electronics

MDPI

Article

Research on UAV Autonomous Recognition and Approach Method for Linear Target Splicing Sleeves Based on Deep Learning and Active Stereo Vision

Guocai Zhang [1], Guixiong Liu [1,*] and Fei Zhong [2]

[1] School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China; 202210180398@mail.scut.edu.cn
[2] South Power Grid Technology Co., Ltd., Guangzhou 510080, China; zhongfei1118@163.com
* Correspondence: megxliu@scut.edu.cn; Tel.: +86-136-0964-2500

Abstract: This study proposes an autonomous recognition and approach method for unmanned aerial vehicles (UAVs) targeting linear splicing sleeves. By integrating deep learning and active stereo vision, this method addresses the navigation challenges faced by UAVs during the identification, localization, and docking of splicing sleeves on overhead power transmission lines. First, a two-stage localization strategy, LC (Local Clustering)-RB (Reparameterization Block)-YOLO (You Only Look Once)v8n (OBB (Oriented Bounding Box)), is developed for linear target splicing sleeves. This strategy ensures rapid, accurate, and reliable recognition and localization while generating precise waypoints for UAV docking with splicing sleeves. Next, virtual reality technology is utilized to expand the splicing sleeve dataset, creating the D_{SS} dataset tailored to diverse scenarios. This enhancement improves the robustness and generalization capability of the recognition model. Finally, a UAV approach splicing sleeve (UAV-ASS) visual navigation simulation platform is developed using the Robot Operating System (ROS), the PX4 open-source flight control system, and the GAZEBO 3D robotics simulator. This platform simulates the UAV's final approach to the splicing sleeves. Experimental results demonstrate that, on the D_{SS} dataset, the RB-YOLOv8n(OBB) model achieves a mean average precision (mAP0.5) of 96.4%, with an image inference speed of 86.41 frames per second. By incorporating the LC-based fine localization method, the five rotational bounding box parameters (x, y, w, h, and *angle*) of the splicing sleeve achieve a mean relative error (MRE) ranging from 3.39% to 4.21%. Additionally, the correlation coefficients (ρ) with manually annotated positions improve to 0.99, 0.99, 0.98, 0.95, and 0.98, respectively. These improvements significantly enhance the accuracy and stability of splicing sleeve localization. Moreover, the developed UAV-ASS visual navigation simulation platform effectively validates high-risk algorithms for UAV autonomous recognition and docking with splicing sleeves on power transmission lines, reducing testing costs and associated safety risks.

Keywords: splicing sleeve; rotational object detection; LC-RB-YOLOv8n(OBB); visual navigation; UAV

check for updates

Citation: Zhang, G.; Liu, G.; Zhong, F. Research on UAV Autonomous Recognition and Approach Method for Linear Target Splicing Sleeves Based on Deep Learning and Active Stereo Vision. *Electronics* **2024**, *13*, 4872. https://doi.org/10.3390/electronics13244872

Academic Editor: Silvia Liberata Ullo

Received: 16 November 2024
Revised: 7 December 2024
Accepted: 9 December 2024
Published: 10 December 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

With growing demands for modern industrial and infrastructure monitoring, unmanned aerial vehicles (UAVs) have become indispensable tools due to their efficiency and flexibility. UAVs play a critical role in the inspection and maintenance of power transmission lines. They have become a primary method for ensuring the safety and stability of transmission lines where high-definition cameras are often employed to assess key components visually [1–3]. However, for splicing sleeves and other crimped metal fittings on transmission lines, high-payload UAVs are required to carry portable digital radiography (DR) equipment. This task demands skilled UAV operators to carefully align and dock the

equipment along the splicing sleeve's axis until it is accurately positioned and suspended over the splicing sleeve, after which DR inspection of the sleeve is performed [4–6].

Figure 1 illustrates the process of a high-payload quadrotor UAV carrying DR equipment during takeoff, approach, aerial descent, and docking under the operator's control. The approach, docking, and successful suspension are critical steps for the UAV to participate in the DR inspection successfully. To assist the operator in achieving a precise UAV approach and DR equipment docking at the final stage, three key challenges must be addressed: (1) reliable aerial identification of the splicing sleeve by the UAV; (2) accurate measurement of the splicing sleeve's position, including the relative distance and spatial orientation between the UAV and the sleeve; (3) development of a robust docking approach strategy that ensures reliable and efficient suspension.

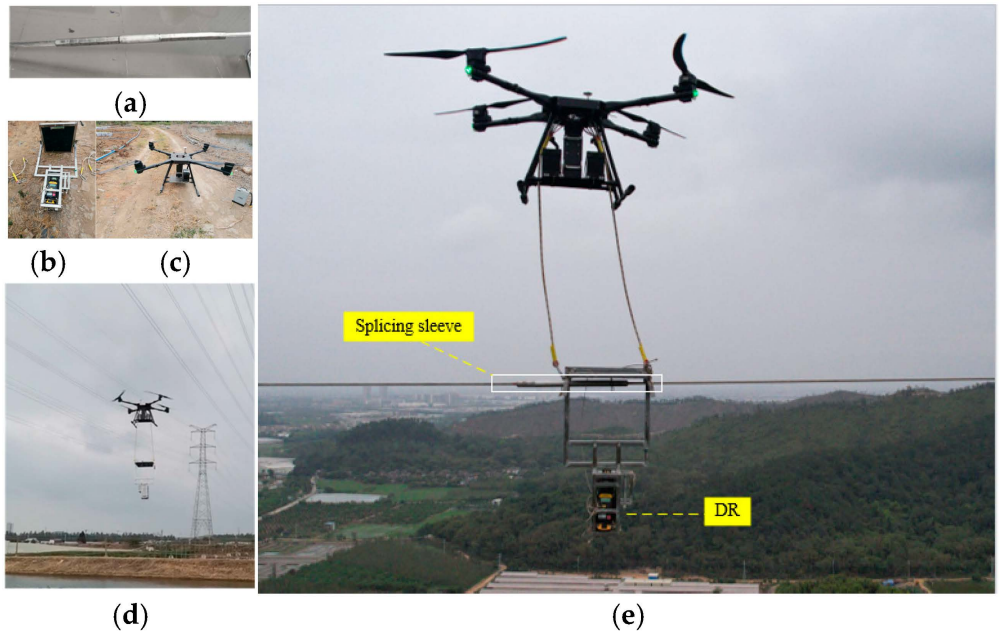


Figure 1. UAV carrying DR equipment approaches and docks with the splicing sleeve on overhead transmission lines. (**a**) Splicing sleeve; (**b**) DR; (**c**) UAV; (**d**) Approaching; (**e**) Docking/Hanging.

