

Article

Benchmark Feature Detection Method for Mobile Robot Automatic Drilling System Integrated with Deep Learning

Jialong Dai ^{1,2} , Jianxin Shen ¹, Wei Tian ^{1,*} , Pengcheng Li ¹, He Liu ³  and Xiangshun Cui ²

¹ College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; djl2013@nuaa.edu.cn (J.D.)

² Jiangsu AERO-APEX Intelligent Technologies Co., Ltd., Nanjing 210016, China

³ AVIC Jiangxi Hongdu Aviation Industry Group Company Ltd., Nanchang 330096, China

* Correspondence: tw_nj@nuaa.edu.cn

Abstract: Benchmark feature detection is critical in mobile robot automatic drilling systems for compensating robot accuracy and assembly errors in aerospace manufacturing. System accuracy is influenced by reference feature recognition, which is often hindered by material interference and background noise. To address these issues, this paper proposes a method that uses a 2D industrial camera for image capture, applies deep learning for initial target recognition and positioning, and then determines the feature extraction location based on the initial recognition. The extracted benchmark positions are accurately fitted using an improved Huber algorithm. Experimental results demonstrate that this approach improves the benchmark feature detection recognition rate by 43.8%, center recognition accuracy by 78.26%, and overall hole processing accuracy by 54.69%.

Keywords: automatic drilling; deep learning; YOLOv5; benchmark detection; robotics; mobile robotic systems

1. Introduction

In recent years, with the rapid development of aerospace technology, the aerospace industry has become a representative of high-end manufacturing, and its production and assembly technology directly determines the performance and reliability of aircraft. In the aircraft manufacturing process, the assembly process accounts for more than 40% of the entire production cycle [1]. As one of the key processes in aircraft assembly, drilling directly determines the performance of the connector and the strength of the overall structure. In the assembly of aircraft structural parts, more than half of the components need to be completed by riveting, and the accuracy of the drilling position will directly affect the force distribution and connection quality of the rivet [2], especially in the working environment, where mobile robots are used to assist in assembly [3]. The drilling accuracy problem is more prominent.

In mobile robot automatic drilling systems, accuracy is affected by three main error sources: (1) robot-induced errors [4], such as rigidity deviations and cumulative positioning errors; (2) dynamic operational errors, including vibrations and path deviations [5]; and (3) visual system errors [6], such as calibration inaccuracies and lighting changes. These errors compound, reducing system precision.

Accurate benchmark feature recognition is crucial for error compensation. Any recognition error propagates through the system, impacting compensation [7]. Precise detection ensures consistent global positioning, even with movement-induced errors [8]. It enables



Academic Editor: Dan Zhang

Received: 4 January 2025

Revised: 1 February 2025

Accepted: 7 February 2025

Published: 17 February 2025

Citation: Dai, J.; Shen, J.; Tian, W.; Li, P.; Liu, H.; Cui, X. Benchmark Feature Detection Method for Mobile Robot Automatic Drilling System Integrated with Deep Learning. *Machines* **2025**, *13*, 154. <https://doi.org/10.3390/machines13020154>

Copyright: © 2025 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/>).

real-time local error compensation and stabilizes the system against environmental fluctuations [9], thus improving overall drilling accuracy.

At present, in robotic automated drilling and assembly systems, benchmark feature extraction can be achieved through three main methods: measuring and obtaining reference points using laser tracker devices, analyzing and identifying benchmark features from scanned point cloud data, and capturing 2D images of benchmark features using a 2D industrial area scan camera for feature recognition. Although several methods have been proposed for benchmark feature detection in robotic drilling systems, they face significant challenges, especially in mobile robotic applications. Laser tracker-based methods [10] can be affected by material interference, while binocular vision [11] and neuromorphic vision [12] suffer from environmental sensitivity, reducing accuracy. Traditional techniques, such as template matching and 3D point cloud methods [13–19], also struggle with poor scan quality and high computational demands, particularly in non-ideal conditions.

In response to these challenges, this paper proposes a novel method that integrates deep learning with projection feature extraction. By using a 2D camera for benchmark feature calculation and analysis, the method overcomes data loss, high computational demands, and hardware limitations associated with point cloud data collection through line lasers. The main contributions are summarized as follows:

- (1) Leveraging deep learning for preliminary positioning. It is highly adaptive, robust to illumination, noise, and target deformation, and can handle a wider range of benchmark holes with better accuracy.
- (2) It uses surrounding benchmark feature sampling and projection feature extraction, combined with an improved Huber algorithm for accurate fitting.
- (3) This method addresses common recognition challenges such as dirt, surface wear, and misaligned prefabricated benchmarks.

By calibrating the relationship between the local benchmark area and the robot's base coordinate system, it not only resolves positioning and path planning issues but also significantly enhances system stability and adaptability in dynamic environments, offering an innovative and reliable solution for improving overall accuracy in mobile robot automatic drilling.

The paper is organized as follows. Section 2 provides a literature review, summarizing the existing research in the field and identifying research gaps. Section 3 introduces the hardware composition of the mobile robot drilling system and the compensation methods for benchmark detection. In Section 4, the methodology for benchmark feature detection is introduced, including the YOLOv5-based approach and the feature extraction process. Section 5 presents the experimental setup and results. Section 6 concludes the paper and discusses potential future work.

2. Literature Review

Wang Qing et al. [10] proposed a hole position correction method using hole margin constraint and Shepard interpolation. By using a laser tracker device to obtain measured skeleton point cloud data and perform denoising and registration, the key benchmark features of the skeleton boundary points are extracted. This method involves the use of a laser tracker device and is relatively complex to operate. Sun Xinyue et al. [11] used a binocular vision inspection system to scan the test workpiece, obtained the benchmark data of the test workpiece with the help of a laser tracker, and used SA (Spatial Analyzer; SA2020.04.09_65432) software to process and fit the point cloud data to obtain the actual benchmark feature coordinate information. The binocular vision system is relatively large and requires the help of external software for calculation and analysis.

Abdulla Ayyad et al. [12] proposed a control method based on neuromorphic vision, combining visual sensors to detect and locate the benchmark features of through-hole types in real time. Although this method shows high accuracy in some cases, its limitations are also obvious. The neuromorphic vision system has weak adaptability to changes in lighting and environment and is easily affected by external interference, resulting in a decrease in the recognition accuracy of benchmark features. Andreas Frommknecht et al. [13] proposed a multi-sensor measurement system to improve the benchmark detection accuracy in robot drilling. The system integrates laser sensors. Laser sensors are affected by surface reflection and material, resulting in inaccurate data. Zhuang Zhiwei et al. [14] proposed a hole position and normal vector detection algorithm based on template matching. They used the principle of collecting three-dimensional point clouds of the measured surface using a line laser profiler, filtered the point cloud data using a filter, converted it into a two-dimensional plane point cloud, and used template matching to identify and locate the reference hole position. This method requires the use of line laser scanning point cloud data, which is easily disturbed by the reflection of the workpiece surface during scanning, resulting in poor quality of the scanned point cloud data. In addition, post-processing requires processing point cloud data, which has high hardware requirements and time-consuming calculation processing. Tan Xiaoqun [15] proposed a benchmark hole detection technology based on line laser scanning and image processing, which converts the three-dimensional point cloud data measured by the line laser camera into a two-dimensional grayscale image for feature extraction and processing, and then converts it to three-dimensional space for analysis and fitting of the benchmark aperture position.

Liu Yuanwei et al. [16] proposed a method to realize the benchmark detection of the robot drilling system through a visual system. The Hough transform method is used to directly analyze the benchmark hole. This method has only been verified on standard circular through holes. This method is only suitable for the situation where the benchmark features are good and are not affected by the environment and surface material. When the processing environment is not good, it will be difficult to ensure the robustness of the benchmark detection. Yan Qiubai et al. [17] used a monocular vision camera to detect benchmark features online, but the benchmark type identified was a through-hole benchmark with good drilling quality. The type was relatively simple and the benchmark feature identification was relatively easy. Zhang Haowei et al. [18] used a 2D industrial area scan camera to identify benchmark features based on a hybrid track drilling system. The identified feature object was a prefabricated through hole as the reference identification type, but the identification feature type was relatively single. Du Tianyu et al. [19] proposed a robust detection method for multiple types of assembly benchmarks using monocular vision. Through the selected benchmark feature arc segments, the approximate circle of the benchmark is fitted, and then the arc is clustered by the set threshold. The final benchmark contour is fitted to identify the benchmarks of through holes, target points, rivets and through-hole clips. Among them, the contrast of through holes, target points and rivets is relatively high, and the recognition difficulty is low, but target point stickers are rarely used in actual assembly. In addition, template matching is relatively easy for the above types of benchmarks, but template matching is prone to failure for benchmark features with low image quality.

These methods perform well in controlled environments, but the dynamic nature of mobile robotic drilling introduces additional challenges. Errors arising from robot mobility and the need for real-time visual resynchronization make benchmark feature detection crucial for maintaining drilling accuracy.

Benchmark feature detection in robotic drilling systems has been the focus of extensive research. The existing methods and their limitations, as well as the innovations of the present work, are shown in Table 1

Table 1. Methods and their limitations vs. innovations of the present work.

Reference	Methodology & Limitations	Innovations of Present Work
[10,11]	Using laser trackers and stereo cameras for measurement results in complex operations and relatively poor timeliness.	
[12–15]	The neuromorphic vision-based control method uses visual sensors for real-time detection of benchmark features but shows weak adaptability to lighting and environmental changes, making it susceptible to interference. Multi-sensor measurement systems integrate laser sensors to improve detection accuracy but are affected by surface reflection and material properties, leading to inaccurate data. Template matching algorithms use line laser scanning to capture point clouds and convert them to 2D for benchmark hole detection, but surface reflection affects data quality, and processing is time-consuming. Line laser scanning with image processing converts point clouds to 2D grayscale for feature extraction, then back to 3D for analysis, but suffers from reflection interference and complex, time-consuming processing.	The use of a 2D camera for image acquisition avoids the issue of missing feature edge data caused by material reflection in line laser cameras. It shortens the time, improves efficiency, and is adaptable to meet various complex reference types. A dual strategy of coarse localization and fine localization is employed. The coarse localization efficiently and stably identifies the target regions of various complex types, and then, based on the region obtained from the coarse localization, the fine localization extracts feature points around the center of the region for fitting, enabling precise identification of the reference target's center position.
[16–19]	The Hough transform method is used for benchmark hole detection, suitable for standard circular through holes, but sensitive to environmental and surface material changes, lacking robustness. Monocular vision cameras detect benchmark features, suitable for good-quality through holes, but have limited applicability. Two-dimensional industrial scan cameras identify benchmark features, mainly for prefabricated through holes, with limited feature types. The monocular vision-based method for detecting multiple benchmark types fits circles and clusters arcs, with easy recognition for through holes, target points, and rivets, but target point stickers are rarely used, and template matching fails with low-quality images.	

