iteration vector current solution is denoted as $\vec{K}(r)$. The variables e_1, e_2, e_3, e_4 , and j are random numbers in the range $(0, 1)$ that are updated in each iteration. OD and CD denote the upper and lower limits for variables in the multi-thresholding segmentation problem, corresponding to the image's pixels' maximum and lowest brightness levels. $\vec{K}_{rand}(r)$ is used to represent the solution vector that was chosen at random from the current population, and $K_n(r)$ denotes the current population average of the solutions.

The escape energy (W) of the cotton boll (bait) influences the movement of the hawks (solutions) in H²O algorithms. The hawks decide whether to approach the cotton boll aggressively or cautiously based on this energy, which is modeled by the following equation:

$$W = 2W_0(1 - r/R) \quad (11)$$

Here, escape energy is denoted as W_0 , and R denotes the maximum repetitions. The value of W_0 changes randomly in every iteration, ranging from -1 to 1 . When W_0 decreases (ranging from 0 to -1), the cotton boll's ability to escape diminishes, leading to a "soft besiege", where the hawks move cautiously. Conversely, when W_0 increases toward 1 , this triggers a "hard besiege", where the hawks make an aggressive move to the prey (cotton boll).

As the iterations progress, W gradually decreases. When $|W| < 1$, the H²O algorithm enters the exploitation phase. During this phase, if $r \geq 0.5$ and $0.5 \geq |W|$, the hawks apply a soft besiege, where they update the threshold vector in the cotton boll segmentation process gradually, represented by the following equations:

$$\vec{K}(r + 1) = \Delta K(r) - W \left| BK_{rabbit}(r) - \vec{K}(r) \right| \quad (12)$$

Here, $\Delta K(r)$ represents the difference between the bait position (the cotton boll threshold vector) and the current solution in the r th iteration. Additionally, e_5 is a random number between 0 and 1 , representing the unpredictable nature of the cotton boll's escape.

B represents the strength of the unpredictable nature of the cotton bolls at all stages of escape. Each time, the value of B is altered randomly to replicate the erratic behaviors of cotton boll movements.

When $r \geq 0.5$ and $0.5 < |W|$, the hawks initiate a hard besiege and move aggressively toward the optimal threshold vector. This aggressive movement is mathematically expressed by the following equation:

$$\vec{K}(r + 1) = K_{rabbit}(r) - W|\Delta K(r)| \tag{13}$$

When $r < 0.5$ and $0.5 \geq |W|$, the bait accumulates sufficient energy, and the timing is right, and a soft besiege with progressive rapid dives is executed before the cotton boll attempts an unexpected escape. The hawks continue to their positions during this phase by following the soft besiege equations.

$$\vec{K}(r + 1) = \begin{cases} Q, & \text{if } Z(Q) < Z(K(r)) \\ O, & \text{if } Z(O) < Z(K(r)) \end{cases} \tag{14}$$

When $r < 0.5$ and $0.5 < |W|$, before the unexpected leap, the bait lacks the energy to break free (hard besiege) with progressive rapid dives. The solutions attempt to reduce the distance between their current positions and the escaped cotton boll threshold.

$$\vec{K}(r + 1) = \begin{cases} Q, & \text{if } Z(Q) < Z(K(r)) \\ O, & \text{if } Z(O) < Z(K(r)) \end{cases} \tag{15}$$

As a result, until the stop condition is satisfied, this procedure is repeated. The optimal threshold vector that is obtained is then represented by the algorithm's output. The Mul-TH³AO algorithm's diagram for determining the threshold vector in the cotton boll segmentation is displayed in Figure 4 and Algorithm 1.

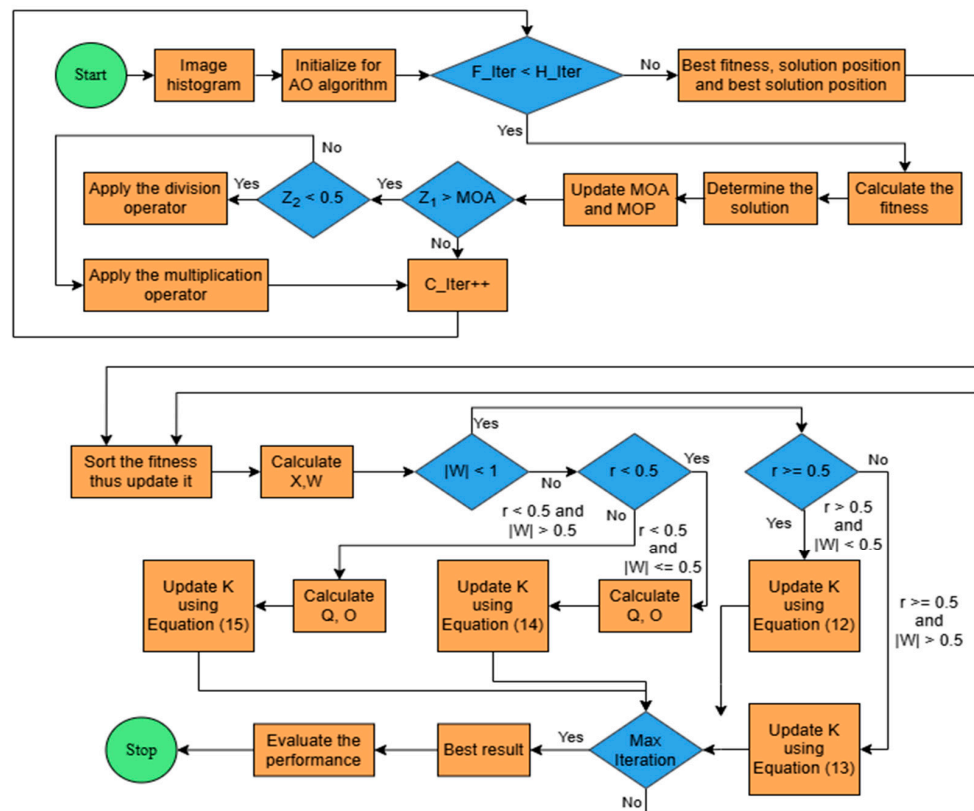


Figure 4. Mul-TH³AO segmentation algorithm flowchart.

Algorithm 1. Pseudo-code of the Mul-TH³AO algorithm

```

Input: Image I, Maximum Iterations  $Max\_Iter$ 
Output: Optimal Threshold Vector  $K^*$ 
Initialize random solutions (threshold vectors)
Define parameters for AO and H2O algorithms
Initialize iteration counter  $Iter = 1$ 
//Fitness Function Definition
Function Fitness ( $X, T$ )
Calculate the histogram of an image  $X$ 
Compute average image density and class densities
    Compute probabilities and frequencies of gray levels
    Calculate fitness value using Equation (5)
    Return fitness value
//Exploration Phase (AO Algorithm)
While  $Iter \leq 50$  [ $F\_Iter < H\_Iter$ ] do
    Find the best solution (determined best so far)
    Update the MOA value
    Update the MOP value
For ( $a = 1$  to solution ) do
    if  $Z_1 > MOA$  then
        if  $Z_2 < 0.5$ 
            Update the  $a^{th}$  solutions' positions using the division operation in Equation (9)
        Else
            Update the  $a^{th}$  solutions' positions using the multiplication operation in Equation (9)
        End if
    End if
End For
End While
// Exploitation Phase (H2O Algorithm)
While  $Iter \leq 100$  do
    For each solution vector  $\vec{K}_a$  in the solutions
        IF  $|W| < 1$  then
            Assign a random number from [0, 1] to  $r$ 
            If  $r \geq 0.5$  and  $|W| \leq 0.5$  then soft besiege
                Update the location vector using Equation (12)

            Else If  $r \geq 0.5$  and  $|W| > 0.5$  then execute hard besiege
                Update the location vector using Equation (13)

            Else If  $r < 0.5$  and  $|W| \leq 0.5$  then execute soft besiege
                with progressive rapid dives
                Update the location vector using Equation (14)

            Else If  $r < 0.5$  and  $|W| > 0.5$  then execute hard besiege
                with progressive rapid dives
                Update the location vector using Equation (15)
        End if
    End if
End for
End While
//Output the best solution
 $K^*$  = best solution found
Return  $K^*$ 