Figure 2 shows the aerial imagery captured by the UAV during the approach for recognition and localization of the splicing sleeve. The splicing sleeve shown in the image presents several challenges, such as a variable scale, high aspect ratio, arbitrary orientation, and occupying only a small portion of the available camera pixels.

Selecting the right sensing devices is crucial to ensure UAVs successfully recognize and approach their targets. Traditional systems based on the Global Positioning System (GPS) and Inertial Navigation System (INS) lack the precision required for high-accuracy aerial docking tasks due to their limited accuracy [7,8]. Although ultrasonic sensors [9] and Light Detection and Ranging (LiDAR) [10,11] perform well in obstacle detection and avoidance, they are expensive, bulky, and susceptible to noise and reflection interference in complex environments, necessitating sophisticated compensation controls. In contrast, UAVs equipped with vision sensors and artificial intelligence technology can use stereo vision systems to capture and process environmental information in real time, allowing for both target recognition and enhanced positioning accuracy [12–14]. This technology is gaining increasing attention. Compared with traditional passive stereo vision systems, active stereo vision systems [15–17], which utilize structured light or other active light source technologies, offer higher adaptability and accuracy, particularly under complex lighting and environmental conditions.

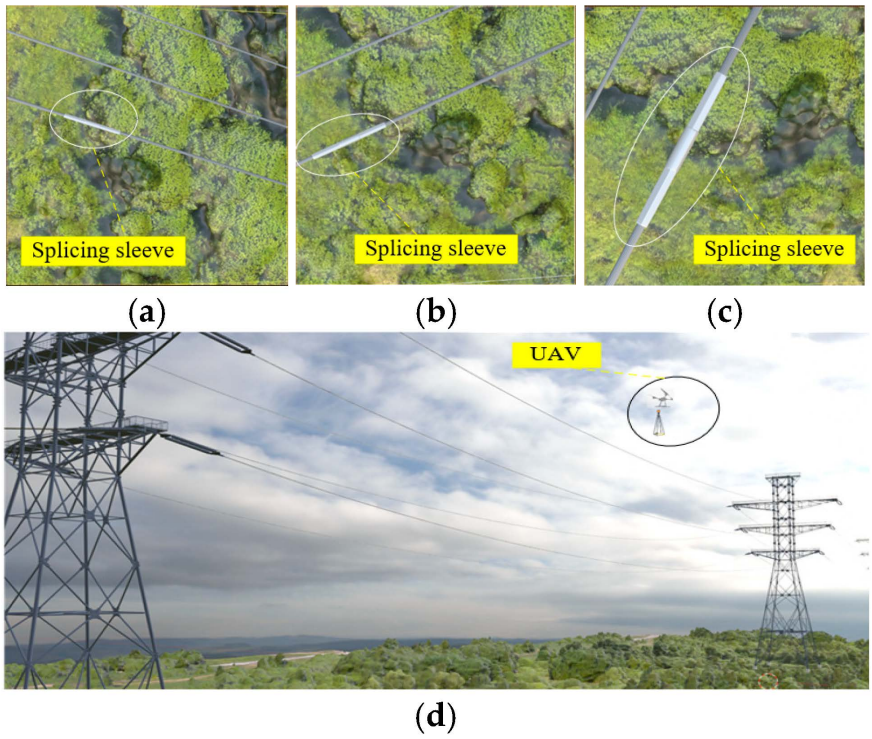


Figure 2. Aerial views of splicing sleeves on overhead transmission lines. (**a**) Distant view; (**b**) Medium-distance view; (**c**) Close-up view; (**d**) Third-person aerial view showing the UAV inspecting the transmission line and splicing sleeves.

Despite significant progress in UAV recognition, localization, and visual navigation research, end-stage visual navigation docking still faces numerous challenges. For example, lighting variations and target diversity in complex environments affect the robustness of visual navigation systems, and different target shapes require the application of varying target detection algorithms. During the UAV's final approach to the splicing sleeve, it is essential to ensure accurate recognition of the sleeve and to guarantee precise localization. In addition, limitations in computational resources and real-time requirements impose higher demands on algorithm efficiency. Therefore, achieving high-precision recognition and localization in the final stage of UAV operation, while improving the real-time performance of the docking process [18,19], remains a key challenge and focus of the current research.

To address these challenges, we propose a UAV autonomous recognition and approach method for linear target splicing sleeves, integrating deep learning and active stereo vision, referred to as UAV-ASS, aimed at solving the key challenges faced by UAVs equipped with DR equipment during the final approach and docking stages with overhead power transmission line splicing sleeves. The main contributions of this study include:

(1) A two-stage rapid and precise localization strategy based on the LC-RB-YOLOv8n (OBB) framework is proposed to address the issues of inaccurate positioning and unstable distance measurement faced by UAVs during high-altitude search, recognition, and approach tasks involving linear target splicing sleeves. This strategy first utilizes reparameterization of training results to obtain a lightweight and fast splicing sleeve recognition model. Subsequently, a local clustering algorithm is employed to enhance the positioning accuracy of splicing sleeves, and finally, the depth values of the splicing sleeves are extracted using the linear nearest neighbor averaging method.

(2) To address the high costs and safety risks associated with image acquisition of splicing sleeves on high-altitude power transmission lines, as well as the difficulties in obtaining sufficient and representative data under complex and variable weather conditions and terrain in real scenarios, the construction of realistic splicing sleeve virtual scenes is proposed. This approach expands the real splicing sleeve dataset

D_{real} to meet the requirements of diverse scenarios, thereby enhancing the robustness and generalization ability of the splicing sleeve recognition model.

(3) To reduce the high-risk nature of visual navigation experiments involving UAV recognition and approach to high-altitude splicing sleeves and to improve algorithm verification and testing efficiency, a UAV-ASS visual simulation platform is proposed. This platform is built using the PX4 open-source UAV flight control system, the ROS, and the physical simulation platform GAZEBO, effectively reducing testing costs and safety risks.

The following is an overview of the key components of this study. Section 2 reviews related research work pertinent to this study. Section 3 systematically presents the principles of the UAV-ASS method based on LC-RB-YOLOv8n(OBB), including the optimization of the rotational object detection model for transmission line splicing sleeves, localization fine-tuning, and waypoint planning. Section 4 provides a comparative experimental analysis to evaluate the effectiveness of the proposed UAV-ASS visual navigation algorithm. Section 5 outlines the conclusions of this study along with suggestions for future work.