3. Mobile Robot Automatic Drilling System

3.1. System Composition

The mobile robot automatic drilling system is a piece of highly efficient automated equipment for the assembly of aircraft parts. Combining advanced mobile robot technology, visual inspection system and high-precision drilling spindle tools, it can achieve high-quality and efficient drilling processing on the surface of complex aircraft parts. The system is mainly composed of an electrical control system, drilling software control system, AGV mobile carrier, automatic tool changer, test drilling platform, industrial robot, end drilling actuator, vertical lifting axis, etc., as shown in Figure 1.

The mobile robot automatic drilling system is mainly used in the assembly of large and medium-sized structural parts in the aviation manufacturing industry, including drilling and riveting of aircraft panels, wings, fuselages and other parts, which require high-precision and high-consistency multi-hole group drilling operations, as well as complex areas that are difficult to process with fixed equipment during the manufacturing process. In the aircraft assembly process, the drilling accuracy directly affects the final assembly quality. Due to the manufacturing errors of the assembled parts and the actual installation errors, the

direct use of the theoretical drilling points generated by the offline programming software for processing cannot meet the accuracy requirements of aircraft assembly. Therefore, in order to meet the accuracy requirements of aircraft assembly drilling, it is necessary to introduce benchmark detection to correct and compensate for the theoretical position error of the hole to be made.

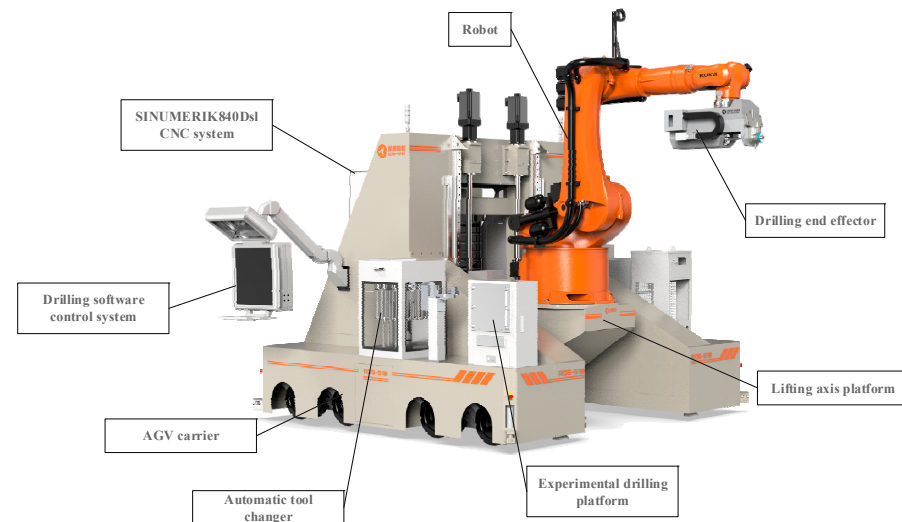


Figure 1. Mobile robot automatic drilling system.

The benchmark detection module consists of a module housing, lens protection cover, a 12-megapixel area scan industrial camera, a fixed-magnification lens, and a high-brightness white ring light. The structure is shown in Figure 2. The 12-megapixel area scan industrial camera offers a resolution of 4096 (pixels) $\times$ 3000 (pixels). Paired with a $0.4\times$ fixed-magnification lens and a shooting distance of 230 mm, it achieves a single-pixel size of 0.008 mm. The benchmark detection module is embedded within the end effector, ensuring that the camera's central axis and the spindle's central axis are installed on the same plane. When performing benchmark detection, the control system executes a station-switching command, moving the center of the reference detection module down to the zero position of the spindle. This ensures that the camera's center is coaxial with the pressure foot's center. Once the benchmark detection module is in position, the control system opens the lens protection cover and activates the light source to capture images and locate the benchmark.

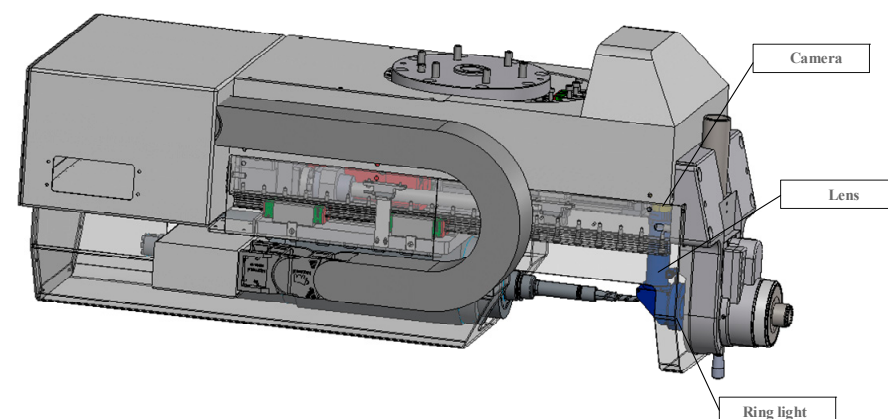


Figure 2. Benchmark detection module.

3.2. Benchmark Detection Compensation Method

Benchmark detection uses the internal visual positioning module of the end effector of the mobile robot automatic drilling system [20] to identify the image pixel coordinates of the center position of the benchmark hole, and then converts it to the position information in the robot end effector coordinate system, and finally obtains the corrected robot hole position to be processed. The benchmark detection process first needs to establish the world coordinate system (World), robot base coordinate system [21] (Robot Base), tool coordinate system [22] (Tool), benchmark detection camera coordinate system (Camera), resynchronized landmark camera coordinate system (ResynCamera), product coordinate system (Production), and fixture coordinate system (Fixture). The coordinate system of the benchmark detection process is shown in Figure 3.

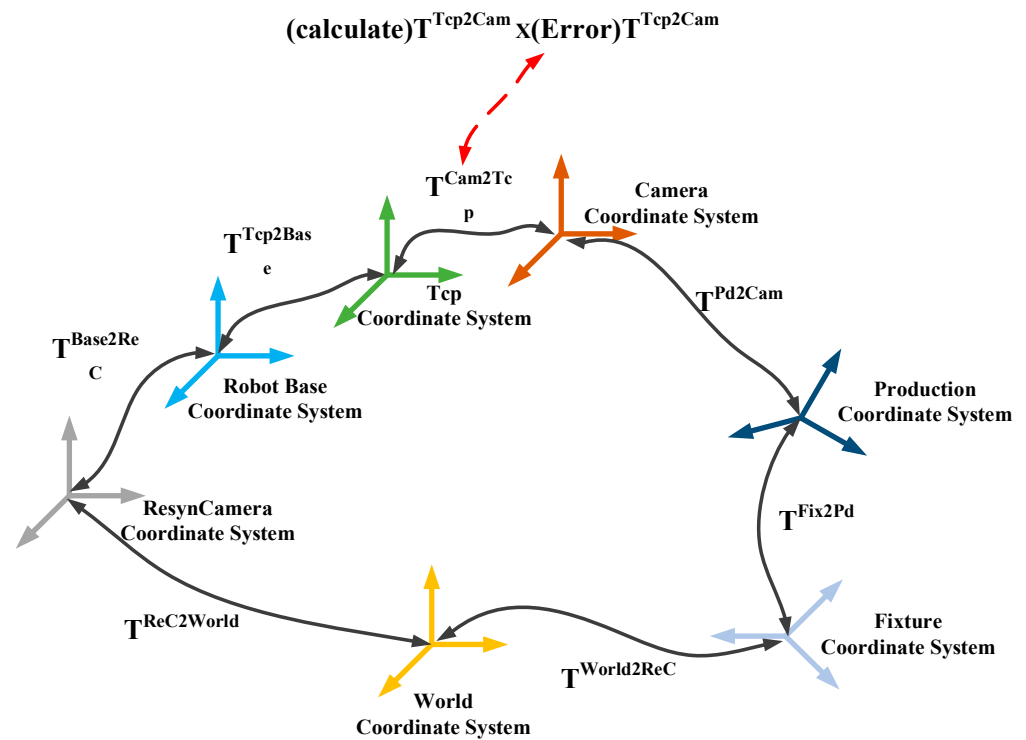


Figure 3. Coordinate system of the benchmark detection process.

According to the coordinate system diagram in Figure 2, the camera obtains the pixel coordinates of the reference hole by photographing the reference hole based on the product coordinate system $\begin{bmatrix} x_{Cal} \\ y_{Cal} \\ z_{Cal} \end{bmatrix}_{pix}$; $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{hole}$ is the reference hole coordinate based on the product coordinate system calculated by pixel coordinates and transformation matrix. Its relationship can be expressed as Equation (1).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{hole} = \begin{bmatrix} x_{Cal} \\ y_{Cal} \\ z_{Cal} \end{bmatrix}_{pix} \times [T^{Cam2Tcp}]^{-1} \times [T^{Tcp2Base}]^{-1} \times [T^{Base2ReC}]^{-1} \times [T^{ReC2W}]^{-1} \times [T^{W2Fix}]^{-1} \times [T^{Fix2Pd}]^{-1} \quad (1)$$

The coordinates of the reference hole in the robot tool center point TCP coordinate system can be calculated by Equation (2).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{tcp} = \begin{bmatrix} x_{Cal} \\ y_{Cal} \\ z_{Cal} \end{bmatrix}_{pix} \times [T^{Cam2Tcp}]^{-1} \quad (2)$$

where $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{tcp}$, the pixel coordinates of the reference hole, are converted to the reference

coordinate points under the robot TCP, $\begin{bmatrix} x_{Cal} \\ y_{Cal} \\ z_{Cal} \end{bmatrix}_{pix}$ is the reference pixel coordinate for identification,

$T^{Cam2Tcp}$ represents the transformation matrix from camera coordinate system to TCP, which can be calculated by conversion Equation (3).

$$[T^{Cam2Tcp}]^{-1} = [T^{pix2pic}]^{-1} \times [T^{pic2cam}]^{-1} \times [T^{cam2board}]^{-1} \times [T^{board2tcp}]^{-1} \quad (3)$$

Further, $\begin{bmatrix} x_{Cal} \\ y_{Cal} \\ z_{Cal} \end{bmatrix}_{pix}$ can be expressed by Equation (4).

$$\begin{bmatrix} x_{Cal} \\ y_{Cal} \\ z_{Cal} \end{bmatrix}_{pix} = \begin{bmatrix} x_{Rstd} - x_e \\ y_{Rstd} - y_e \\ z_{Rstd} - z_e \end{bmatrix}_{pix} \quad (4)$$

where $(x_{Rstd}, y_{Rstd}, z_{Rstd})$ is the absolute standard value of the reference hole on the work-piece in the image pixel coordinates, and (x_e, y_e, z_e) is the error between the actual identification of the reference hole and the absolute standard value. The difference between the two is the actual identified usable value. From Equation (5), we can analyze that when the value of (x_e, y_e, z_e) infinitely approaches 0, the value of $(x_{Cal}, y_{Cal}, z_{Cal})$ is closer to the absolute standard value.

$$\lim_{x_e, y_e, z_e \rightarrow 0} \begin{bmatrix} x_{Cal} \\ y_{Cal} \\ z_{Cal} \end{bmatrix}_{pix} - \begin{bmatrix} x_{Rstd} \\ y_{Rstd} \\ z_{Rstd} \end{bmatrix}_{pix} = 0 \quad (5)$$

The reference hole recognition algorithm is used to identify the pixel coordinates of the image, $(x_{Cal}, y_{Cal}, z_{Cal})$; error value compensation further reduces the error of benchmark hole fitting and improves the overall benchmark detection accuracy.