```

3.4. Cotton Boll Detection and Counting Using the A-frC²AcbdN Model

In this study, a novel A-frC²AcbdN model is developed specifically for cotton boll detection and counting. Figure 5 illustrates the proposed A-frC²AcbdN model's architecture.

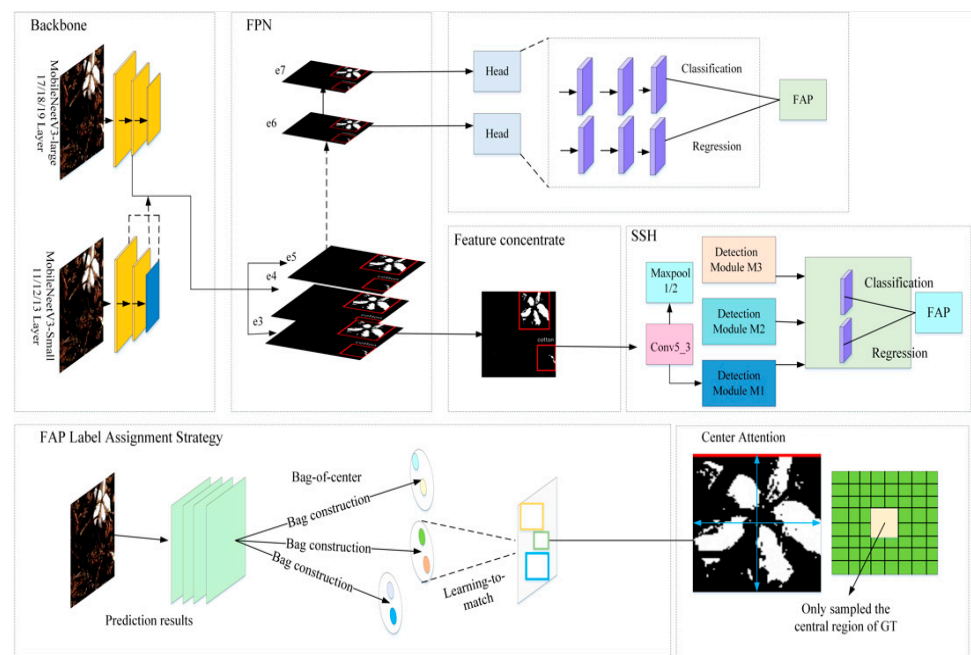


Figure 5. A-frC²AcbdN model.

Here, the A-frC²AcbdN model is built on the fully convolutional one-stage (FCOS) framework, which incorporates MobileNetV3 into a feature pyramid network (FPN) to improve feature extraction. The detecting heads are optimized to minimize the overall model parameters while increasing detection speed during inference. In addition, a tiny object improvement module is developed, which will use a single-stage headless (SSH) face detector to target small objects. This is followed by a scale-invariant network topology that enhances small object recognition performance. To design the loss function, a unique label assignment strategy is developed, which addresses the anchor bounding box matching requirement while enhancing the flexibility of bag-of-center building. Finally, a center correction method is used in the post-processing stage to modify the predicted bounding boxes, resulting in increased accuracy. The A-frC²AcbdN model is explained in detail below.

3.4.1. Compact Weight Backbone Network

The FCOS model avoids complex anchor calculations, significantly reducing overhead and improving detection performance compared to anchor-based detectors. To further streamline the network, MobileNetV3 [35] replaces the original ResNet [36] backbone. MobileNetV3's lightweight structure enhances inference efficiency, making it a better choice for the FCOS backbone.

In this approach, the residual block of ResNet has been substituted with the Bneck structure from MobileNetV3 to serve as the FPN input. Various strategies are implemented to accommodate different backbone versions:

- (i) If using the large version, the backbone and the 17th Bneck are inputs to the FPN.
- (ii) If using the small version, the 11th, 12th, and 13th Bneck are inputs to the FPN.

This compact weight backbone network improves detector performance for small objects by addressing information loss during small object detection.

3.4.2. Small Object Enhancement Module

A new small object optimization module has been created to improve small object detection. This model takes advantage of single-stage headless (SSH) [37] face detection, which can work well on different products and expand the understanding of the model. This module improves the model's ability to detect small objects by integrating SSH.

In the module, the feature pyramid network extracts features at different scales and then focuses them on a unified output feature vector. The feature vector is then input into a small target detection module, which includes a detection module and a context module. The detection module includes classifiers and regressors to detect and localize small objects, while the identification module uses large filters to increase the size of the window around the object. Since the difference between the parameters will not affect the detection accuracy, only anchor poles with a ratio of 1 are stored during detection. Finally, the resulting anchors are combined and filtered through the anchor processing stage and further refined using the target mapping scheme described in the next section.

3.4.3. Label Assignment Strategy

A label assignment strategy has been proposed to select the optimal anchor point for each item, inspired by the label assignment technique introduced in FreeAnchor [38]. FreeAnchor employs a self-learning object identification method to align the anchor box with each object. Its objective is to remove manually created anchor divisions while optimizing three crucial visual object detection metrics. To obtain high recall, the detector must first make sure that each object has at least one correct anchor box prediction. Second, correctly identifying the anchor box with the least regression error should attain high accuracy. Third, the anchor box's prediction must comply with non-maximum suppression; otherwise, precise-positioned predictions with low classification scores might be suppressed. FreeAnchor's loss function is defined by

$$P(\theta) = -\omega_1 \sum_x \log(\text{Mean} - \max(M_x)) + \omega_2 LF(R\{t_y \in T_-\}) (1 - R_{xy}^{uv}(\theta)) \quad (16)$$

$$M_x = \left\{ R_{xy}^{class}(\theta) R_{xy}^{pos}(\theta) | t_y \in T_x \right\} \quad (17)$$

Here, θ denotes the network parameter, and the balance factor is denoted by ω_1 and ω_2 . The mean-max method is used to select the best anchor box for each object from a given set of anchor boxes. At the start of training, most anchor boxes are used. However, as the training advances and the network becomes fully trained, the confidence in specific anchor boxes grows. The likelihood set of the anchor bag T_x is denoted by M_x , the focal loss is denoted by LF , T_x denotes the x^{th} anchor box set, $T_- \in T$ denotes the negative sample set, $R\{t_y \in T_-\}$ represents the probability that the t_y of an anchor does not match with any object, uv denotes the background, $R_{xy}^{class}(\theta)$ denotes the classification confidence, $R_y^{uv}(\theta)$ denotes the background confidence, and $R\{t_y - w_x\}^{xy}$ represents the probability that t_y accurately predicts w_x .