2. Related Work

The method of UAV autonomous recognition and approach to linear targets, combining deep learning and active stereo vision, primarily involves the recognition and localization of key targets by UAVs, as well as UAV autonomous docking and landing technology. This section will briefly review the relevant research in these two areas and further elaborate on the uniqueness of this study. The specific contents include:

2.1. UAV Stereo Vision for Key Target Recognition and Localization

At present, UAVs equipped with vision-sensing technology are extensively used for image recognition and localization of power transmission lines and other critical targets. Typically, image processing techniques or deep learning algorithms are employed to extract target features and achieve precise localization. Jia et al. [20] proposed a real-time method for obtaining the distance between a UAV and the corresponding clamp using the YOLOv8n(Det) algorithm in combination with a 3D coordinate detection algorithm based on stereo cameras. This approach provides guidance to ensure the UAV remains in a safe position. Li et al. [21] leveraged RGB-D saliency detection, together with real-time flight data and device parameters, to determine the longitude, latitude, and altitude of insulators. Elsaharti et al. [22] developed macro feature vectors from UAV-captured images in real time and matched them with pre-stored vectors from CAD models, successfully achieving rapid indoor target localization. Daramouskas et al. [23] proposed a UAV-based target detection, tracking, and localization solution using optical cameras with an improved YOLOv4 network for target detection, and combining the positioning information from four UAV cameras. Li et al. [24] applied UAVs for fire object detection. They captured 2D fire images via sensors, computed depth maps with stereo vision, and reduced interference through HSV-Mask filters and a non-zero mean method. GPS and Inertial Measurement Unit (IMU) module data were combined to obtain the latitude, longitude, and altitude coordinates of the fire areas. Li et al. [15] mounted the Intel Realsense D455 depth camera on a UAV and applied deep learning-based object detection algorithms to identify longan fruits, achieving precise localization using RGB-D information. While these studies addressed the problem of 3D position measurements for aerial UAV targets, the measurements of target position, especially the aerial angular orientation, remain largely unexplored. Table 1 compares the differences in UAV applications of stereo vision for target recognition and localization in the related literature.

Table 1. Comparison of Related Studies on UAV Stereo Vision for Target Recognition and Localization.

Study	Target Type	Depth Sensing Method	Algorithm	Key Contribution
Jia et al. [20]	Overhead clamp	Passive stereo vision	YOLOv8n + 3D coordinate detection	Real-time distance measurement between UAV and clamp to ensure safety.
Li et al. [21]	Insulator	RGB-D depth detection	RGB-D saliency detection	Combines real-time flight data to locate insulators' longitude, latitude, and altitude.
Elsaharti et al. [22]	Indoor target	Passive stereo vision + CAD model matching	Real-time macro feature vector matching	Matches features captured by UAV to pre-stored CAD models for rapid indoor localization.
Daramouskas et al. [23]	General targets	Multi-camera localization	Improved YOLOv4 + multi-camera fusion	Introduces multi-camera-based detection for improved precision and tracking.
Li et al. [24]	Fire objects	Passive stereo vision	HSV-Mask filtering + non-zero mean method	Detects fire areas, computes depth, and combines GPS/IMU data for geographical localization.
Li et al. [15]	Longan fruits	Active stereo vision(D455)	MobileNet + YOLOv4	Improves the speed and accuracy of target detection and location for longan picking by UAVs based on vision.
This study	Overhead splicing sleeves	Active stereo vision	LC-RB-YOLOv8n(OBB)	Addresses the problem of rapid recognition and precise distance measurement for UAV docking with linear targets.

2.2. UAV Autonomous Docking and Landing Technology

The challenge of visually guided UAV docking and landing has been a key research focus in this field [25]. Li et al. [26] explored autonomous docking between UAVs and mobile platforms based on Ultra Wideband (UWB) and vision sensors, proposing an integrated estimation and control scheme that is divided into three stages: hovering, approaching, and landing. Yang et al. [27] designed a hybrid system combining UAVs and climbing robots for multi-scale power line inspection. They developed a special feature extraction operator and a density feature recognition algorithm, using stereo vision to measure the depth of the power line landing points, thereby enabling stable autonomous landing of the hybrid UAV inspection robots on power lines. Chen et al. [28] proposed a pan-tilt-based visual servo system, which uses onboard camera status and image data to guide precise UAV landing on a square platform in Global Navigation Satellite System (GNSS)-denied environments, applying different strategies at various landing stages. Zhou et al. [29] applied an improved ant colony algorithm for UAV flight path planning, using deep learning algorithms to identify insulators on overhead transmission lines and locate defects, significantly reducing inspection time and improving efficiency. Although these studies demonstrate the use of visual perception for navigation in different scenarios, current research on UAV visual navigation [25,30] offers relatively little discussion on how to achieve UAV recognition, approach, and suspended docking to linear aerial targets, such as splicing sleeves. Table 2 summarizes the above-mentioned studies.

Table 2. Comparison of Related Studies on UAV Autonomous Docking and Landing Technology.

Study	Target Scenario	Sensor Type	Algorithm	Key Contribution
Li et al. [26]	Docking with moving platforms	UWB + vision sensors	Integrated estimation and control	Proposed a three-stage hovering, approaching, and landing control method for UAV docking.
Yang et al. [27]	Powerline inspection	Stereo vision	Feature extraction + depth measurement	Combined UAV and climbing robots for multi-scale powerline inspection, enabling stable landing.
Chen et al. [28]	GNSS-denied environment	Omnidirectional camera + gimbal system	Image-guided landing control	Guided UAV landing on a square platform in GNSS-denied conditions with stage-specific strategies.
Zhou et al. [29]	Overhead line inspection	Camera + deep learning	Improved ant colony algorithm + defect detection	Introduced UAV path planning and visual inspection, significantly improving inspection efficiency.
This study	Overhead splicing sleeves	Active stereo vision	LC-RB-YOLOv8n(OBB) + virtual simulation	Proposes a deep learning and active stereo vision-based docking method, solving angular measurement challenges.

2.3. Uniqueness of This Study

This study presents a unique approach that integrates deep learning and active stereo vision to enable autonomous recognition and approach of linear aerial targets, such as splicing sleeves, by UAVs. Unlike previous studies that primarily focused on target localization and three-dimensional spatial measurements, this study not only emphasizes precise target localization but also places special emphasis on target angular orientation measurement, which is crucial for reliable docking with linear aerial structures. Additionally, this study expands the dataset using virtual reality technology and establishes the UAV-ASS visual simulation platform to validate the proposed method, providing valuable insights for the application of UAVs in the power industry.