$T^{pix2pic}$, $T^{pic2cam}$, $T^{cam2board}$, $T^{board2tcp}$ are the transformation matrix relationships from pixel to image, image to camera, camera to calibration board, and calibration board to TCP. $T^{pix2pic}$ is the transformation matrix from the image coordinate system (x, y) to pixel coordinate system (u, v) , which is calibrated by checkerboard. The camera intrinsic parameters are calculated by Equation (6), and the pixel coordinates can be converted into image physical coordinates through the calculated camera intrinsic parameters.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/dx & 0 & u_0 \\ 0 & 1/dy & v_0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (6)$$

$T^{pic2cam}$ represents the transformation matrix relationships from (X_c, Y_c, Z_c) to image coordinate system (x, y) , which is calculated by Equation (7).

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (7)$$

$T^{cam2board}$ represents the transformation matrix relationships from (X_c, Y_c, Z_c) to the calibration plate coordinate system (X_b, Y_b, Z_b) , which is calculated by Equation (8).

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ O & 1 \end{bmatrix} \times \begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} \quad (8)$$

Establish the calibration plate coordinate system on the calibration plate and take two points in the calibration plate coordinate system, $P1 = (X_{b1}, Y_{b1}, Z_{b1})$, $P2 = (X_{b2}, Y_{b2}, Z_{b2})$ and $Z_{b1} = Z_{b2} = 0$, then the camera coordinate point can be obtained as P'_1 and P'_2 :

$$\begin{cases} [P'_1, 1] = [R, T] \times [X_{b1}, Y_{b1}, Z_{b1}]^T \\ [P'_2, 1] = [R, T] \times [X_{b2}, Y_{b2}, Z_{b2}]^T \end{cases} \quad (9)$$

which can be rewritten as:

$$\begin{cases} R_{31}X_{b1} + R_{32}Y_{b1} + T_z = 0 \\ R_{31}X_{b2} + R_{32}Y_{b2} + T_z = 0 \end{cases} \quad (10)$$

According to the calibration method of Zhang Zhengyou [23], the R and T matrix obtained can be used to calculate the normal vector of the camera plane A, B, C.

$T^{board2tcp}$ represents the transformation matrix relationships from calibration plate coordinate system (X_t, Y_t, Z_t) to the end of the mobile robot automatic drilling TCP coordinate system (X_b, Y_b, Z_b) , expressed as

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \\ 1 \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & T_{3 \times 1} \\ O & 1 \end{bmatrix} \times \begin{bmatrix} X_t \\ Y_t \\ Z_t \\ 1 \end{bmatrix} \quad (11)$$

4. Benchmark Detection Method Integrating Deep Learning and Improved Huber Benchmark Fitting

Among the traditional circle detection methods, the Hough transform circle detection method [24] is a typical circle detection algorithm. It is a traditional detection method based on parameter space search. It maps the geometric characteristics of the circle in the image to the parameter space and determines the parameters of the circle (center and radius) by voting based on the gradient direction of the edge points. Compared with the direct geometric fitting method, the Hough transform circle detection method has the advantages of good robustness and wide adaptability. Therefore, the Hough transform circle detection method is an important detection method in many industrial and visual applications. In subsequent experiments, the Hough transform circle detection method is also used as a traditional detection method for comparison. Its main steps are to parameterize the geometric equation of the circle, map the edge detection information to the parameter space, and vote to determine the position and size of the circle. The standard equation of

the circle is Equation (12), where a and b are the coordinates of the center of the circle, and r is the radius of the circle.

$$(x - a)^2 + (y - b)^2 = (r)^2 \quad (12)$$

The detection principle is to use the canny edge detection algorithm to extract the edge of the image. For each edge point (x, y) , the possible center and radius range are defined in the parameter space (a, b, r) . A voting mechanism is used to generate a series of possible (a, b, r) values in the parameter space for each edge point, and vote for these parameter combinations. For a known radius r , the center of the circle is calculated as $a = x \pm r \cos(\theta)$, $b = y \pm r \sin(\theta)$, where θ is the edge gradient direction. Finally, the number of votes for the parameter combination counted by the accumulator is analyzed, and the point with the highest vote is considered to be the detected circle.

Based on the fusion of deep learning and improved Huber benchmark fitting algorithm, the benchmark hole area is quickly located through deep learning object detection to achieve initial positioning. The edge feature points are further accurately extracted through image sampling, projection, differential filtering and polarity classification methods [25–27]. The correct polarity is selected according to the detection direction, and then the edge feature points are screened according to the preset threshold to obtain the edge feature points. The screened feature points are fitted with a circle to achieve precise positioning.

4.1. Fusion of Deep Learning and Improved Huber Benchmark Fitting Algorithm

The overall process of deep learning fusion benchmark feature detection is shown in Figure 4, which mainly includes five steps: image preprocessing, initial positioning, repositioning, result output, and calculation of hole position error correction matrix. Image preprocessing is employed to reduce the noise of the image and make the image smoother; initial positioning uses YOLOv5 deep learning [28] technology to perform initial positioning of the benchmark, and the initial positioning center position of the target benchmark is exported as the image feature sampling and projection interval for repositioning; repositioning uses the benchmark center position output by the initial positioning to determine the sampling area range, accurately extract the benchmark features, and use the random consistency method to initially screen the feature points and Huber final fitting benchmark center.

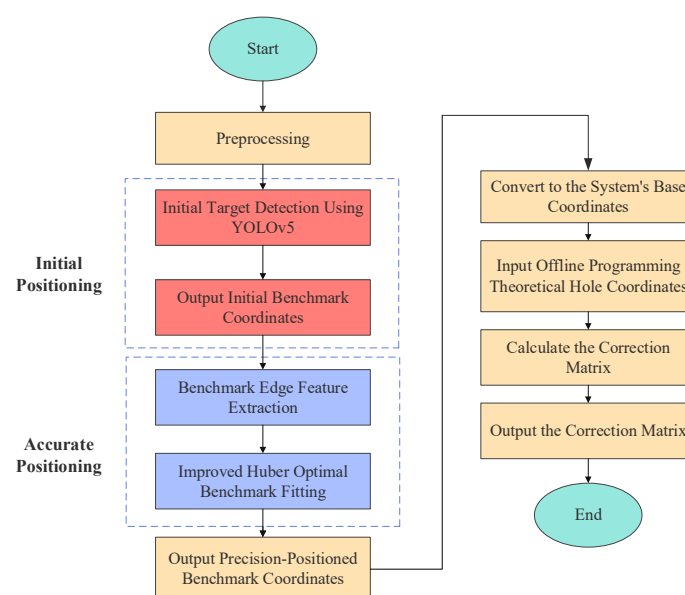


Figure 4. Overall flow chart of deep learning-based benchmark feature detection.

In the context of various forms of benchmark detection in aircraft assembly drilling, compared with the traditional Hough transform benchmark detection algorithm, the fusion of deep learning and improved Huber benchmark fitting algorithm has achieved significant improvement in robustness. The traditional Hough transform circle detection algorithm relies on edge detection results, which is highly sensitive to image noise, parameter settings and edge integrity. It is easy to produce false detections due to noise points, or miss detections due to occlusion and incomplete edges. The fusion of deep learning and the improved Huber benchmark fitting algorithm extracts features through the deep learning object detection model to complete the initial positioning, and performs outstandingly in suppressing noise and improving the robustness of detection. At the same time, the improved fitting benchmark strategy combines the initial positioning results to accurately locate the benchmark target area, reducing the dependence on edge integrity and parameter adjustment.

4.2. Initial Positioning of Assembly Benchmark Based on YOLOv5

In the process of benchmark hole detection, in order to improve the detection accuracy of assembly benchmark, the method based on YOLOv5 object detection [29] is used to realize the initial positioning detection of assembly benchmark. The initial position of assembly benchmark is first located by object detection, and then the projection method is used to extract features, and the improved Huber method is used to accurately locate and detect assembly benchmark. High-precision benchmark positioning is achieved through the two processes of initial positioning auxiliary detection and precise positioning detection. As a single-stage object detection algorithm, YOLOv5 has the advantage of combining speed and accuracy, providing technical support for industrial real-time detection.

Based on the analysis of the characteristics of the benchmark targets, this paper selects CSPDarknet53, PANet, and CNN as the network structures for Backbone, Neck, and Head, respectively, when building the network model architecture.

In the object detection model, the Backbone network is the core module for extracting image features. Its main function is to convert the input original image $X \in R^{C \times H \times W}$ into a multi-level feature map $F \in R^{C' \times H' \times W'}$ for subsequent detection tasks. The input image $X \in R^{C \times H \times W}$ is a three-dimensional tensor, which contains the two-dimensional spatial dimension width W and height H and channel dimension C information. The feature map $F \in R^{C' \times H' \times W'}$ is the feature map tensor, C' is the number of channels after convolution processing, H' and W' represent the smaller length and width of the feature map generated after convolution processing. The CSPDarknet53 network is optimized by using CSPNet (Cross Stage Partial Network) to improve the feature extraction efficiency. CSPNet is a neural network architecture optimization technique. The core idea of CSPNet is to divide the feature map into two parts, one of which is directly passed through, and the other is processed by the residual module, and then the two parts of the feature map are spliced together, which enhances the transmission of information flow and effectively reduces the computational burden. Its feature fusion formula can be expressed as Equation (13):

$$F_{CSP} = \text{Concat}\left(F_{\text{direct}}, \text{Residual}\left(\text{Conv}\left(F_{\text{split}}\right)\right)\right). \quad (13)$$

F_{direct} is a directly passed feature map, which is the feature directly output after passing a certain stage of the network. F_{split} is the part into which the input feature map X is divided, and the features are obtained after being processed by the convolutional layer and the residual structure. $\text{Residual}\left(\text{Conv}\left(F_{\text{split}}\right)\right)$ is a feature map processed by convolution and residual modules, which enhances the depth and information transfer of the network. Concat is the directly transmitted features, and the residual processed features are concatenated in the channel dimension to form the final fused feature map.