However, the current method of using anchor tags for a free-anchor search is not effective. The accuracy of detection is improved by optimizing the assignment of the free-anchor label. This is achieved by using the mean-maximum function to calculate the IoU of each anchor point and determine the best anchor point for each object. The following sections will provide a detailed explanation of these two steps.

P-IoU Calculation

The bounding box R is considered the ground truth, and the anchor point Z is located on the feature map. i_R , j_R , k_R , and l_R represent the distance from a point R to the top, bottom, left, and right of the ground truth bounding box R , respectively. In the artificial setup, the point Z is treated as the center, and a hypothetical bounding box S is created around it with the same shape S as the truth box. i_S , j_S , k_S , and l_S represent the distance

from a point R to the left and right of the hypothetical bounding box S . First, the $i_R, j_R, k_R, l_R, i_S, j_S, k_S, l_S, Q_R$, and Q_S are calculated as follows:

$$\begin{aligned} i_S &= j_S = (i_R + j_R)/2 \\ k_S &= l_S = (k_R + l_R)/2 \\ Q_R &= (i_R + j_R) * (k_R + l_R) \\ Q_S &= (i_S + j_S) * (k_S + l_S) \end{aligned} \quad (18)$$

Next, the quadrangular coordinates i^*, j^*, k^*, l^* of the intersection box are determined using the minimum operation of the respective distances. The pseudo intersection over union (P-IoU) is then calculated between the pseudo bounding box and the real boxes as follows:

$$P - IoU = \frac{|(k^* + l^*) * (i^* + j^*)|}{|Q_R + Q_S - (k^* + l^*) * (i^* + j^*)|} \quad (19)$$

Optimal Anchor Point Selection

In the process described, a bag of centers is constructed from each anchor point R_x that falls within the boundaries of the ground truth box Z . The P-IoU of the pseudo bounding box S_x -formed R_x in the bag-of-centers is then computed, and the P-IoU values are subsequently sorted in descending order. Following this, the anchor point with the highest confidence level is chosen as a positive sample, while the remaining locations are designated as negative samples. It is important to note that the loss term at the angle level has been adjusted to reduce the impact of negative angles on the overall iteration process.

The tilt angles of the ground truth center, denoted by R and H , are computed and incorporated into the loss function. The loss function for the angle value is mathematically expressed as follows:

$$P_{ang} = d(\arg \cos(\theta)\omega_1 + \arg \sin(\theta)\omega_2) \quad (20)$$

Here, d denotes as the distance between two points (R and H), and θ denotes the angle between the horizontal direction. Additionally, ω_1 and ω_2 denote the super parameters. The optimized loss function can be formulated as follows:

$$P'(\theta) = P(\theta) + P_{ang} \quad (21)$$

The label assignment approach is utilized for optimal anchor point selection for every object using this process.

3.4.4. Center Correction Mechanism

To address potential difficulties with the proposed label assignment method, a center correction mechanism is used to reduce the influence of extreme outliers. This approach makes use of center attention [39] to refine the identified anchor point using a specific parameter. The attention parameter is calculated by normalizing the distance between the pseudo bounding box's anchor and the four sides of the ground truth bounding box.

The center attention changes the center position based on the distance between the centers. Pixels in the center are allocated an initial value, which is then changed based on the overlap of the pseudo and ground truth bounding boxes.

In order to update and address this parameter, this study was assessed using a pixel-based sampling method. Specifically, the center point radius is set to 1, and the virtual box center should fall within a 3×3 pixel neighborhood. Then, the HeatMap Loss function is used to compute the deviation of the center of the pseudo bounding box, where $E_{a,b,c}$ indicates the distance weight (a, b) from the positive target center location. The closer this $E_{a,b,c}$ value is to one, the stronger the corrective effect. $\hat{E}_{a,b,c}$ indicates a higher likelihood that (a, b) accurately predicts the target's center. Additionally, C reflects the identified target

category. Assuming $C = 1$, the HeatMap Loss function can be mathematically represented as follows:

$$P_D = \frac{-1}{R} \sum_{abc} \begin{cases} \log(t_{abc})^{(1-t_{anc})^v} & \text{if } S_{abc} = 1 \\ \log(1 - t_{abc})^{(1-S_{abc})^w} (t_{abc})^v & \text{Otherwise} \end{cases} \quad (22)$$

After collecting the parameters of the 9-square grid, the pseudo bounding box i, j, k, l is adjusted using these parameters to move the center of the actual bounding box. Finally, the proposed A-frC²AcbdN model is successful in detecting and counting cotton bolls. The efficiency of the A-frC²AcbdN technique is evaluated based on various performance metrics, which include accuracy, precision, F1-score, recall, specificity, MAE, and RMSE.

4. Results and Discussion

The performance of the proposed technique was analyzed using various related techniques to determine its efficiency based on several performance metrics. The system configuration and hyperparameter details of the proposed technique are outlined in Tables 2 and 3, respectively.

Table 2. System configuration.

System	Configuration
Processor	Intel (R) Core (TM) i5-9500 COU @ 3.00 GHz.
Installed RAM	16.0 GB (15.8 GB usable)
System type	64-bit operating system, ×64-based processor
Pen and touch	No pen or touch input is available for this display

Table 3. Hyperparameter details of the proposed technique.

Process	Activation Function
Center_prediction	Sigmoid
Box_prediction	Linear
Class	SoftMax
Epoch	300

4.1. Performance Analysis

The efficiency of the proposed technique was analyzed with various related techniques to determine its performance. The comparison of accuracy and precision is presented in Figure 6.

The accuracy of the proposed technique is 94%, which is higher compared with other related techniques such as Mobilenet_v2 (91%), Attention_ResNet50 (89%), deep convolutional neural network (DCNN) (86%), and convolutional neural network (CNN) (84%). Similarly, the precision of the proposed technique is 93.8%, which is higher than that of the other related techniques, highlighting its efficient performance. The existing models have less performance value because they have fewer feature learning capabilities, a high computation time, and erroneous detection and counting. While the A-frC²AcbdN model for the compact weight backbone network enables the model to extract more features from high learning rates, the small object enhancement model detects cotton bolls more accurately. Finally, the proposed model overcomes the existing models' issues, which results in more accurate and precise cotton boll counting and detection. The comparison of recall, F1-score, and specificity are presented in Figure 7.