3. Methodology

Figure 3 illustrates the schematic diagram of the proposed UAV-ASS method, which consists of four main modules: (1) a two-stage rapid and accurate localization strategy for rotational targets (LC-RB-YOLOv8n(OBB)); (2) coordinate transformation and waypoints planning; (3) dataset construction incorporating virtual reality; and (4) the UAV-ASS visual navigation simulation platform. Initially, the LC-RB-YOLOv8n(OBB) module deployed on the UAV reads aligned RGB and depth images of the splicing sleeve from the stereo depth camera in real-time, outputting the finely adjusted rotational target position of the splicing sleeve (including rotation angle θ_F and position parameters x_{OF}, y_{OF}, w_F, h_F) and the distance between the UAV and the splicing sleeve, $D_{\text{UAV-SS}}$. Next, the pixel coordinates of the splicing sleeve are transformed into UAV body coordinates, and flight waypoints are generated through route planning, enabling the UAV to autonomously approach and dock with the splicing sleeve. This study also constructs a dataset adapted to diverse application scenarios using virtual reality technology. Additionally, a UAV-ASS visual navigation simulation platform integrating the ROS, the open-source UAV autopilot PX4, and the GAZEBO robotics simulation system is independently developed to validate the proposed UAV-ASS method.

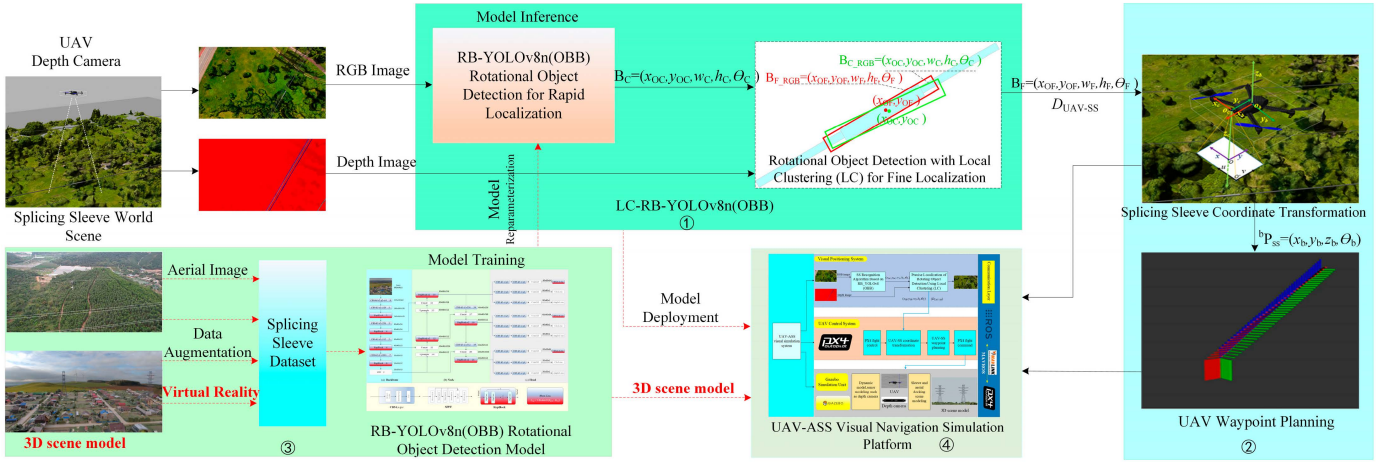


Figure 3. Block Diagram of UAV-ASS Method.

3.1. A Two-Stage Rapid and Accurate Localization Strategy for Rotational Targets (LC-RB-YOLOv8n(OBB))

3.1.1. Rapid Localization of Rotational Object Detection Using RB-YOLOv8n(OBB)

The target detection algorithm used for UAV approaches to splicing sleeves must be able to handle multi-scale targets and provide high-precision angle measurements, and at the same time, meet the real-time processing requirements for high-frame-rate image transmission during high-altitude, high-speed UAV operations. This study presents a lightweight, end-to-end rotational object detection model, RB-YOLOv8n(OBB) to address these requirements. The model offers excellent real-time performance. The model is based on the YOLOv8n(OBB) [31] rotational object detection module, which has demonstrated outstanding performance on the remote sensing dataset DOTA [32], and has been further refined and optimized for the unique challenges of this application.

Figure 4 illustrates the network architecture diagram of the RB-YOLOv8n(OBB) model proposed in this study. The network consists of three main components: the Backbone, Neck, and Head. The Backbone includes three modules—CBS, RepBlock, and SPPF -with nine feature output layers. The structures of the CBS and SPPF modules are shown in the Figure 4. The CBS module is used for feature extraction, normalization, and non-linear processing, while the SPPF module is for pooling and concatenation of feature maps at different scales. The RepBlock module replaces the C2f module in the classical YOLOv8n(OBB) backbone, with its core function being to obtain an efficient inference network through reparameterization of the trained network. This facilitates deployment on edge hardware, enabling lightweight, real-time processing. The feature maps from the 4th, 6th, and 9th layers of the Backbone are fed into the Neck.

The Neck consists of three modules: RepBlock-s2, Concat, and Upsample. The RepBlock-s2 module replaces the C2f-s2 module in the classical YOLOv8n(OBB) Neck, following the same enhancement strategy used in the Backbone. The Concat and Upsample modules achieve bottom-up PAN (Path Aggregation Network) and top-down FPN (Feature Pyramid Network) feature fusion, enhancing the multi-scale feature fusion capabilities. Finally, the feature maps from the 15th, 18th, and 21st layers of the Neck are passed to the Head.

The Head adopts a decoupled structure, generating feature maps for the bounding box, rotation angle, and classification loss. The feature map sizes for predicting bounding boxes are $80 \times 80 \times 64$, $40 \times 40 \times 64$, and $20 \times 20 \times 64$; for predicting rotation angles, the feature map sizes are $80 \times 80 \times 1$, $40 \times 40 \times 1$, and $20 \times 20 \times 1$. Since the rotational object detection task focuses solely on splicing sleeves, the classification feature map has only one channel. To calculate the bounding box regression loss L_{reg} in RB-YOLOv8n(OBB), the Gaussian probabilistic ProbIoU is used to better capture the overlap and uncertainty of the bounding boxes.