The Neck is an intermediate module in the object detection network. Its main function is to fuse features of different scales to support the detection of objects of different sizes. This paper adopts PANet as the Neck structure. PANet enhances the information transmission of low-level features by introducing a bottom-up path, and combines the top-down path to achieve efficient fusion of multi-scale features. Its feature fusion mechanism can be expressed by Equation (14):

$$F_{fusion} = Concat\left(TopDown(F_{high}), BottomUp(F_{low})\right) \quad (14)$$

F_{high} is the upper-layer feature map (usually with smaller spatial size and richer semantic information). F_{low} represents lower-level feature maps (usually with larger spatial size and more detailed information). *TopDown* is a top-down path that is used to gradually transfer high-level features to low-level features, thereby enhancing the semantic information of low-level features. *BottomUp* is a bottom-up path that gradually transfers low-level features to high-level features to ensure that the detailed information of the low level can be fully utilized. *Concat* is used to splice the feature maps from different paths in the channel dimension to form a multi-scale fused feature map. The Head is the final module of the object detection model and is responsible for performing object classification, bounding box regression, and confidence prediction. This paper adopts a convolutional neural network (CNN) as the network structure of the Head. By inputting multi-scale fusion features into the CNN, the network can perform object classification and bounding box regression. The final output is Equation (15):

$$\{P_{class}, B_{box}, P_{conf}\} = CNN(F_{fusion}) \quad (15)$$

where P_{class} is the classification probability, which represents the probability distribution of the category to which each candidate box in the input image belongs. B_{box} is a bounding box regression, which represents the exact position and size of the object (usually represented by four parameters x, y, w, h , which are the center coordinates, width and height of the box, respectively). P_{conf} is prediction confidence, which indicates the model's confidence in the existence of the object, and usually expresses the probability of whether the object exists. $CNN(F_{fusion})$ inputs the fused feature map F_{fusion} into the convolutional neural network to perform object classification, regression and confidence prediction.

Through the efficient feature extraction of the Backbone, the multi-scale feature fusion of the Neck, and the accurate detection output of the Head, the entire network can complete the task of efficient and robust object detection. The mutual cooperation and functional division of each module enables the model to handle objects of different scales and complexities at the same time.

4.3. Surrounding Benchmark Sampling Projection Feature Extraction Algorithm

The Surrounding Benchmark Sampling Projection Feature Extraction Algorithm demonstrates novelty in two key aspects. **Precise Sampling Location:** The method evenly distributes sampling points around the target center, narrowing the sampling area to enhance feature extraction accuracy. **Robustness to Noise and Outliers:** The algorithm incorporates a projection strategy that effectively filters out noise and minimizes the impact of outliers, improving its robustness in real-world applications. Surrounding reference sampling projection feature extraction requires image sampling, projection, differential filtering and polarity classification. Determine the sampling direction for the target area, and according to the set sampling size $L \times W$, obtain a $L \times W$ (length $\times$ width) size image along the sampling direction. We project the sampled image to obtain a $1 \times L$ size image, and calculate the average intensity of each projection line, that is, the average grayscale

value. The average intensity obtained by projection processing can reduce the probability of detection failure caused by noise in the measurement sampling area. According to the sampling image in Figure 5, the average intensity values of the projection line can be calculated as 255, 218, 142, 44, 11, 0.

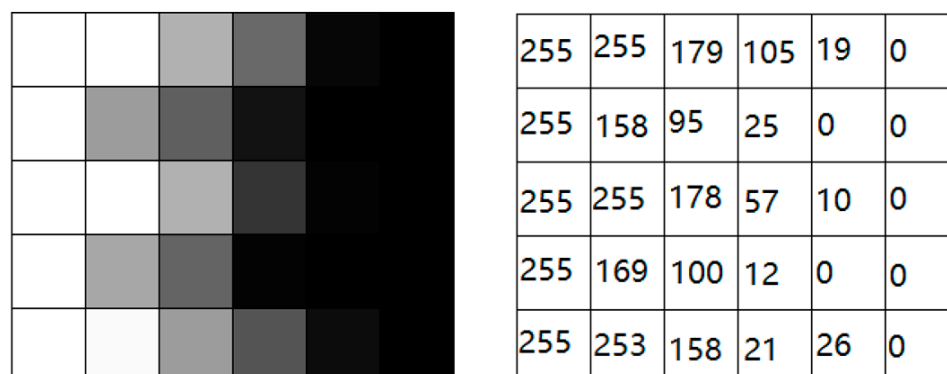


Figure 5. Sampling diagram.

The calculated average intensity value is subjected to a separate convolution operation, and two one-dimensional convolution kernels are used for filtering operations. Differential filtering is performed in the horizontal and vertical directions, respectively, and it can be separated into two independent one-dimensional filters in the x direction and the y direction. Specifically, we first apply a 1D convolution kernel (K_x) in the x direction, and then we apply a second 1D convolution kernel (K_y) in the y direction to the result of the first convolution. The formula is shown in the following Equation (16):

$$Filter2D(I, K) = filter2D(filter2D(I, K_x), K_y) \quad (16)$$

where I is the average intensity of the input, K is the convolution kernel, K_x and K_y are the one-dimensional convolution kernels in the horizontal and vertical directions, with sizes of $1 \times n$ and $m \times 1$. Execute the differential processing Equation (17), and the differential processing can eliminate the influence caused by the change of the absolute intensity value in the center of the measurement area.

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x) \quad (17)$$

Referring to Figure 5, there is no change in the shadow, the absolute value of the intensity is 0, and the change is -255 when the color changes from a grayscale value of 255 to a grayscale value of 0. The more obvious the grayscale difference value in the sampling image, the greater the deviation value obtained by differentiation, and the edge intensity waveform is shown in Figure 6.

Set the edge threshold and edge polarity value, and filter out interference edges by using the edge threshold preset in the differential waveform. Edges greater than the set edge threshold are valid edge feature points and stored in the variable container. Edges have polarity, which is determined by the direction of edge detection, and the polarity parameters can be set according to the detection requirements. Figure 7 shows polarity and threshold screening, which are threshold screening for positive polarity and threshold screening for negative polarity.

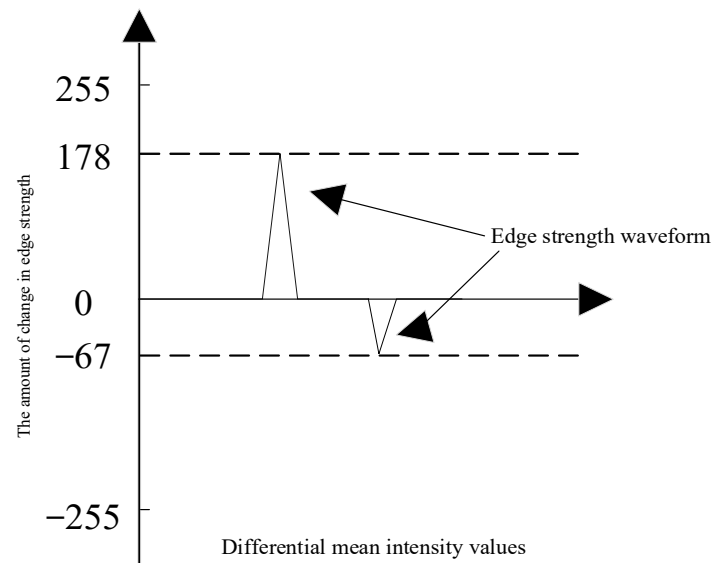


Figure 6. Edge intensity waveform.

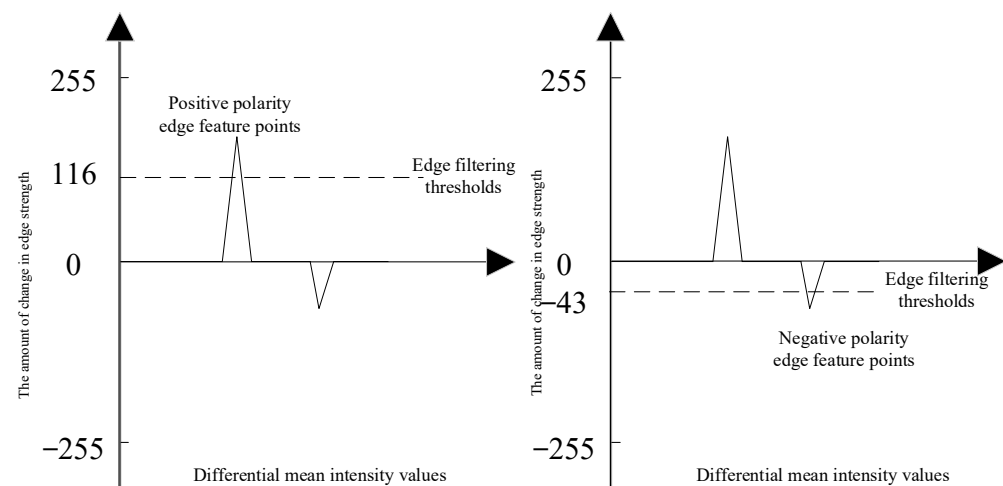


Figure 7. Polarity and threshold screening.

The output result can be obtained according to the following Equation (18):

$$g(x, y) = \begin{cases} 1, & |R(x, y)| \geq T \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

4.4. Improved Huber Circle Fitting Benchmark Identification Algorithm

The improved Huber fitting circle method adds the RANSAC [30] algorithm, performs a circle fitting screening on the extracted feature points to eliminate some interference points, and uses an iterative algorithm to correctly estimate the mathematical model parameters from a set of data containing interference points. Interference points generally refer to noise in the data, such as mismatches in matching and outliers in the estimated curve. After the RANSAC algorithm is used for rough screening, the Huber [31] ultra-precision fitting algorithm is used to fit the reference circle to obtain the optimal position information.