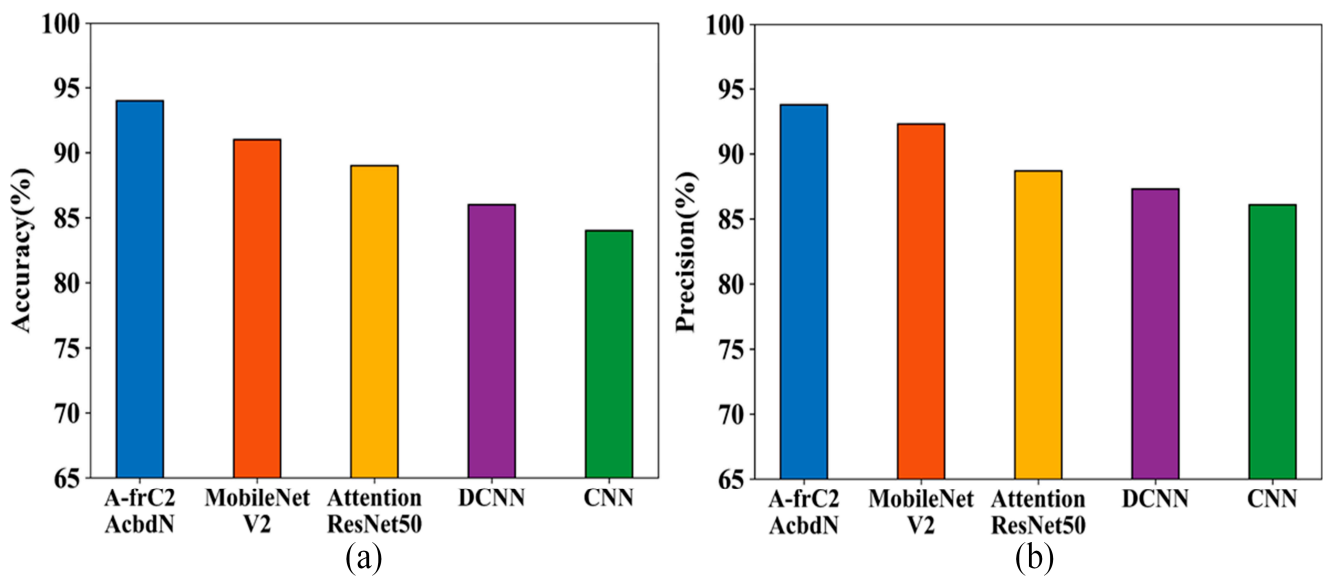


Figure 6. Comparison of the (a) accuracy and (b) precision of the proposed model (A-frC²AcbdN) with the existing models (Mobilenet_v2, Attention_ResNet50, deep convolutional neural network (DCNN), convolutional neural network (CNN)).

The recall of the proposed technique is 92.99%, which is higher than the other related techniques of Mobilenet_v2 (88.9%), Attention_ResNet50 (86.9%), DCNN (83.9%), and CNN (81.9%). Similarly, the F1-score and specificity of the proposed technique are 93.4% and 92.99%, respectively, which are higher than the other existing models, again highlighting its efficient performance. The proposed algorithm uses Mul-TH³AO to select the optimal threshold vector to segment cotton bolls. This is achieved through the A-frC²AcbdN model, resulting in a high performance in cotton boll detection and counting. The comparison of error metrics such as MAE and RMSE are presented in Figure 8.

In Figure 8, the proposed model can be analyzed through two error metrics, MAE and RMSE, with values of 0.061 and 0.244, respectively. The results demonstrate that the proposed model surpasses the existing cotton boll detection and counting strategies in performance, as the existing method achieves superior error values. The proposed model removes noise from the images via an AGWF. Pre-processing the image allows the model to focus on learning relevant features rather than being distracted by irrelevant information. The center correction system adjusts the anchor point using the pseudo bounding box and the ground truth bounding box, reducing overall error and enhancing accuracy. Figure 9 displays the time complexity of the proposed and existing models.

The proposed A-frC²AcbdN model significantly decreases data processing time by combining three key features. The AGWF will eliminate the noise in images, simplify the data, and improve processing speed. Additionally, the Mul-TH³AO algorithm is employed to select the optimal threshold for data segmentation, ensuring accurate labeling and fast execution. Finally, the model includes a clever technique to check labels using P-IoU calculations, ensuring that labeling is both fast and precise. As a result, the A-frC²AcbdN model reduces training time complexity by 35 min and testing time complexity by 15 s when compared to previous models, resulting in significant performance gains while minimizing processing time, making it a more efficient and effective solution. The accuracy and loss analysis of both the proposed and related techniques are presented in Figure 10.

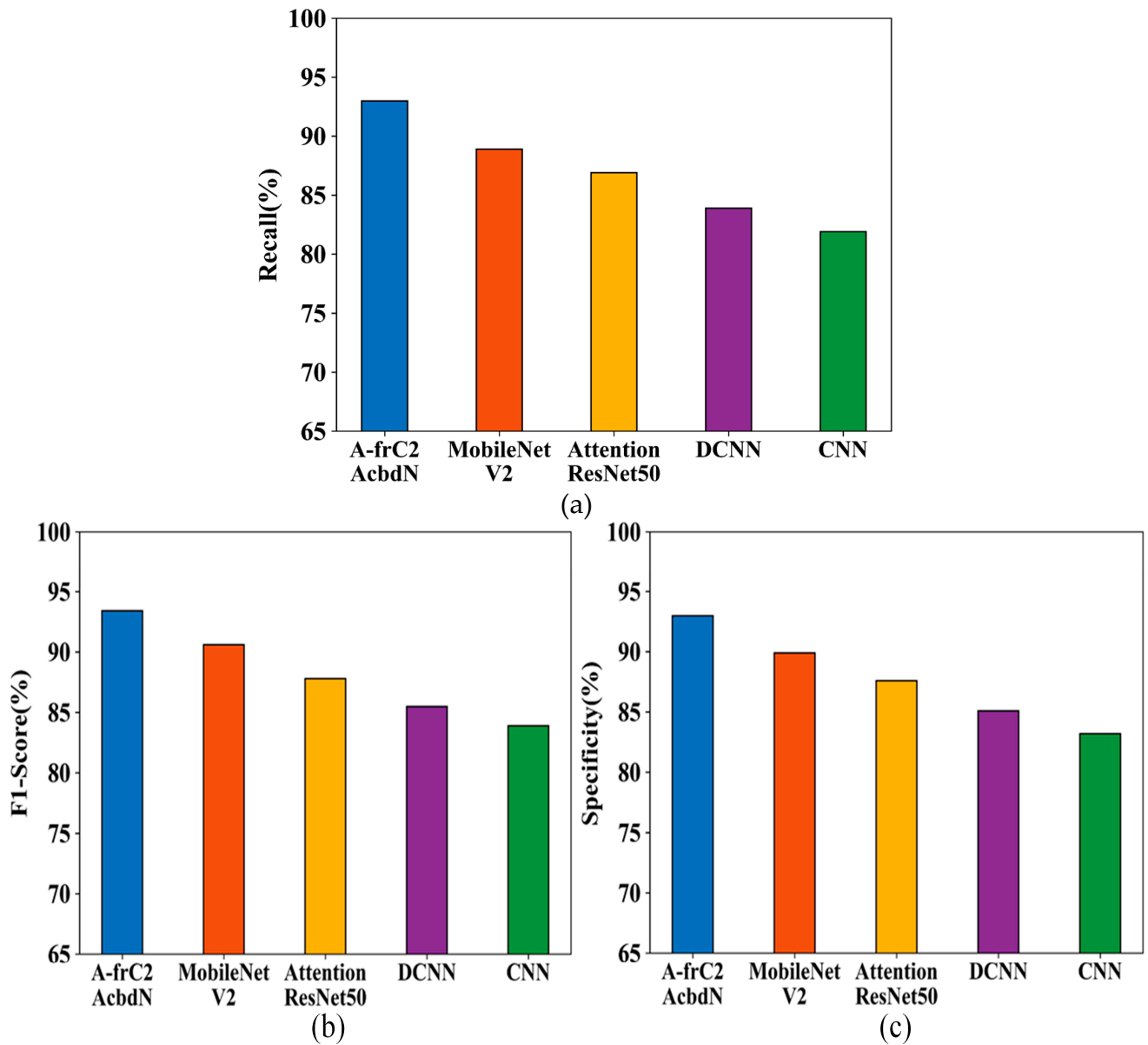


Figure 7. Comparison of the (a) recall (b) F1-score, and (c) specificity of the proposed model (AfrC²AcbdN) with the existing models (Mobilenet_v2, Attention_ResNet50, deep convolutional neural network (DCNN), and convolutional neural network (CNN)).