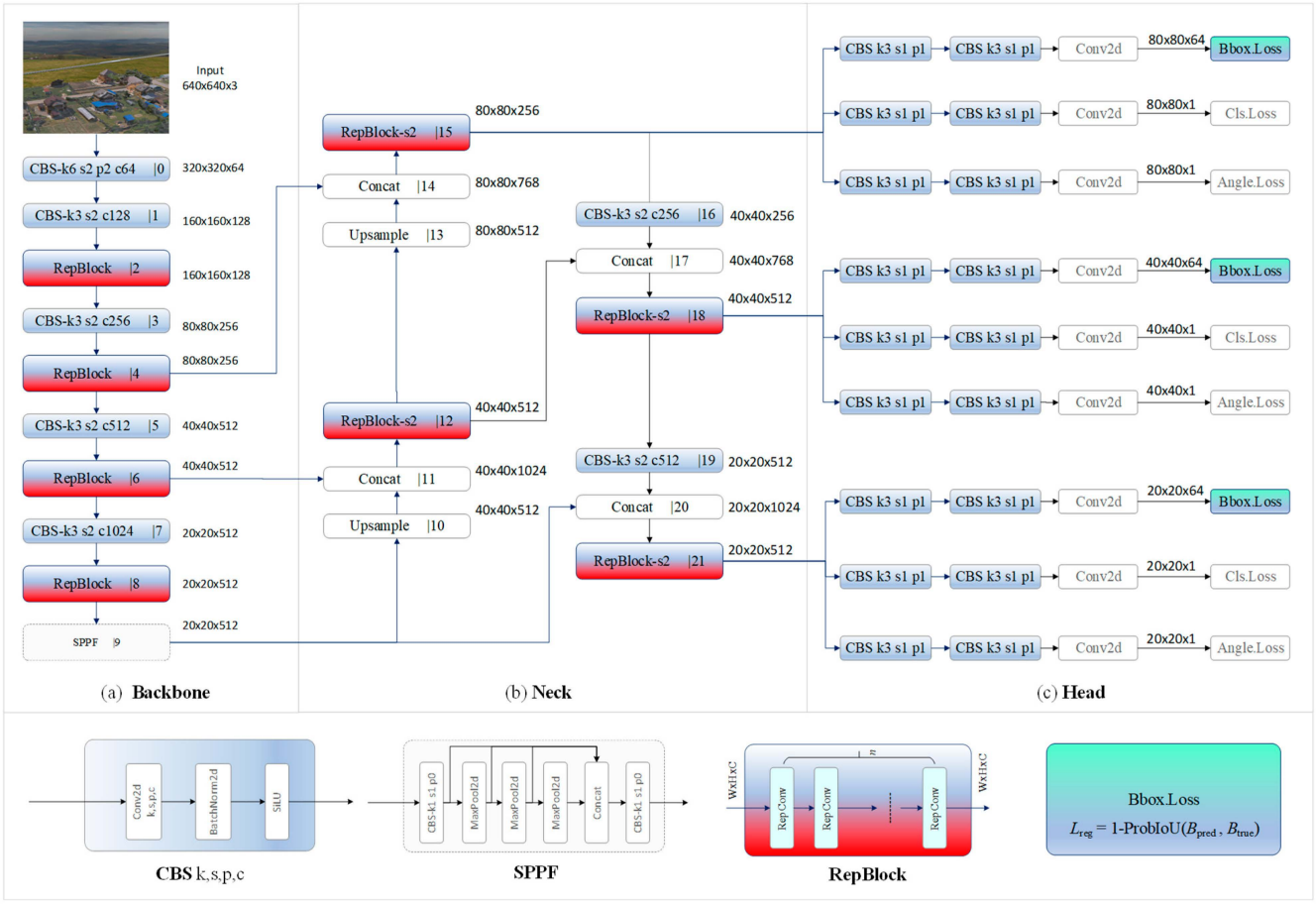


Figure 4. The network architecture diagram of the RB-YOLOv8(OBB) model.

The following section focuses on introducing the principles and advantages of replacing the C2f module in the YOLOv8n(OBB) with the RepBlock module. Figure 5a illustrates the structure of the original C2f module in the YOLOv8n (OBB) backbone network. This module consists of 2 CBS layers and n Bottleneck modules, connected through Split and Concat operations. The inference network shares the same structure as the training network. To further enhance inference speed and reduce the computational complexity of the model, the RepBlock module was introduced. Through reparameterization, the multi-branch structure of RepConv (Training State) in the training phase is transformed into a single-branch structure RepConv (Inference State), as shown in Figure 5b. Although the total parameter count of the RepBlock module increases during training, resulting in extended training time, it significantly improves inference speed while maintaining detection accuracy during the inference phase.

In the training phase, RepConv consists of three branches: 3 × 3 conv, 1 × 1 conv, and an Identity branch, where the Identity branch performs no operations. Each branch undergoes BatchNorm2d standardization, after which the outputs are summed up and passed through the activation function to produce Y_{train}. The input feature map is set to be X, the activation function is set to be SiLU, and the normalization function BN. Then we have [33]:

$$Y_{\text{train}} = \text{SiLU}(\text{BN}(\text{Conv}_{3\times3}(X) + \text{BN}(\text{Conv}_{1\times1}(X) + \text{BN}(\text{Identity}(X))) \quad (1)$$

The reparameterization process in this study fuses the convolutional layer (Conv) and batch normalization (BN) layer of each branch into an equivalent convolutional layer (Conv_{eq}), which are then stacked to form the final convolutional layer in the inference phase, as shown in Figure 5. Assuming that after training, the weight $W^{(n)}$ of the Conv

layer in a certain branch of RepConv (training state) is known, and the BN layer parameters include the running mean $\mu^{(n)}$, running variance $\sigma^{2(n)}$, scaling factor $\gamma^{(n)}$, bias $\beta^{(n)}$, and a small constant $\varepsilon^{(n)}$, with the standard deviation given by $std^{(n)} = \sqrt{\sigma^{2(n)} + \varepsilon^{(n)}}$, After adjustment by the BN layer, the $Conv_{eq}$ weight $W'^{(n)}$ and bias $b'^{(n)}$ for this branch can be calculated as follows [33]:

$$W'^{(n)} = W^{(n)} \cdot \frac{\gamma^{(n)}}{std^{(n)}} = W^{(n)} \cdot \frac{\gamma^{(n)}}{\sqrt{\sigma^{2(n)} + \varepsilon^{(n)}}}; \; b'^{(n)} = \beta^{(n)} - \frac{\mu^{(n)}\gamma^{(n)}}{std^{(n)}} = \beta^{(n)} - \frac{\mu^{(n)}\gamma^{(n)}}{\sqrt{\sigma^{2(n)} + \varepsilon^{(n)}}} \quad (2)$$

In Equation (2), (n) is used as an index with values of (3), (1), and (0), representing the parameters of the 3 × 3 conv, 1 × 1 conv, and Identity branch, respectively.