Point set $P = \{(x_1, y_1), (x_2, y_2), (x_3, y_3) \dots (x_n, y_n)\}$; randomly select three points $(x_{r1}, y_{r1}), (x_{r2}, y_{r2}), (x_{r3}, y_{r3})$, from the point set P , and fit a circle to the three randomly selected points using Equation (19), where (x_c, y_c) are the coordinates of the center of the circle and r is the radius.

$$(x - x_c)^2 + (y - y_c)^2 = r^2 \quad (19)$$

The Huber loss function can effectively reduce the impact of outliers on the fitting results, which can be expressed by Equation (20):

$$\rho(e_i) = \begin{cases} \frac{1}{2}e_i^2 & \text{for } |e_i| \leq \delta \\ \delta(|e_i| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \quad (20)$$

Among them, e_i represents the residual value from the point to the center of the fitted circle, and δ is the judgment threshold of the Huber loss function. The fitting error is the distance from each data point to the circle model. For a given circle (x_c, y_c, r) and data point (x_i, y_i) , the error can be obtained by calculating the distance from the point to the center of the circle minus the radius r . It is expressed as Equation (21):

$$e_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r \quad (21)$$

Equation (22) is the result of calculating the minimum sum of losses of the data point set.

$$\left(\min_{(x_c, y_c, r)} \right) \sum_{i=1}^m \rho \left(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r \right) \quad (22)$$

Improved Huber circle fitting benchmark identification algorithm implementation steps:

Step 1: Randomly select three points from the point set $P = \{(x_1, y_1), (x_2, y_2), (x_3, y_3) \dots (x_n, y_n)\}$ and record them as (x_{r1}, y_{r1}) , (x_{r2}, y_{r2}) and (x_{r3}, y_{r3}) .

Step 2: Use Equation (19) to calculate the center (x_c, y_c) and radius r of the fitted circle.

Step 3: For each point $P = \{(x_1, y_1), (x_2, y_2), (x_3, y_3) \dots (x_n, y_n)\}$ in the point set (x_i, y_i) , use Equation (21) to calculate the residual e_i with the fitting circle, and then use Equation (20) to calculate the corresponding Huber loss for each point (x_i, y_i) .

Step 4: Use the RANSAC algorithm to select the inliers, and select the inliers according to the Huber loss value of each point calculated in step 3.

Step 5: Use Equation (22) to optimize the fitted circle by minimizing the Huber loss, recalculate the center (x_c, y_c) and radius r , and further refine the fitting by weighted least squares method.

Step 6: Repeat steps 1–5, randomly sampling three points each time, fitting a new model, and calculating the set of inliers. Select the fitting circle with the largest number of inliers as the best model.

Step 7: Output the center (x_c, y_c) and radius r of the best-fitting circle.

The above steps realize the circle fitting of the Huber loss function, which reduces the influence of outliers on the fitting results. In the dataset with noise or outliers, the circle fitting using the Huber function can obtain more robust and accurate results.

5. Experimental Verification and Analysis

The benchmark detection experiment was carried out using the mobile robot automatic drilling system as a platform. Its visual system consists of hardware such as a 12 million high-pixel industrial camera, lens, and light source. The benchmark detection camera uses an area scan industrial camera with a camera resolution of 4096×3000 pixels, a white 90-degree ring light source, and a fixed-magnification lens. The processing experiment of the mobile robot automatic drilling system is shown in Figure 8.

At present, the existing hole and rivet assembly benchmark data are collected, and the collected assembly benchmark images are expanded by data enhancement. There are 200 samples of each type, and the samples are divided into a training set and test set in a ratio of 9:1.

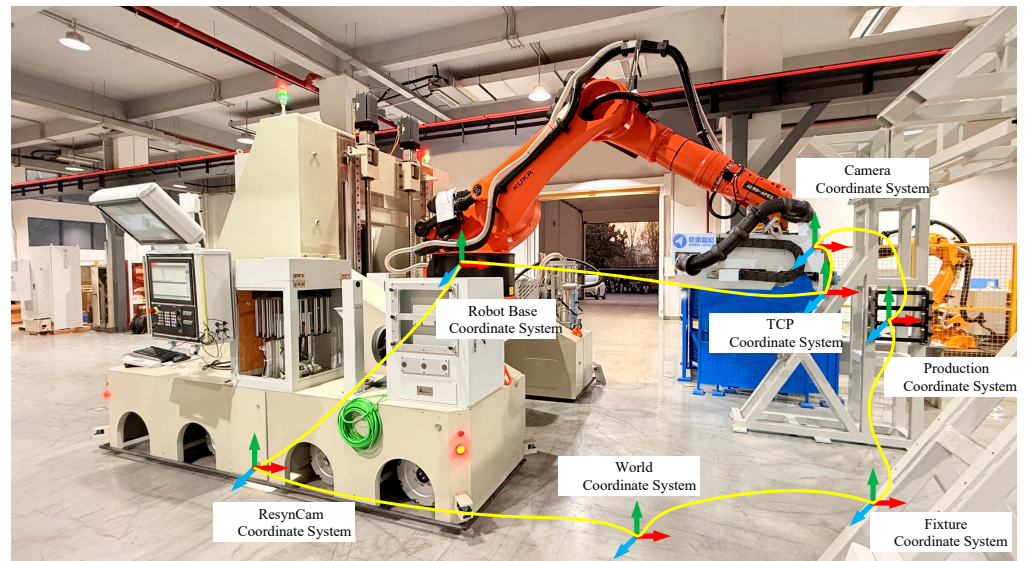


Figure 8. Mobile robot automatic drilling test system.

In this experiment, various performance indicators are analyzed through the training process of the YOLOv5 model. The pixel size of the image input is 640×640 , and the stochastic gradient descent (SGD) optimizer is used for training for a total of 250 rounds. The initial learning rate is set to 0.01, the batch size is 4, and the momentum is set to 0.937. Figures 9 and 10 show the changing trends of the mean average precision (mAP), precision, and recall of the model during the training process.

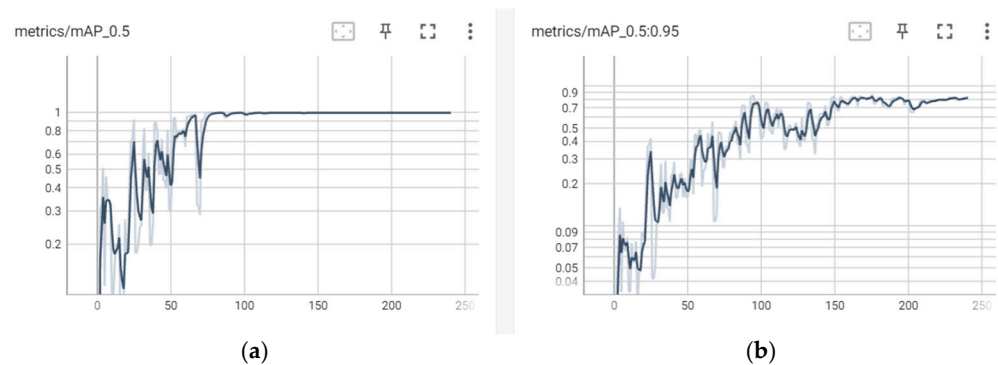


Figure 9. mAP average precision mean graph. (a) $mAP_{0.5}$ Mean average precision. (b) $mAP_{0.5:0.95}$ Mean average precision.

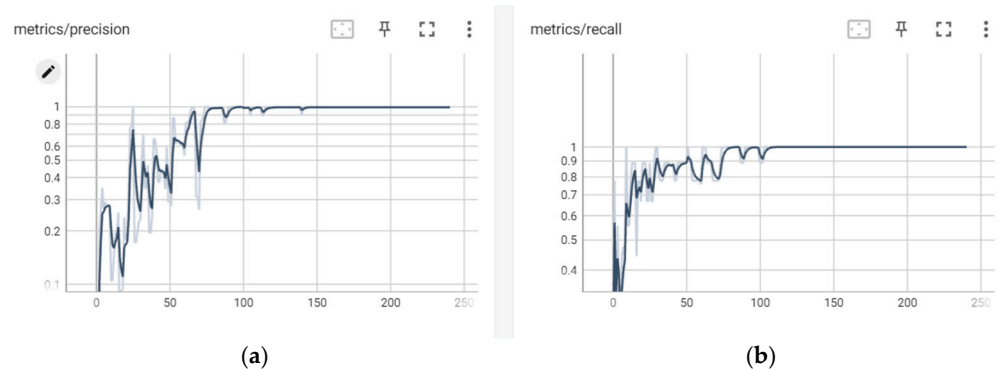


Figure 10. Precision and recall graph. (a) Precision. (b) Recall.

$mAP@0.5$ (Mean Average Precision at IoU = 0.5): This metric calculates the mean average precision with an Intersection over Union (IoU) threshold of 0.5. The formula for $mAP@0.5$ can be expressed by Equation (23):

$$mAP@0.5 = \frac{1}{N} \sum_{i=1}^N AP_i \quad (23)$$

where AP_i is the average precision for the i -th class, and N is the total number of classes. A predicted bounding box is considered correct if the IoU with the ground truth is greater than or equal to 0.5.

$mAP@0.5:0.95$ (Mean Average Precision at IoU thresholds from 0.5 to 0.95): This metric calculates the mean average precision over IoU thresholds from 0.5 to 0.95 with a step size of 0.05. It provides a more stringent evaluation of the model's performance over a range of IoU thresholds. The formula for $mAP@0.5:0.95$ is shown as Equation (24):

$$mAP@0.5 : 0.95 = \frac{1}{N} \sum_{i=1}^N \frac{1}{10} \sum_{t=0.5}^{0.95} AP_i(t) \quad (24)$$

where $AP_i(t)$ is the average precision for the i -th class at a given IoU threshold t , and the summation is performed over the 10 thresholds from 0.5 to 0.95.

Precision: Precision measures the proportion of true positive detections out of all the detections made by the model. The formula can be expressed by Equation (25):

$$Precision = \frac{TP}{TP + FP} \quad (25)$$

where TP is the number of true positives (correctly detected objects), and FP is the number of false positives (incorrectly detected objects).

Recall: Recall measures the proportion of true positives out of all the ground truth objects. The formula can be expressed by Equation (26):

$$Recall = \frac{TP}{TP + FN} \quad (26)$$

where TP is the number of true positives (correctly detected objects), and FN is the number of false negatives (ground truth objects missed by the model).

As can be seen from Figure 9a, $mAP_{0.5}$ rises rapidly within the first 100 rounds of training, and tends to stabilize after nearly 100 rounds, and finally stabilizes at a level close to 1. This shows that the model can accurately detect the object when the IoU threshold is 0.5, and has high detection performance. Although the $mAP_{0.5:0.95}$ shown in Figure 9b is relatively flat during the rise, it eventually stabilizes at above 0.8, indicating that the model still has good object detection capabilities under stricter IoU requirements (ranging from 0.5 to 0.95). The model can not only effectively detect under looser IoU threshold conditions, but also maintain efficient detection under stricter object positioning requirements, indicating that it has high positioning accuracy for objects and good generalization ability.