The accuracy and loss of both training and testing are analyzed to determine the efficient performance of the proposed technique compared to other related techniques. Usually, the data are divided based on an 80:20 ratio for training and testing. Thus here, 80% of the data are utilized for training the model, and 20% are used for testing. The accuracy of both training and testing is slightly improved compared to the related techniques. The loss of both training and testing slightly reduced its error compared to the other existing models. The overall performance of the proposed and related techniques is described in Table 4.

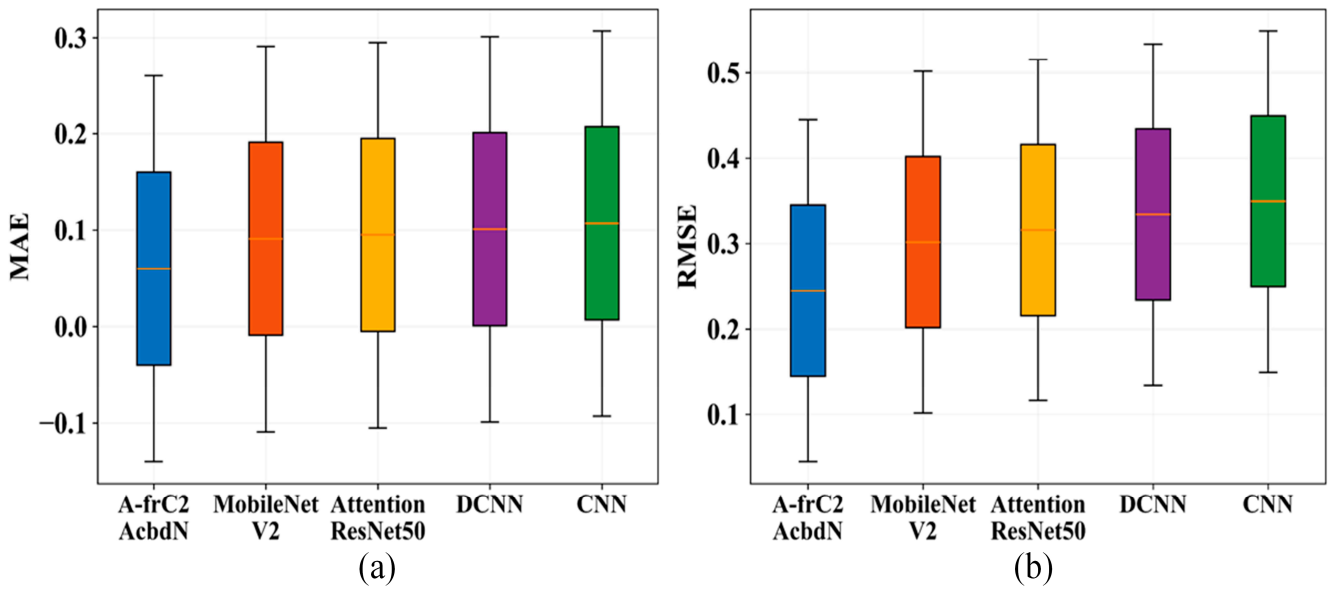


Figure 8. Comparison of the (a) MAE and (b) RMSE of the proposed model (AfrC²AcbdN) with the existing models (Mobilenet_v2, Attention_ResNet50, deep convolutional neural network (DCNN), and convolutional neural network (CNN)).

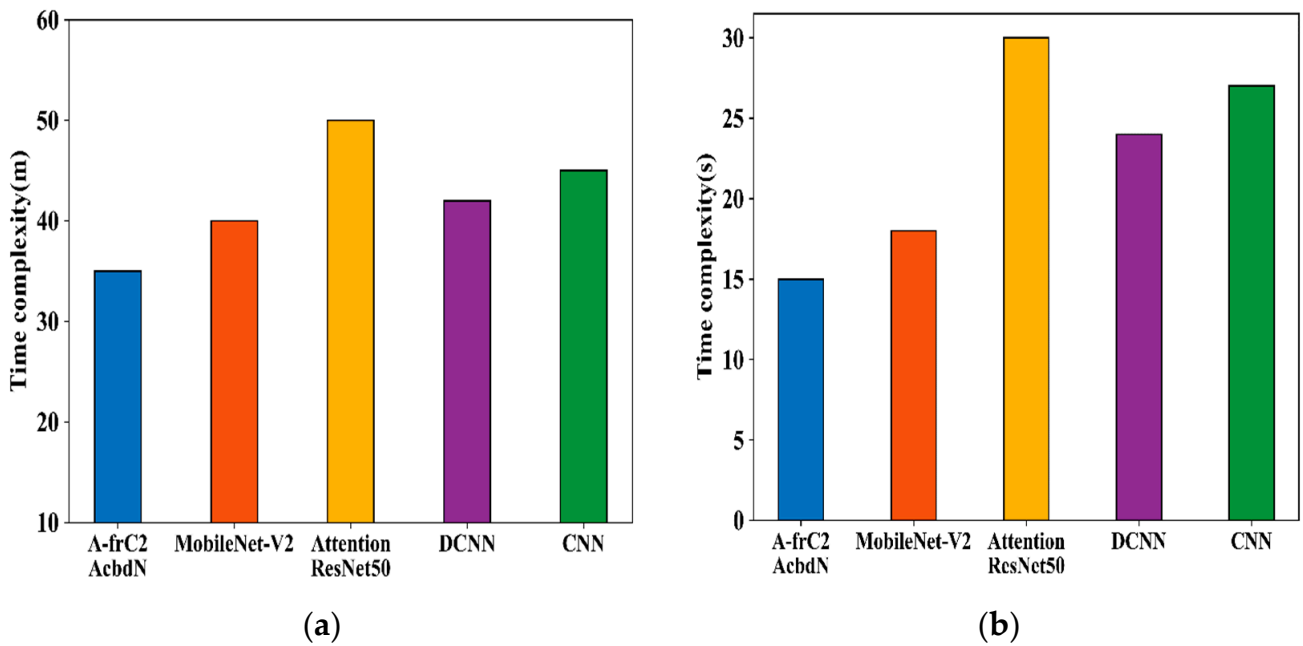


Figure 9. Analysis of the time complexity of the (a) training and (b) testing of the proposed model (AfrC²AcbdN) compared with the existing models (Mobilenet_v2, Attention_ResNet50, deep convolutional neural network (DCNN), and convolutional neural network (CNN)).

The proposed technique achieves a 94.97% accuracy, 93.82% precision, 92.99% recall, 93.48% F1-score, 92.99% specificity, 0.06 MAE, and 0.24 RMSE. The output of the proposed technique for each stage is described in Table 5.