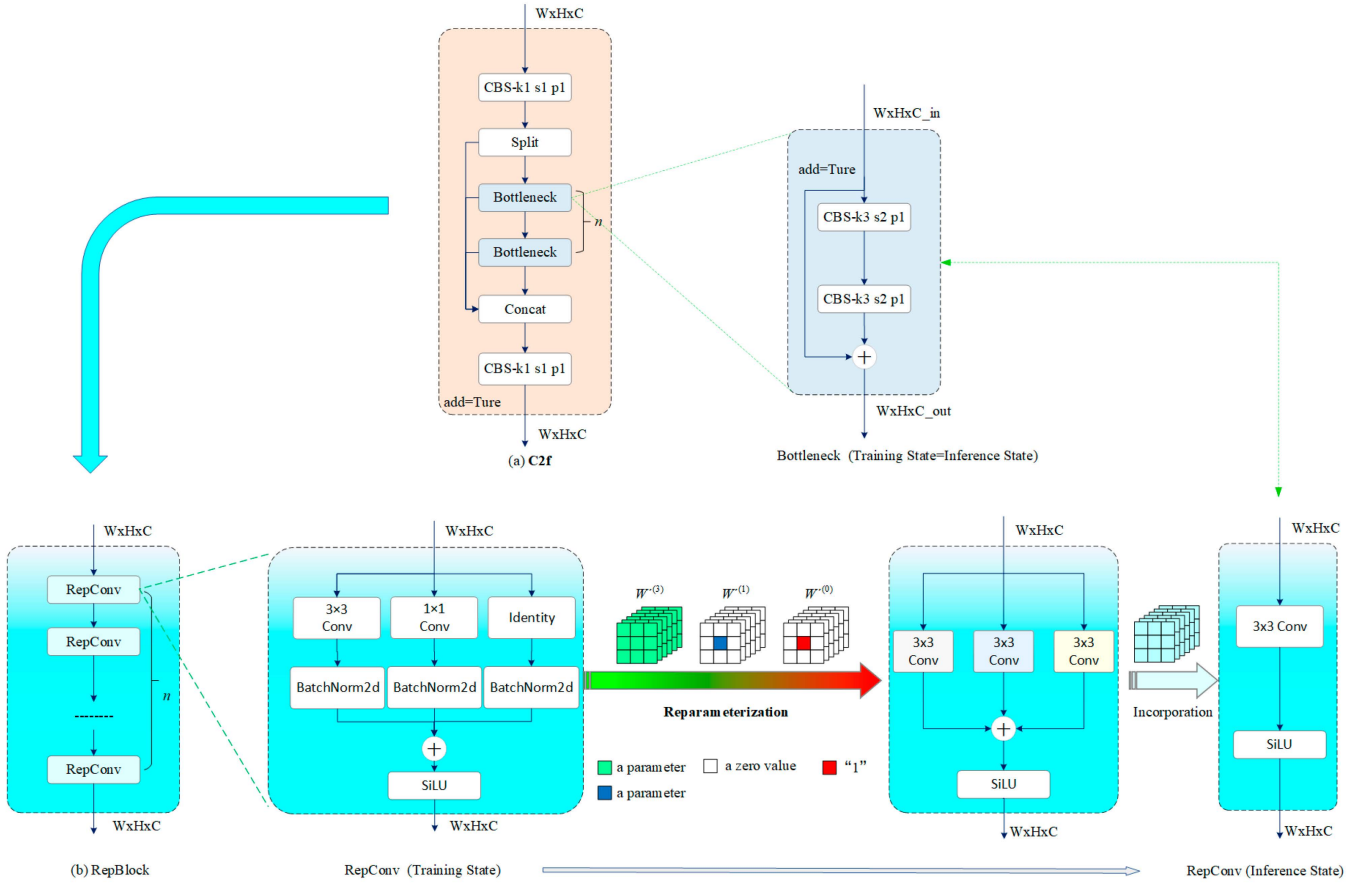


Figure 5. Structure and Reparameterization Process of the RepBlock Module.

The $Conv_{eq}$ weights and biases for each branch in the RepConv structure during the inference state were determined as follows:

(1) In the $Conv_{eq}$ calculation for the 3 × 3 conv branch, since the weight $W^{(3)}$ obtained during training has the same dimensions as the weight in the inference phase, the equivalent weight $W'^{(3)}$ and bias $b'^{(3)}$ for the 3 × 3 conv can be directly calculated by substituting the trained values of $W^{(3)}$, $\mu^{(3)}$, $\sigma^{2(3)}$, $\gamma^{(3)}$, $\beta^{(3)}$ and $\varepsilon^{(3)}$ into Equation (2).

(2) In the $Conv_{eq}$ for the 1 × 1 conv branch, since the weight $W^{(1)}$ obtained during training does not have the same dimensions as the weight in the inference phase, it can be expanded using a padding operation (*Pad*) to form $W^{(1)}{}_{3\times3}$, ensuring that $W^{(1)}{}_{3\times3}$ matches the dimensions of the weight in the inference phase. The expanded $W^{(1)}{}_{3\times3}$ is denoted as:

$$W_{3\times3}^{(1)} = Pad\left(W^{(1)}\right) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & W^{(1)} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3}$$

At this point, the equivalent weight $W'^{(1)}$ and bias $b'^{(1)}$ for the 1×1 conv branch can be calculated by substituting $W^{(1)}{}_{3\times3}$, $\mu^{(1)}$, $\sigma^{2(1)}$, $\gamma^{(1)}$, $\beta^{(1)}$ and $\varepsilon^{(1)}$ into Equation (2).

(3) In the Conv_{eq} calculation for the Identity branch, since this branch only contains the BN layer, a 3×3 pseudo convolutional kernel is first created, with the center of the kernel set to 1 and the remaining elements set to zero. The number of convolutional kernels is made equal to the number of input channels. The pseudo weight for this branch, $W^{(0)}$, is denoted as:

$$W^{(0)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4}$$

At this point, the equivalent weight $W'^{(0)}$ and bias $b'^{(0)}$ for the Identity branch can be calculated by substituting $W^{(0)}$, $\mu^{(0)}$, $\sigma^{2(0)}$, $\gamma^{(0)}$, $\beta^{(0)}$ and $\varepsilon^{(0)}$ into Equation (2).

Therefore, the expression for the output $Y_{inference}$ of a single branch in the RepConv during the inference phase is given by:

$$Y_{inference} = \text{SiLU}\left(\text{Conv}_{3\times3}\left(X, W_{eq}, b_{eq}\right)\right) = \text{SiLU}\left(\text{Conv}_{3\times3}\left(X, W'^{(3)} + W'^{(1)} + W'^{(0)}, b\prime^{(3)} + b\prime^{(1)} + b\prime^{(0)}\right)\right) \tag{5}$$

The reparameterization separates the inference and training processes, with the model's computations shown in Equations (1) and (5), respectively. Compared to the RepConv structure in the training phase, the RepConv structure during inference is significantly simplified, dramatically reducing computational complexity and facilitating hardware implementation. The accuracy and real-time performance of this structure are further evaluated in subsequent experiments.