Precision and recall are important indicators for measuring the performance of object detection models. Precision indicates the proportion of samples predicted by the model to be positive that are actually positive, while recall measures the proportion of samples that the model can successfully identify as true positives.

As can be seen from Figure 10a, the accuracy fluctuates greatly in the first 50 rounds of training, but as the number of training rounds increases, the accuracy gradually stabilizes and eventually approaches 1. This shows that as the model parameters are continuously optimized, the number of false positive samples gradually decreases, and the model can more accurately classify and predict positive samples. As shown in Figure 10b, the recall

rate also fluctuates to a certain extent in the early stage of training, but gradually stabilizes at 1 after 50 rounds of training, indicating that the model can identify almost all objects with a very low missed detection rate.

Based on the above indicators, the performance of the trained model shows good convergence. Whether $mAP_{0.5}$ is close to 1 or $mAP_{0.5:0.95}$ is stable above 0.8, it shows that the model can stably detect the object under different IoU threshold conditions. In addition, the rapid improvement of precision and recall and the performance that eventually tends to 1 further verify the superiority of the model in positive category recognition.

Figure 11 shows the detection results on the validation set. By analyzing the results of the validation set, the effectiveness and robustness of the model in practical applications are further confirmed. The validation results show that the trained model can demonstrate high accuracy and stable detection capabilities, and has strong generalization performance.

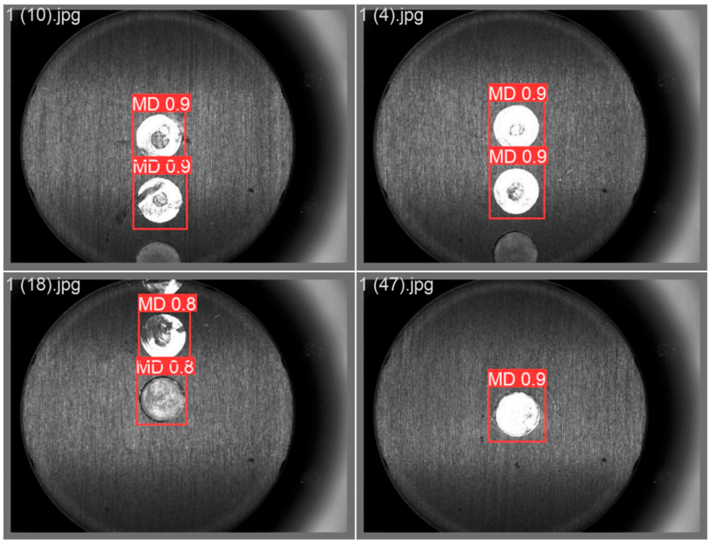


Figure 11. Validation set test diagram.

The proposed algorithm detects reference holes for covered holes, eccentric holes, flat head rivets, polished rivets, round head nails, cross nails and core clips. The comparison of various feature detection results is shown in Table 2, where the first column is the original image, the second column is the initial positioning result of YOLOv5 object detection, the third column is the result of the traditional Hough transform circle detection method, and the fourth column is the detection of the proposed method. The comparison of various feature detection results is shown in Figure 11. The effect diagrams shown in the figure are all local areas intercepted from the original size result diagram after calculation processing.

Table 2. Comparison of various feature detection results.

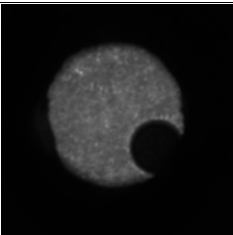
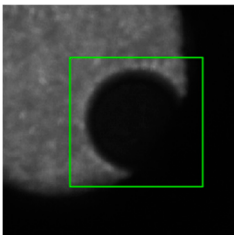
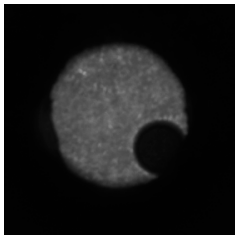
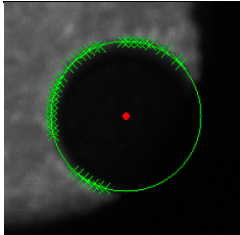
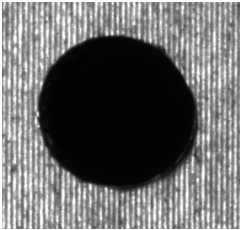
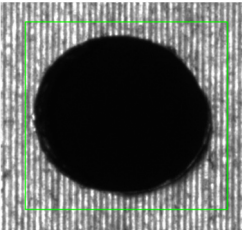
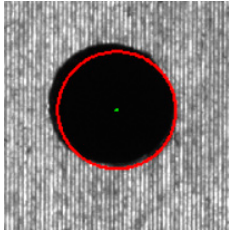
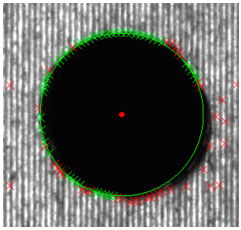
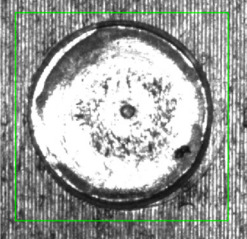
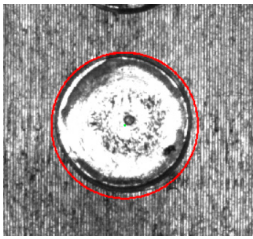
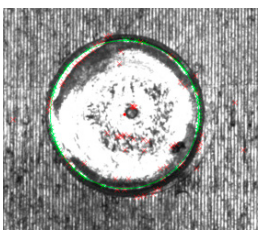
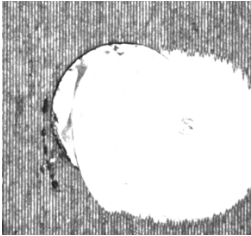
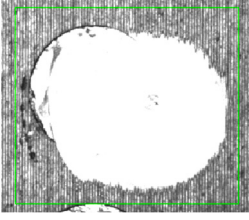
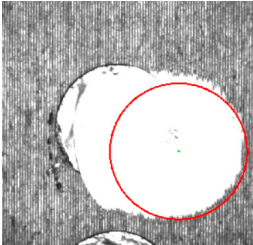
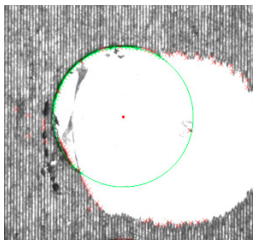
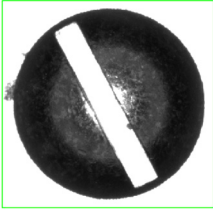
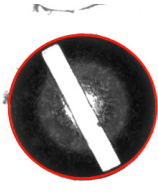
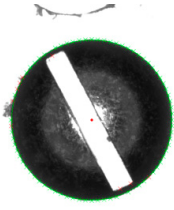
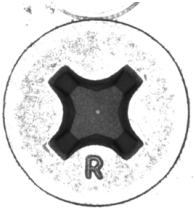
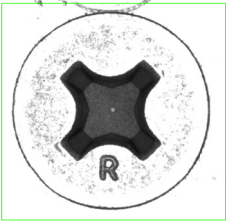
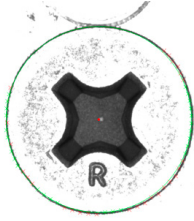
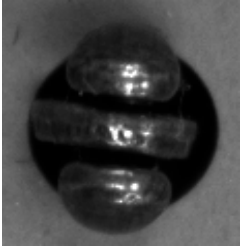
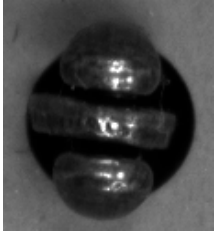
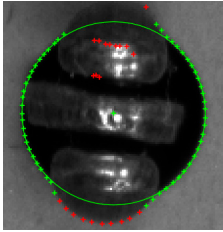
Benchmark Type	Original Image Features	YOLOv5 Initial Positioning Detection	Traditional Method Detection Effect	Proposed Detects the Effect
Covered hole				

Table 2. Cont.

Benchmark Type	Original Image Features	YOLOv5 Initial Positioning Detection	Traditional Method Detection Effect	Proposed Detects the Effect
Eccentric hole				
Flat head rivets				
Polished rivets				
Round head nail				
Cross nails				
Core clip				

5.1. Benchmark Recognition Effectiveness and Recognition Accuracy

Verification of the effectiveness of benchmark identification: The effectiveness verification of the recognition algorithm is mainly to verify whether the benchmark detection algorithm can accurately detect the assembly benchmark in the image. First, it is necessary

to complete the image acquisition work. The mobile robot automatic drilling system is used to collect various types of assembly benchmark images on the simulation parts, including occluded holes, eccentric holes, flat head rivets, polished rivets, round head nails, cross nails and through-hole clips. There are seven types of assembly benchmarks; each type has 30 test samples, as shown in Table 3. The comparison of the effectiveness of the traditional Hough transform circle detection method and the improved method in this paper is shown in Figure 12, and the number of identifiable benchmark features is counted, respectively.

Table 3. Experiment feature type and sample size.

Feature Type	Covered Hole	Eccentric Hole	Flat Head Rivets	Polished Rivets	Round Head Nail	Cross Nails	Core Clip
Sample size	30	30	30	30	30	30	30

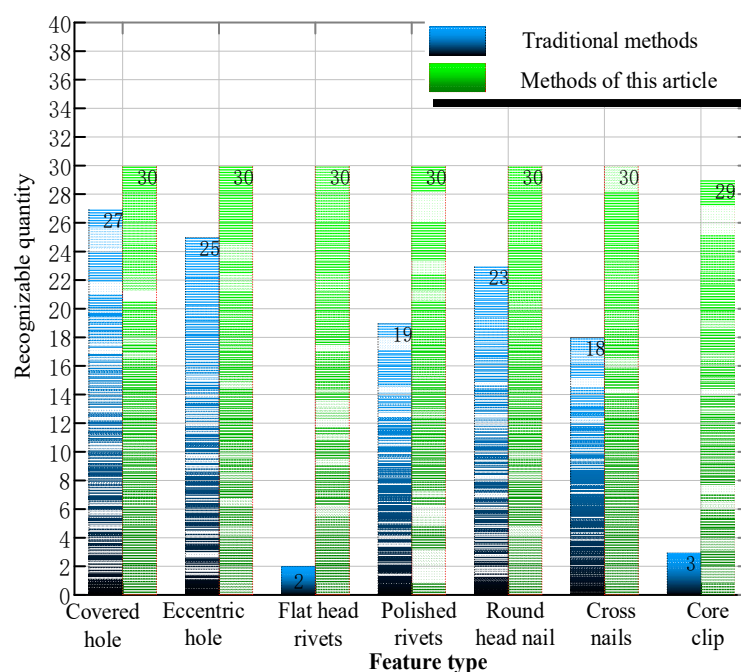


Figure 12. Statistics on the effectiveness of seven types of benchmark recognition.