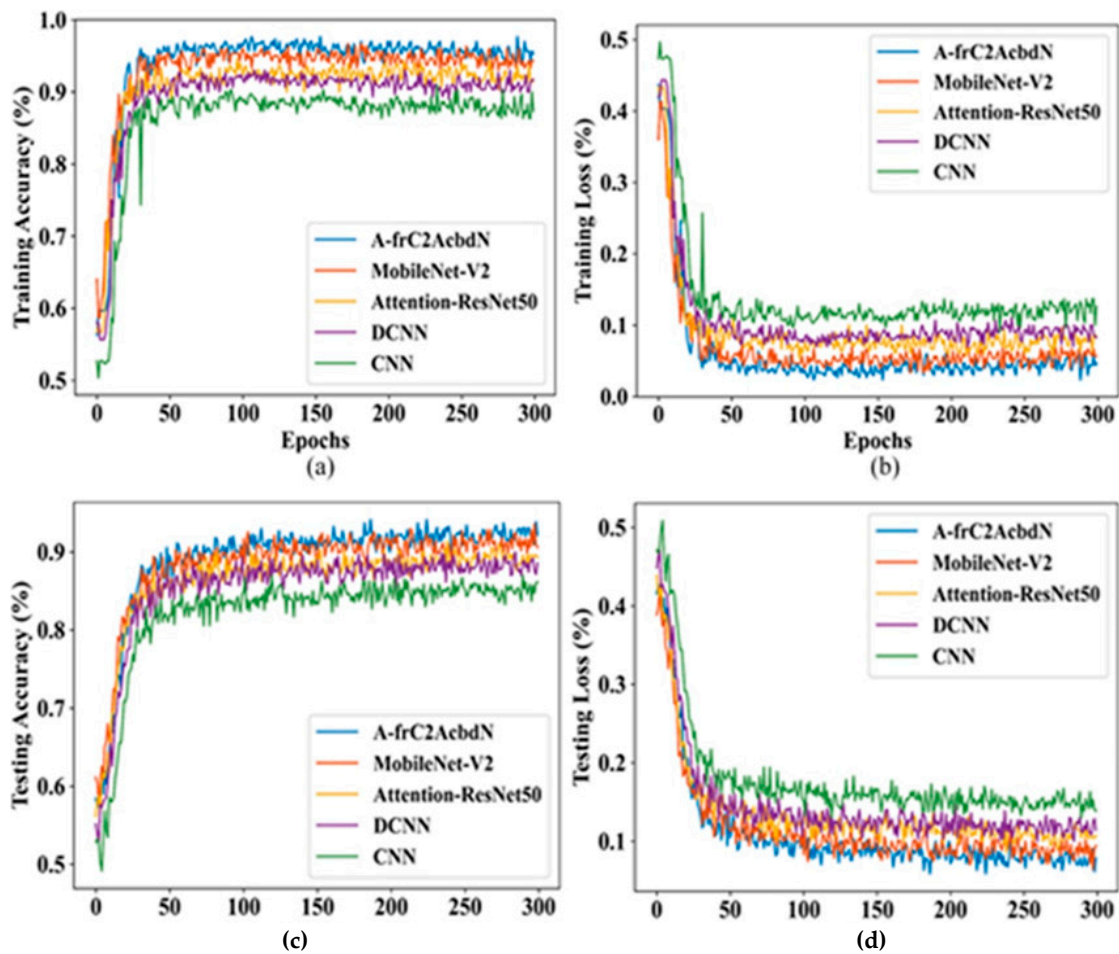



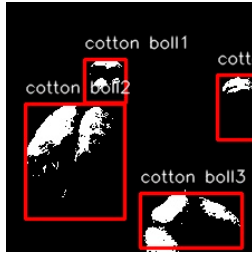



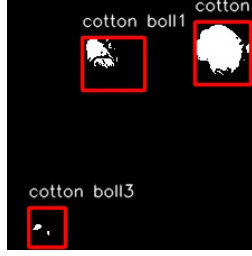



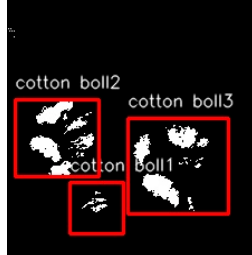
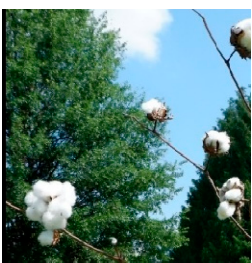


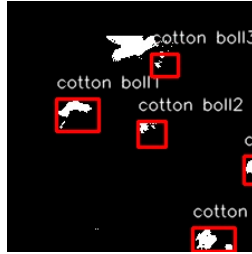
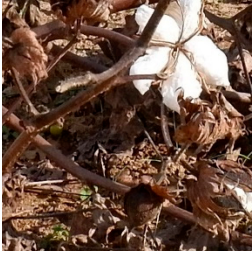
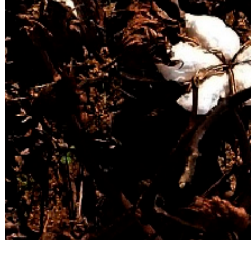

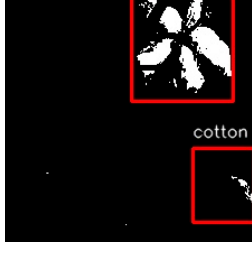


Figure 10. Performance analysis of the (a) training accuracy, (b) training loss, (c) testing accuracy, and (d) testing loss of the proposed model (AfrC²AcbdN) compared with the existing models (Mobilenet_v2, Attention_ResNet50, deep convolutional neural network (DCNN), and convolutional neural network (CNN)).

Table 4. Performance analysis of both the proposed and related techniques.

Performance	Techniques Used				
	CNN	DCNN	Attention_ResNet50	Mobilenet_v2	Proposed
Accuracy (%)	84.23	86.00	89.00	91.00	94.97
Precision (%)	86.10	87.32	88.75	92.31	93.82
Recall (%)	81.91	83.94	86.98	88.91	92.99
F1-score (%)	83.96	85.52	87.85	90.63	93.48
Specificity (%)	83.24	85.11	87.69	89.98	92.99
MAE	0.107	0.101	0.095	0.091	0.06
RMSE	0.349	0.334	0.315	0.301	0.24
Train Time complexity (m)	45	42	50	40	35
Test Time complexity (s)	27	24	30	18	15

Table 5. The outcome of the proposed technique.

Original Image	Pre-Processed Image (AGWF)	Segmented Image (IH ² AOA)	Cotton-Boll-Detected Image (A-frC ² AcbdN)	Counting of Cotton Bolls
				4
				3
				3
				5
				2

4.2. Ablation Study

The proposed research work includes the stages of pre-processing, segmentation, and cotton boll detection and counting. The ablation study analyzes the performance deviation resulting from the elimination of the techniques used. Table 6 indicates the performance

outcomes of conducting an ablation study on the annotated dataset for weakly supervised cotton boll detection and counting.

Table 6. Ablation study analysis.