The splicing sleeve's rotated bounding box is generated using RB-YOLOv8n(OBB). Figure 6 illustrates the rotated bounding box output from the rapid localization, represented as $B_{C_RGB} = (x_{OC}, y_{OC}, w_C, h_C, \theta_C)$, where x_{OC} and y_{OC} are the horizontal and vertical coordinates of the center point O of the rotated bounding box, respectively, and w_C, h_C and θ_C represent the width, height, and rotation angle of the bounding box, respectively.

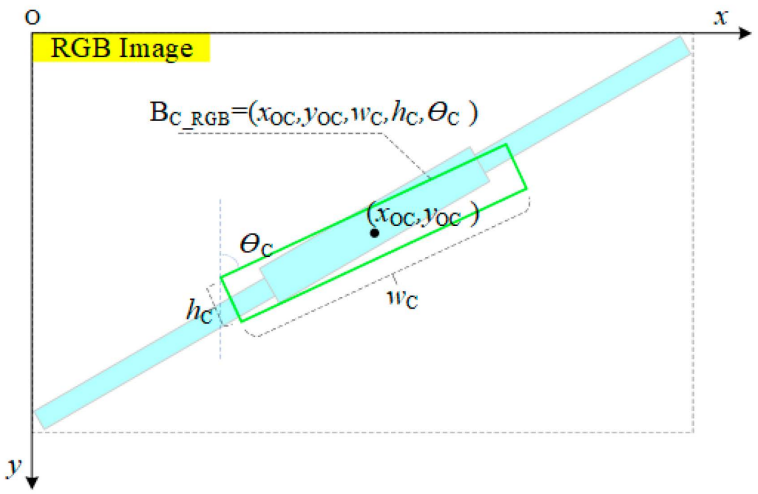


Figure 6. Schematic Diagram of the Rotated Bounding Box Output for the Splicing Sleeve from Rapid Localization.

3.1.2. Fine Localization of Rotational Object Detection Using Local Clustering (LC)

When the UAV equipped with a depth camera performs high-altitude search and identification tasks for splicing sleeves, characteristics such as the high aspect ratio, arbitrary angle, and small pixel ratio of the linear target splicing sleeves, along with UAV body vibrations, make $B_{C_RGB} = (x_{OC}, y_{OC}, w_C, h_C, \theta_C)$ insufficient for accurately and reliably representing the position of the splicing sleeve, as shown in Figure 6.

Figure 7 illustrates the schematic diagrams of the fine localization process using local clustering. First, the $B_{C_RGB} = (x_{OC}, y_{OC}, w_C, h_C, \theta_C)$ bounding box is mapped to identify the local region R_{DE} in the depth image that contains the rapidly localized splicing sleeve, thereby reducing the computational area. Next, the depth values D_R of the local region R_{DE} are extracted, and clustering and fitting are performed on D_R to obtain a more accurately localized bounding box $B_{R_Depth} = (x', y', w', h', \theta')$, enhancing the precision of the localization. Finally, through coordinate transformation, the value of $B_{R_Depth} = (x', y', w', h', \theta')$ in the depth image is mapped to $B_{F_RGB} = (x_{OF}, y_{OF}, w_F, h_F, \theta_F)$ in the RGB image, completing the transition from rapid localization to fine localization of the splicing sleeve.

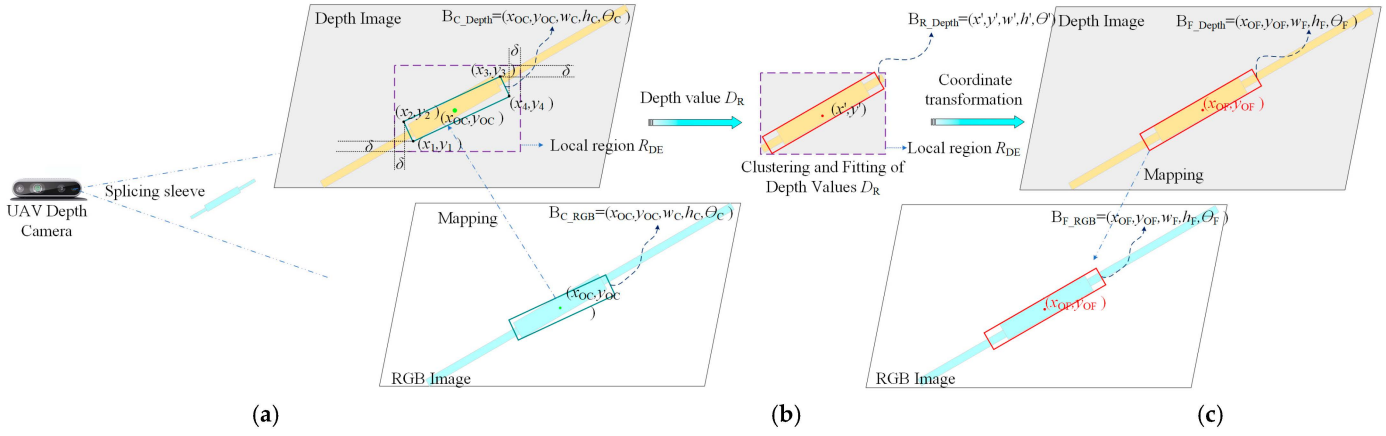


Figure 7. Diagram of the Fine Localization Principle for Rotational Object Detection Using LC. (**a**) Boundary calculation of the coarsely localized rectangular box for the splicing sleeve in the local region R_{DE}; (**b**) Boundary calculation of the coarsely localized rectangular box for the splicing sleeve in the local region R_{DE}, and clustering and fitting of the depth values D_R in region R_{DE}; (**c**) Fine localization of the splicing sleeve's rotated bounding box B_{F_RGB}.

The boundary of the local region R_{DE}, which contains the rapidly localized rectangular box of the splicing sleeve, and the clustering and fitting process of the depth values D_R within region R_{DE}, are determined as follows:

(1) Determination of the Local Region boundary R_{DE} for the Rapidly Localized Rectangular Box of the Splicing Sleeve:

The rapid localization result of the splicing sleeve in the RGB image, $B_{C_RGB} = (x_{OC}, y_{OC}, w_C, h_C, \theta_C)$, obtained by RB-YOLOv8n(OBB), is mapped to the depth image as $B_{C_DE} = (x_{OC}, y_{OC}, w_C, h_C, \theta_C)$. The boundary of the local region R_{DE} must fully enclose the splicing sleeve's rectangular box B_{C_DE}, with an additional margin for fine adjustment of B_{C_DE}, as shown in Figure 7a.