The experimental verification prepared 30 samples for the above seven types of benchmarks for detection, totaling 210. Compared with the traditional detection method and the proposed method, the traditional detection method identified a total of 117 holes, and the proposed method identified a total of 209 holes, and the overall detection recognition rate increased by 43.8%. Among them, the recognition rates of blocked holes, eccentric holes, flat head rivets, polished rivets, round head nails, cross nails and through-hole clips increased by 10%, 16.6%, 93.3%, 36.6%, 23.3%, 40%, and 86.6%, respectively. The experimental results show that the proposed method has a good effect on benchmark feature recognition, can be used for heterogeneous benchmark feature recognition, and has better robustness.

Verification of benchmark recognition accuracy: To facilitate the comparison of the accuracy of benchmark center recognition, a through-hole benchmark with a higher recognition rate of the traditional Hough transform circle detection algorithm is selected, and 10 groups of benchmark image results are identified for calculation. The calculation method is used to calculate the square difference with the measurement results of the third-party measurement software VisionMaster (VisionMaster4.2.0) from Hikvision company (Hangzhou, China). The VisionMaster measurement software was selected as a benchmark for calculating variance due to its industry recognition and proven accuracy. By comparing

our method to this reference, we were able to assess its performance and demonstrate improvements in measurement accuracy. $P_{std}(x, y)$ is the measurement value of the third-party measurement software, $P_t(x, y)$ is the measurement result value of the traditional Hough transform circle detection method, and $P_i(x, y)$ is the measurement result value of the algorithm in this paper. The square difference between the measurement values of the two methods and $P_{std}(x, y)$ is calculated to obtain the error values E_t and E_i , as shown in Equation (27).

$$E = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (27)$$

By calculating 10 groups of benchmark samples, the values of benchmark X, Y and error values E_t and E_i are shown in Table 4. As shown in Figure 13, the algorithm error comparison shows that the difference between the results calculated by the algorithm in this paper and the third-party measurement software is small, with an average value of 0.848712 pixels, and the difference between the results calculated by the traditional Hough transform circle detection algorithm and the third-party measurement software is large, with an average value of 3.905571 pixels. Compared with the average error value, the recognition accuracy is improved by 78.26%, so the recognition accuracy of the algorithm in this paper is significantly higher than that of the traditional Hough transform circle detection algorithm.

Table 4. Comparison of reference coordinate differences.

The Algorithm Proposed in This Paper			Traditional Algorithms		
Benchmark X Difference (Pixel)	Benchmark Y Difference (Pixel)	E_i	Benchmark X Difference (Pixel)	Benchmark Y Difference (Pixel)	E_t
−0.923	−0.441	1.022942	2.137	0.089	2.138852
−0.749	−0.724	1.041718	2.631	−0.044	2.631368
0.495	−0.076	0.5008	2.605	0.623	2.678461
−0.689	0.28	0.743721	4.331	2.232	4.872308
−0.391	−0.501	0.635517	3.539	2.259	4.198524
0.864	0.338	0.927761	3.564	3.068	4.702629
−0.175	−0.464	0.495904	3.045	−0.614	3.106287
−1.138	−0.176	1.151529	6.352	2.404	6.791695
−0.295	−0.567	0.639151	1.845	2.461	3.0758
−1.328	−0.014	1.328074	4.722	1.149	4.859782

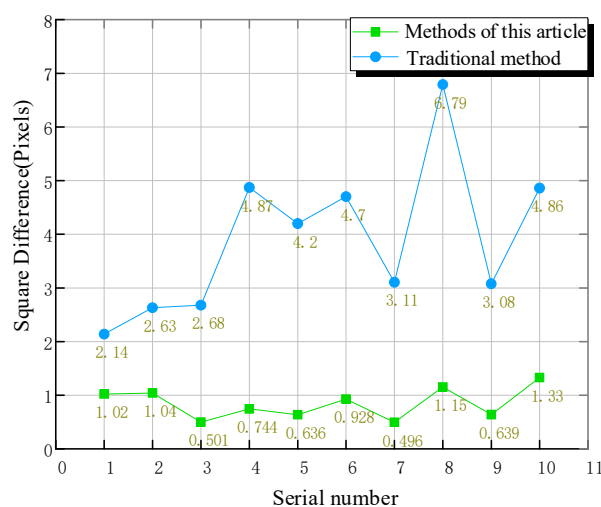


Figure 13. Algorithm error comparison.

5.2. Experimental Verification of Automatic Drilling Benchmark Detection for Mobile Robots

In order to verify the final drilling accuracy of the algorithm, a mobile robot automatic drilling system was used to perform benchmark detection and drilling experiments on the curved test drilling plate. The experimental drilling process diagram is shown in Figure 14a. The material of the plane wall panel is a composite material CF2P, with a thickness of 5 mm and a hole diameter of 5 mm. The mobile robot automatic drilling system was used to make two benchmark holes at an interval of 250 mm on the experimental wall panel, making two groups of upper and lower holes. Then, the improved algorithm in this paper and the traditional algorithm were used to perform benchmark feature hole recognition to perform benchmark detection and complete automatic drilling. Theoretical points were generated by offline programming, where the theoretical hole spacing was 25 mm. After the processing was completed, a laser tracker was used to measure the hole positions compensated by the benchmark feature recognition of the two algorithms. The Spatial Analyzer (SA) software was used to calculate and analyze the position accuracy of the holes.

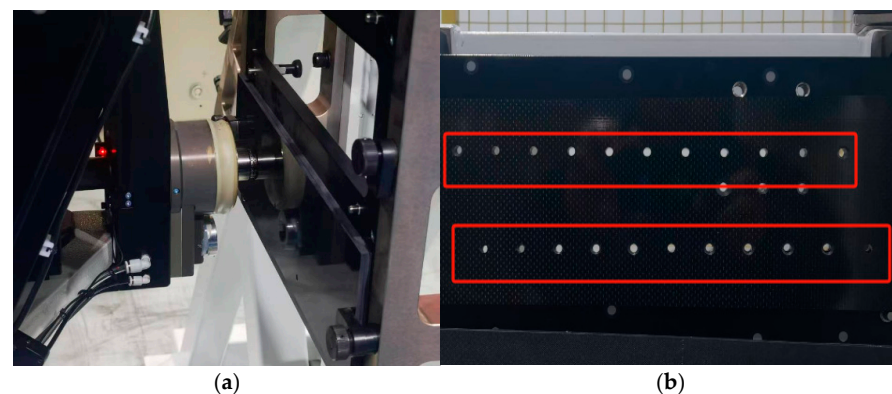


Figure 14. Experimental drilling process diagram. (a) Experimental drilling process. (b) Experimental drilling results.

The upper row in Figure 14b is the automatic drilling by the benchmark feature detection of the algorithm in this paper, and the lower row is the automatic drilling by the traditional Hough transform circle benchmark feature detection. By fitting the 11 points in each row to generate a straight line, and then calculating the distance value from each point to the straight line, and then analyzing the distance difference from each point to the fitted straight line, the consistency of the mobile robot automatic drilling system can be judged. The data are summarized in Table 5.

Table 5. Distance between hole positions and fitting straight lines.

Proposed Algorithm Distance (mm)	Traditional Algorithm Distance (mm)
0.08	0.49
0.04	0.09
0.05	0.05
0.01	0.07
0.04	0.12
0.02	0.06
0.03	0.04
0.08	0.06
0.04	0.1
0.06	0.08
0.19	0.56

From the results of calculation and analysis in Table 5, it can be seen that the distance of the straight line distance based on the algorithm proposed in this paper is less than 0.19 mm, compared with the distance of the straight line distance of the traditional algorithm of less than 0.56 mm, the maximum discrete distance of the hole position made by the algorithm in this paper is increased by 0.37 mm, indicating that the straightness of the hole is better.

The actual measurement value of 11 points in each row is further calculated to calculate the spacing between two adjacent holes, and the difference is calculated by comparing with the standard hole spacing. The standard spacing is the actual distance between the two measured benchmarks divided into 10 parts as the standard spacing value, and is used as the standard reference value for calculating the difference. After calculation and analysis, as shown in Figure 15, the maximum difference between the measured hole spacing of the algorithm in this paper and the corresponding standard spacing is 0.29 mm, while the maximum difference between the hole spacing of the traditional Hough transform circle algorithm and the measured value is 0.64 mm. Comparative analysis shows that the accuracy of automatic drilling by the robot using the benchmark detection algorithm in this paper has been significantly improved, and the data are summarized in Table 6 below.

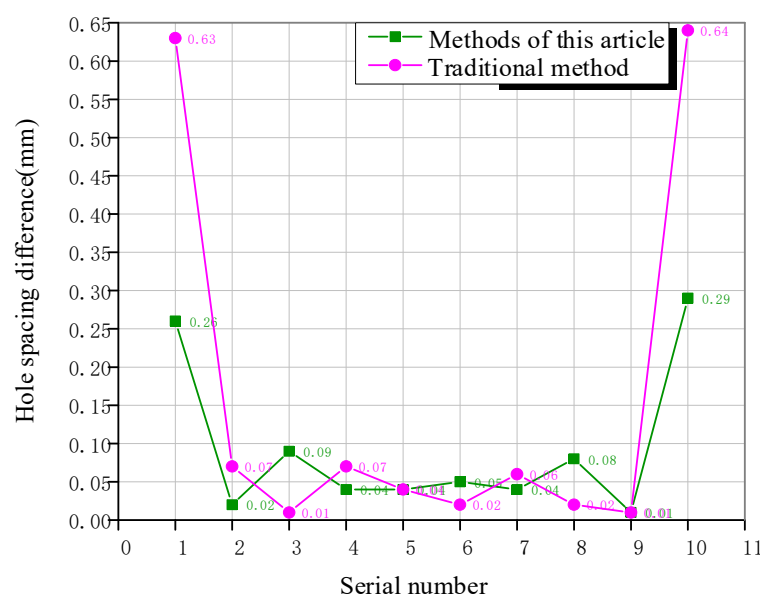


Figure 15. Comparison of actual hole spacing differences of the mobile robot automatic drilling system.

Table 6. The difference in hole spacing between reference holes of curved wall panels.