Module	Technique Used	Performance (%)		
		Accuracy	Precision	Recall
Module 1	Without pre-processing + without segmentation + with detection and counting model	80	79	78
Module 2	Without segmentation + with pre-processing + with detection and counting model	86	85	85
Module 3	Without pre-processing + with segmentation + with detection and counting model	90	89	89
Module 4	With pre-processing + with segmentation + with detection and counting model	94	93	93

Module 4 is the proposed model, which incorporates both pre-processing and segmentation, leading to better performance. In comparison, the other modules demonstrate significantly lower accuracy: Module 1 utilizes the A-frC²AcbdN model without any pre-processing or segmentation, resulting in poor performance; Module 2 includes pre-processing but omits segmentation, resulting in a similar low accuracy; and Module 3 includes segmentation but without pre-processing, resulting in decreased performance. The analysis demonstrated in Module 4 with the combination of pre-processing and segmentation is essential for optimal accuracy, as the absence of either component, as seen in Modules 1, 2, and 3, results in a significant decrease in the model's efficacy.

4.3. Discussion

This section presents a comparative analysis of the proposed technique against several related methods, summarized in Table 7.

Table 7. Comparative analysis.

Author Name and Reference	Technique Used	Performance
Yu Jiang et al. [26]	DeepFlower	RMSE—0.79
Naseb Singh et al. [27]	YCbCr method, band ratio, and chromatic algorithm	Identification rate—91.05%
Mengli Zhang et al. [28]	YOLO SSPD	12.38 RMSE
RuiXu et al. [29]	CNN	90% Precision
Proposed	A-frC ² AcbdN	94.97% Accuracy, Precision 93.82%, RMSE 0.24%

The proposed A-frC²AcbdN model achieves an accuracy of 94%, surpassing the existing models, which suffer from limitations such as a high time consumption, high model complexity, requirement for larger datasets, overfitting problems, higher loss rate, insufficient features considered, and inability to separate overlapped cotton bolls. The proposed technique's superior performance can be attributed to several key factors, including the effective use of an annotated dataset from weakly supervised cotton boll detection and counting, which provides high-quality images of cotton bolls, as well as the noise elimination process that employs the AGWF algorithm, which reduces error and improves overall accuracy. Furthermore, the segmentation algorithm combines AO and Harris Hawks optimization to accurately identify and separate cotton bolls from the background, reducing

computational complexity. At the same time, the compact weight backbone network allows the model to extract more features from high learning rates, improving accuracy. Furthermore, the small object enhancement module and label assignment strategy allow the model to detect and count cotton bolls with high precision, reducing loss through the P-IoU calculation. At the same time, the center correction mechanism updates the anchor point based on the pseudo bounding box and ground truth bounding box, reducing overall error and improving accuracy. Overall, the proposed technique performs well in cotton boll detection and counting, making it a potential solution for the agricultural industry. It allows farmers and researchers to better understand and manage cotton crops.

5. Conclusions

An efficient cotton boll detection and counting method is developed to achieve an efficient performance and overcome the complexity of time. Initially, pre-processing is performed using the data extracted from the dataset to increase the quality of the images. An AGWF is used to eliminate noise from the acquired pictures, whereas IH²AOA is employed for segmentation. Finally, cotton boll detection and counting were performed using the A-frC²AcbdN model. From weakly supervised cotton boll detection and counting, the proposed approach made use of an annotated dataset. The proposed method yielded an accuracy of 94%, outperforming that of the other related techniques. Additionally, the results for the precision, recall, F1-score, and specificity of the suggested technique were 93.82%, 92.99%, 93.48%, and 92.99%, respectively. The MAE and RMSE of the proposed technique were 0.06 and 0.24, respectively, which are lower than the other related techniques. In the future, the performance of the proposed technique can be improved by utilizing an advanced YOLO model for more efficient cotton boll counting.

Author Contributions: Resources, U.N.D. and A.B.; Writing—original draft, A.B.; Writing—review & editing, U.N.D. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

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

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

References

1. Deepa, S.; Alli, A.; Sheetac; Gokila, S. Machine learning regression model for material synthesis prices prediction in agriculture. *Mater. Today Proc.* **2021**, *81*, 989–993. [\[CrossRef\]](#)
2. Chakraborty, S.; Prasad, K. A quality function deployment-based expert system for cotton fibre selection. *J. Inst. Eng. (India) Ser. E* **2018**, *99*, 43–53. [\[CrossRef\]](#)
3. Gharakhani; Thomasson, A. Robotic cotton harvesting and field fiber seed separation approaches and challenges. In Proceedings of the Beltwide Cotton Conferences, Austin, TX, USA, 8–10 January 2020; pp. 768–773.
4. Fue, K.G.; Porter, W.M.; Rains, G.C. Deep learning based real-time GPU-accelerated tracking and counting of cotton bolls under field conditions using a moving camera. In Proceedings of the 2018 ASABE Annual International Meeting, Detroit, MI, USA, 29 July–1 August 2018; p. 1.
5. Bawa, A.; Samanta, S.; Himanshu, S.K.; Singh, J.; Kim, J.; Zhang, T.; Chang, A.; Jung, J.; DeLaune, P.; Bordovsky, J.; et al. A support vector machine and image processing based approach for counting open cotton bolls and estimating lint yield from UAV imagery. *Smart Agric. Technol.* **2022**, *3*, 100140. [\[CrossRef\]](#)
6. Dube, N.; Bryant, B.; Sari-Sarraf, H.; Ritchie, G.L. Cotton boll distribution and yield estimation using three-dimensional point cloud data. *Agron. J.* **2020**, *112*, 4976–4989. [\[CrossRef\]](#)
7. Shi, G.; Du, X.; Du, M.; Li, Q.; Tian, X.; Ren, Y.; Zhang, Y.; Wang, H. Cotton yield estimation using the remotely sensed cotton boll index from UAV images. *Drones* **2022**, *6*, 254. [\[CrossRef\]](#)
8. Tedesco-Oliveira, D.; da Silva, R.P.; Maldonado, W., Jr.; Zerbato, C. Convolutional neural networks in predicting cotton yield from images of commercial fields. *Comput. Electron. Agric.* **2020**, *171*, 105307. [\[CrossRef\]](#)
9. Fue, K.G.; Porter, W.M.; Barnes, E.M.; Rains, G.C. Ensemble method of deep learning, color segmentation, and image transformation to track, localize, and count cotton bolls using a moving camera in real-time. *Trans. ASABE* **2021**, *64*, 341–352. [\[CrossRef\]](#)