Standard Distance (mm)	Proposed Algorithm Distance (mm)	Difference (mm)	Standard Distance (mm)	Traditional Algorithm Distance (mm)	Difference (mm)
25.0055	24.74	0.26	25.0032	25.65	−0.63
25.0055	24.98	0.02	25.0032	25.08	−0.07
25.0055	24.91	0.09	25.0032	25.01	−0.01
25.0055	25.05	−0.04	25.0032	24.93	0.07
25.0055	25.05	−0.04	25.0032	24.96	0.04
25.0055	24.95	0.05	25.0032	24.98	0.02
25.0055	24.96	0.04	25.0032	25.07	−0.06
25.0055	25.09	−0.08	25.0032	25.03	−0.02
25.0055	25.00	0.01	25.0032	24.99	0.01
25.0055	25.30	−0.29	25.0032	24.36	0.64

6. Conclusions

Aiming at the needs of complex and diverse benchmark types in actual engineering applications of aircraft automated assembly, this paper proposes and studies a benchmark hole detection method that integrates a deep learning and projection method to extract features and perform quadratic fitting for precise positioning. The use of deep learning object detection to initially identify the position of benchmark holes solves the problem of being unable to meet the needs of diverse benchmark types and avoids limiting the selection of a single type of benchmark for benchmark detection hole positions during hole processing. At the same time, the method of precise positioning feature extraction and fitting the center of the benchmark hole is improved, which can accurately extract sub-pixel feature points based on the sudden change of edge gradients, ensuring the data source of the fitting benchmark center. Finally, the improved Huber fitting algorithm is used to fit the benchmark data to obtain the benchmark center. Through the verification of the effectiveness and accuracy of benchmark recognition, the overall recognition rate of benchmark recognition effectiveness has increased by 43.8%, and the recognition accuracy has increased by 78.26%. In the experimental drilling plate experiment, the improved algorithm in this paper has improved the accuracy of benchmark feature recognition, thereby effectively improving the accuracy compensation of benchmark detection. The difference between the theoretical hole spacing and the actual measured hole spacing in the drilling process is between -0.29 mm and 0.26 mm, and the absolute difference is less than 0.29 mm. Compared with the traditional algorithm, the absolute difference is reduced by 54.69%.

The experiment proves that after the accuracy of benchmark feature recognition is improved by the algorithm in this paper, the processing accuracy of the entire mobile robot automatic drilling system is compensated, ensuring the high-precision application of the mobile robot in a dynamic working environment. Therefore, by optimizing the detection and recognition method of benchmark features, the processing accuracy of the mobile robot automatic drilling system can be significantly improved.

Although significant progress has been made in this study with respect to the current materials and reference features, future work will focus on further optimizing the algorithm to adapt to a broader range of material types and reference types. Specifically, the system will maintain high robustness when dealing with materials of varying hardness, surface quality, or strong reflective properties, as well as right-angle edge benchmarks. Future research will concentrate on enhancing the algorithm's adaptability, ensuring stable operation in dynamic and ever-changing manufacturing environments.

Author Contributions: Conceptualization, J.D. and J.S.; Methodology, validation, and writing—original draft preparation, J.D. and X.C.; Writing—review and editing, all authors; Supervision, P.L. and W.T.; Project administration, P.L. and W.T.; Funding acquisition, W.T. All authors have read and agreed to the published version of the manuscript.

Funding: Project supported by the National Natural Science Foundation of China (grant number U22A20204 and 52205530), and the National Defense Basic Scientific Research program of China (grant number JCKY2023205A003).

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors gratefully acknowledge H.L. for his help with data collection.

Conflicts of Interest: Authors Jialong Dai and Xiangshun Cui were employed by the company Jiangsu AERO-APEX Intelligent Technologies Co., Ltd. Author He Liu was employed by the company AVIC Jiangxi Hongdu Aviation Industry Group Company Ltd. The remaining authors declare that the

research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zhou, F.; Xue, H.; Zhou, W.; Xu, G. Key technologies and development of aircraft digital flexible assembly. *Aeronaut. Manuf. Technol.* **2006**, *9*, 30–35.
2. Du, Z.; Yao, Y.; Wang, J. Research status and development trends of robot drilling and riveting system. *Aeronaut. Manuf. Technol.* **2015**, *4*, 26–31.
3. Guo, F.; Liu, J.; Wang, Z.; Zou, F.; Zhao, X. Positioning error guarantee method with two-stage compensation strategy for aircraft flexible assembly tooling. *J. Manuf. Syst.* **2020**, *55*, 285–301. [\[CrossRef\]](#)
4. Lian, Z.; Guo, Y.; Hou, C.; Zhao, S.; Dong, H. Optimization of robot drilling perpendicularity based on contact-type pressure foot. *J. Nanjing Univ. Aeronaut. Astronaut.* **2023**, *55*, 418–426.
5. Wang, Z.; Zhang, R.; Keogh, P. Real-Time Laser Tracker Compensation of Robotic Drilling and Machining. *J. Manuf. Mater. Process* **2020**, *4*, 79. [\[CrossRef\]](#)
6. Tian, J.; Wang, L.; Mei, B.; Zhu, W. Hole feature detection for aircraft parts by integrating visual saliency and group decision making. *Opt. Precis. Eng.* **2024**, *32*, 125–136. [\[CrossRef\]](#)
7. Liang, Z. Modeling and Synthesis of Positioning Variation in Robotic Drilling. Master's Thesis, Zhejiang University, Hangzhou, China, 25 June 2021.
8. Li, R.; Ding, N.; Zhao, Y.; Liu, H. Real-time trajectory position error compensation technology of industrial robot. *Measurement* **2023**, *208*, 112418. [\[CrossRef\]](#)
9. Kleber, R.; Emília, V.; Wesley, R.; Augusto, D. Comparison of visual servoing technologies for robotized aerospace structural assembly and inspection. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102237.
10. Wang, Q.; Zheng, S.; Li, J.; Ke, Y.; Chen, L. A correction method for hole positions based on hole margin constraints and Shepard interpolation. *Acta Aeronaut. Et Astronaut. Sin.* **2015**, *36*, 4025–4034.
11. Sun, X.; Tian, W.; Hu, J.; Liao, W. A hole position correction method of interpolation Coons surface based on genetical gorithm. *J. Beijing Univ. Aeronaut. Astronaut.* **2021**, *47*, 1814–1822.
12. Abdulla, A.; Mohamad, H.; Dewald, S.; Rajkumar, M.; Fahad, A.; Yahya, Z. Neuromorphic vision based control for the precise positioning of robotic drilling systems. *Robot. Comput. Integr. Manuf.* **2023**, *79*, 102419.
13. Frommknecht, A.; Kuehnle, J.; Effenberger, I.; Pidan, S. Multi-sensor measurement system for robotic drilling. *Robot. Comput. Integr. Manuf.* **2017**, *47*, 4–10. [\[CrossRef\]](#)
14. Zhuang, Z.; Tian, W.; Li, B.; Shi, H.; Du, X. Detection algorithm of hole position and normal based on template matching. *Comput. Integr. Manuf. Syst.* **2021**, *27*, 3485–3493.
15. Tan, X.; Tang, J.; Yu, W.; Li, X.; Sun, Z. Research on reference hole detection technology based on line laser scanning and image processing. *Mod. Manuf. Eng.* **2019**, *4*, 115–121.
16. Liu, Y.; Shi, Z.; Yuan, P.; Chen, D.; Lin, M.; Li, Z. A visual positioning and measurement system for robotic drilling. In Proceedings of the 2016 IEEE 14th International Workshop on Advanced Motion Control (AMC), Auckland, New Zealand, 22–24 April 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 461–466.
17. Yan, Q.; Chen, W. Automatic Modification of Local Drilling Holes via Double Pre-Assembly Holes. *World J. Eng. Technol.* **2015**, *3*, 191–196. [\[CrossRef\]](#)
18. Zhang, H.; Cheng, W.; Wang, M.; Pan, J.; Huang, W. Light weight Mobile Hole drilling System Positioning Method Based on Redundant Reference Holes. *China Mech. Eng.* **2021**, *32*, 2697–2704.
19. Du, T.; Wang, M.; Chen, W. Robust detection method of multi type assembly reference hole based on monocular vision. *Acta Aeronaut. Et Astronaut. Sin.* **2023**, *44*, 427852.
20. Dong, S.; Zheng, K.; Meng, D.; Liao, W.; Sun, L. Robotic drilling of large complex components: A review. *Acta Aeronaut. Et Astronaut. Sin.* **2022**, *43*, 31–48+2.
21. Shi, Z.; He, X.; Zeng, D.; Lei, P. Error compensation method for mobile robot positioning based on error similarity. *Acta Aeronaut. Et Astronaut. Sin.* **2020**, *41*, 428–439.
22. He, X.; Tian, W.; Zeng, Y.; Liao, W.; Xiang, Y. Robot positioning error and residual error compensation for aircraft assembly. *Acta Aeronaut. Et Astronaut. Sin.* **2017**, *38*, 292–302.
23. Zhang, Z. Flexible camera calibration by viewing a plane from unknown orientations. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; IEEE: Piscataway, NJ, USA, 1999; pp. 666–673.
24. Pang, Y. Research on Circle Detection Algorithm Based on Conformal Geometric Algebra. Master's Thesis, Civil Aviation University of China, Tianjin, China, 16 August 2024.

25. Yin, Z.; Wang, Z.; Fan, C.; Wang, X.; Qiu, T. Edge Detection via Fusion Difference Convolution. *Sensors* **2023**, *23*, 6883. [[CrossRef](#)] [[PubMed](#)]
26. Zhang, T.; Wei, S.; Ji, S. E2EC: An End-to-End Contour-based Method for High-Quality High-Speed Instance Segmentation. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 27 September 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 4443–4452.
27. Su, Z.; Liu, W.; Yu, Z.; Hu, D.; Liao, Q.; Tian, Q.; Pietikäinen, M.; Liu, L. Pixel Difference Networks for Efficient Edge Detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 5117–5127.
28. Zhang, J.; Zhang, J.; Zhou, K.; Zhang, Y.; Chen, H.; Yan, X. An Improved YOLOv5-Based Underwater Object-Detection Framework. *Sensors* **2023**, *23*, 3693. [[CrossRef](#)] [[PubMed](#)]
29. Zhang, C.; Yang, Z.; Liao, L.; You, Y.; Sui, Y.; Zhu, T. RPEOD: A Real-Time Pose Estimation and Object Detection System for Aerial Robot Target Tracking. *Machines* **2022**, *10*, 181. [[CrossRef](#)]
30. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Read. Comput. Vis.* **1987**, 726–740. [[CrossRef](#)]
31. Tong, H.Z. Functional linear regression with Huber loss. *J. Complex.* **2023**, *74*, 101693. [[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.