10. Liang, Z.; Cui, G.; Xiong, M.; Li, X.; Jin, X.; Lin, T. YOLO-C: An Efficient and Robust Detection Algorithm for Mature Long Staple Cotton Targets with High-Resolution RGB Images. *Agronomy* **2023**, *13*, 1988. [[CrossRef](#)]
11. Wang, Y.; Xiao, C.; Wang, Y.; Li, K.; Yu, K.; Geng, J.; Li, Q.; Yang, J.; Zhang, J.; Zhang, M.; et al. Monitoring of Cotton Boll Opening Rate Based on UAV Multispectral Data. *Remote Sens.* **2023**, *16*, 132. [[CrossRef](#)]
12. Xu, W.; Yang, W.; Chen, S.; Wu, C.; Chen, P.; Lan, Y. Establishing a model to predict the single boll weight of cotton in northern Xinjiang by using high resolution UAV remote sensing data. *Comput. Electron. Agric.* **2020**, *179*, 105762. [[CrossRef](#)]
13. Siegfried, J.; Adams, C.B.; Rajan, N.; Hague, S.; Schnell, R.; Hardin, R. Combining a cotton 'Boll Area Index' with in-season unmanned aerial multispectral and thermal imagery for yield estimation. *Field Crop. Res.* **2022**, *291*, 108765. [[CrossRef](#)]
14. Kumar, S.; Jain, A.; Shukla, A.P.; Singh, S.; Raja, R.; Rani, S.; Harshitha, G.; AlZain, M.A.; Masud, M. A comparative analysis of machine learning algorithms for detection of organic and nonorganic cotton diseases. *Math. Probl. Eng.* **2021**, *2021*, 1–18. [[CrossRef](#)]
15. Yeom, J.; Jung, J.; Chang, A.; Maeda, M.; Landivar, J. Automated open cotton boll detection for yield estimation using unmanned aircraft vehicle (UAV) data. *Remote Sens.* **2018**, *10*, 1895. [[CrossRef](#)]
16. Zhang, C.; Feng, X.; Li, L.; Song, Y. Identification of cotton contaminants using neighborhood gradient based on YCbCr color space. In Proceedings of the 2010 2nd International Conference on Signal Processing Systems (ICSPS), Dalian, China, 5–7 July 2010; pp. V3-733–V3-738.
17. Liu, Q.; Zhang, Y.; Yang, G. Small unopened cotton boll counting by detection with MRF-YOLO in the wild. *Comput. Electron. Agric.* **2022**, *204*, 107576. [[CrossRef](#)]
18. Li, Y.; Cao, Z.; Lu, H.; Xiao, Y.; Zhu, Y.; Cremers, A.B. In-field cotton detection via region-based semantic image segmentation. *Comput. Electron. Agric.* **2016**, *127*, 475–486. [[CrossRef](#)]
19. Tan, C.; Sun, J.; Paterson, A.H.; Song, H.; Li, C. Three-view cotton flower counting through multi-object tracking and RGB-D imagery. *Biosyst. Eng.* **2024**, *246*, 233–247. [[CrossRef](#)]
20. Latif, M.R.; Khan, M.A.; Javed, M.Y.; Masood, H.; Tariq, U.; Nam, Y.; Kadry, S. Cotton Leaf Diseases Recognition Using Deep Learning and Genetic Algorithm. *Comput. Mater. Contin.* **2021**, *69*, 2917–2932. [[CrossRef](#)]
21. Ruan, A.; Xu, M.; Ban, S.; Wei, S.; Tian, M.; Yang, H.; Hu, A.; Hu, D.; Li, L. LettuceNet: A Novel Deep Learning Approach for Efficient Lettuce Localization and Counting. *Agriculture* **2024**, *14*, 1412. [[CrossRef](#)]
22. Zhao, J.; Pan, F.; Li, Z.; Lan, Y.; Lu, L.; Yang, D.; Wen, Y. Detection of cotton waterlogging stress based on hyperspectral images and convolutional neural network. *Int. J. Agric. Biol. Eng.* **2021**, *14*, 167–174. [[CrossRef](#)]
23. Adke, S.; Li, C.; Rasheed, K.M.; Maier, F.W. Supervised and weakly supervised deep learning for segmentation and counting of cotton bolls using proximal imagery. *Sensors* **2022**, *22*, 3688. [[CrossRef](#)]
24. Singh, N.; Tewari, V.; Biswas, P.; Dhruw, L.; Pareek, C.; Singh, H.D. Semantic segmentation of in-field cotton bolls from the sky using deep convolutional neural networks. *Smart Agric. Technol.* **2022**, *2*, 100045. [[CrossRef](#)]
25. Priya, D. Cotton leaf disease detection using Faster R-CNN with Region Proposal Network. *Int. J. Biol. Biomed.* **2021**, *6*, 23.
26. Jiang, Y.; Li, C.; Xu, R.; Sun, S.; Robertson, J.S.; Paterson, A.H. DeepFlower: A deep learning-based approach to characterize flowering patterns of cotton plants in the field. *Plant Methods* **2020**, *16*, 1–17. [[CrossRef](#)] [[PubMed](#)]
27. Singh, N.; Tewari, V.; Biswas, P.; Pareek, C.; Dhruw, L. Image processing algorithms for in-field cotton boll detection in natural lighting conditions. *Artif. Intell. Agric.* **2021**, *5*, 142–156. [[CrossRef](#)]
28. Zhang, M.; Chen, W.; Gao, P.; Li, Y.; Tan, F.; Zhang, Y.; Ruan, S.; Xing, P.; Guo, L. YOLO SSPD: A small target cotton boll detection model during the boll-spitting period based on space-to-depth convolution. *Front. Plant Sci.* **2024**, *15*, 1409194. [[CrossRef](#)]
29. Xu, R.; Li, C.; Paterson, A.H.; Jiang, Y.; Sun, S.; Robertson, J.S. Aerial images and convolutional neural network for cotton bloom detection. *Front. Plant Sci.* **2018**, *8*, 2235. [[CrossRef](#)]
30. Kurban, R. Gaussian of differences: A simple and efficient general image fusion method. *Entropy* **2023**, *25*, 1215. [[CrossRef](#)]
31. Salehi, H.; Vahidi, J.; Abdeljawad, T.; Khan, A.; Rad, S.Y.B. A SAR image despeckling method based on an extended adaptive wiener filter and extended guided filter. *Remote Sens.* **2020**, *12*, 2371. [[CrossRef](#)]
32. Costache, R.; Pal, S.C.; Pande, C.B.; Islam, A.R.M.T.; Alshehri, F.; Abdo, H.G. Flood mapping based on novel ensemble modeling involving the deep learning, Harris Hawk optimization algorithm and stacking based machine learning. *Appl. Water Sci.* **2024**, *14*, 78. [[CrossRef](#)]
33. Abualigah, L.; Diabat, A.; Sumari, P.; Gandomi, A.H. A novel evolutionary arithmetic optimization algorithm for multi-level thresholding segmentation of COVID-19 CT images. *Processes* **2021**, *9*, 1155. [[CrossRef](#)]
34. Katherine; Rulaningtyas, R.; Ain, K. CT scan image segmentation based on hounsfield unit values using Otsu thresholding method. *J. Phys. Conf. Ser.* **2021**, *1816*, 012080. [[CrossRef](#)]
35. Zhao, L.; Wang, L. A new lightweight network based on MobileNetV3. *KSII Trans. Internet Inf. Syst. (TIIS)* **2022**, *16*, 1–15.
36. Bharati, S.; Podder, P.; Mondal, M.R.H.; Prasath, V.S. CO-ResNet: Optimized ResNet model for COVID-19 diagnosis from X-ray images. *Int. J. Hybrid Intell. Syst.* **2021**, *17*, 71–85. [[CrossRef](#)]
37. Aggarwal, Y.; Guha, P. FDLite: A Single Stage Lightweight Face Detector Network. *arXiv* **2024**, arXiv:2406.19107.

38. Cheng, G.; Wang, J.; Li, K.; Xie, X.; Lang, C.; Yao, Y.; Han, J. Anchor-free oriented proposal generator for object detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [[CrossRef](#)]
39. Feng, J.; Wang, Q.; Zhang, G.; Jia, X.; Yin, J. CAT: Center Attention Transformer with Stratified Spatial–Spectral Token for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–15. [[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.