The four corner points of the splicing sleeve's rectangular box B_{C_DE} are set to be (x_i, y_i), where $i \in \{1, 2, 3, 4\}$. The minimum horizontal and vertical coordinates of these corner points are defined as $x_{min} = \text{Min}(x_1, x_2, x_3, x_4)$ and $y_{min} = \text{Min}(y_1, y_2, y_3, y_4)$, respectively; the maximum horizontal and vertical coordinates are defined as $x_{max} = \text{Max}(x_1, x_2, x_3, x_4)$ and $y_{max} = \text{Max}(y_1, y_2, y_3, y_4)$, respectively. The smallest region for R_{DE} is then given by:

$$R_{DE_min} = \left\{ (x, y) \in \mathbb{R}^2 \middle| x_{min} \le x \le x_{max}, y_{min} \le y \le y_{max} \right\} \quad (6)$$

With the assumption that the margin for fine-tuning the localization is δ in both the horizontal and vertical directions, the expression for the local region R_{DE} is given by:

$$R_{\mathrm{DE}} = \left\{ (x, y) \in \mathbb{R}^2 \middle| x_{\min} - \delta \le x \le x_{\max} + \delta, y_{\min} - \delta \le y \le y_{\max} + \delta \right\} \tag{7}$$

The value of δ is determined based on the principle of sufficiency, with smaller values being preferable. Once the boundary calculation for the local region R_{DE} is completed, subsequent operations, such as depth value clustering and fitting are executed within the R_{DE}.

(2) Clustering and Fitting of Depth Values D_{R} in the Local Region R_{DE}

Due to the presence of the splicing sleeve, the depth values in the local region R_{DE} exhibit different distributions. By calculating the depth values D_R in this region via clustering and fitting, a more accurate representation of the splicing sleeve's position, $\mathrm{B}_{\mathrm{R_DE}} = (x', y', w', h', \theta')$, is obtained. For clustering D_R, the fast and convenient K-Means algorithm could be used, and the fitting of the minimum area rotated bounding box is achieved using the cv2.minAreaRect() function from the OpenCV library, as shown in Figure 7b.

① Clustering of Depth Values D_{R} in R_{DE}

First, the depth values $D_R = \{D(x_i, y_i) | (x_i, y_i) \in R_{DE}\}$ are extracted, and then the K-Means algorithm to cluster D_R is applied. The number of clusters is k, the cluster centers are μ_i, the DR samples are d_n, and the indicator variable is r_{ni}. The objective is to minimize the cost function J, which is expressed as follows [34]:

$$J_{\min} = \sum_{n=1}^{N} \sum_{i=1}^{K} r_{ni} \| d_n - \mu_i \|^2; \ where \ r_{ni} = \begin{cases} 1, \ d_n \in cluster \ i \\ 0, \ d_n \notin cluster \ i \end{cases} \tag{8}$$

To solve this equation, the clustering centers μ_i for D_R are first initialized, and then the samples d_n are assigned to their respective clusters. Next, the clusters are updated based on $\mu_i = \sum_{n=1}^{N} r_{ni} d_n / \sum_{n=1}^{N} r_{ni}$, and the centers μ_i for each cluster are recalculated. This process continues until the cluster centers no longer show significant changes, at which point the loss function J is considered to have reached its minimum value $J_{\min}$. After clustering the D_R values, the cluster to which the depth value D ($x' = x_{\mathrm{OC}} - x_{\min} + \delta$, $y' = y_{\mathrm{OC}} - y_{\min} + \delta$) belongs is identified as the splicing sleeve category.

② Depth Value D_{R} Clustering and Fitting

The point set for the depth clustering category corresponding to the splicing sleeve is defined as $P = \{(x_i, y_i)\}_{i=1}^{N}$. The cv2.minAreaRect() function from the OpenCV library is used to fit a minimum area of the rotated bounding box for the point set P, resulting in the rotated bounding box $\mathrm{B}_{\mathrm{R_DE}}$ in the local region R_{DE} of the depth map, as follows:

$$\mathrm{B}_{\mathrm{R_DE}} = \mathrm{cv2.minAreaRect}(P) = (x', y', w', h', \theta') \tag{9}$$

The rotated bounding box $\mathrm{B}_{\mathrm{R_DE}}$ in the local region R_{DE} of the depth map cannot be used directly. It must first be transformed into the rotated bounding box $\mathrm{B}_{\mathrm{F_RGB}}$ on the RGB image plane, which represents the final result of fine localization after applying LC to the rotational object (see Figure 7c). The expression is given as:

$$B_{\mathrm{F_RGB}} = (x_{\mathrm{OF}}, y_{\mathrm{OF}}, w_{\mathrm{F}}, h_{\mathrm{F}}, \theta_{\mathrm{F}}) = (x' + x_{\min} - \delta, y' + y_{\min} - \delta, w', h', \theta') \tag{10}$$

After $\mathrm{B}_{\mathrm{F_RGB}} = (x_{\mathrm{OF}}, y_{\mathrm{OF}}, w_{\mathrm{F}}, h_{\mathrm{F}}, \theta_{\mathrm{F}})$ is obtained, the UAV's relative distance to the splicing sleeve $D_{\mathrm{UAV\text{-}SS}}$ is determined from the depth map. $D(x_i, y_i)$ is set to represent the depth values along the line $(y - y_{\mathrm{OF}}) = tan(\theta_{\mathrm{F}})\ (x - x_{\mathrm{OF}})$ in the depth map, within the range of $[-\sigma/2, +\sigma/2]$ pixels, where $\sigma = w_{\mathrm{F}}/10$. After removing outliers caused by factors such as the smooth surface of the splicing sleeve, texture loss, and lighting variations, assuming there are N valid depth points, and with θ_{F} being the rotation angle of the bounding box,

as shown in Figure 8, the $D_{\text{UAV-SS}}$ is calculated using the linear nearest neighbor averaging method as:

$$D_{\text{UAV-SS}} = \frac{1}{N} \sum_{x_i = x_{\text{OF}} - \frac{\sigma}{2}\cos\theta_{\text{F}}}^{x_i = x_{\text{OF}} + \frac{\sigma}{2}\cos\theta_{\text{F}}} D(x_i, \tan\theta_F \times (x_i - x_{\text{OF}}) + y_{\text{OF}});\ where\ D(x_i, \tan\theta_F \times (x_i - x_{\text{OF}}) + y_{\text{OF}}) \neq null \quad (11)$$

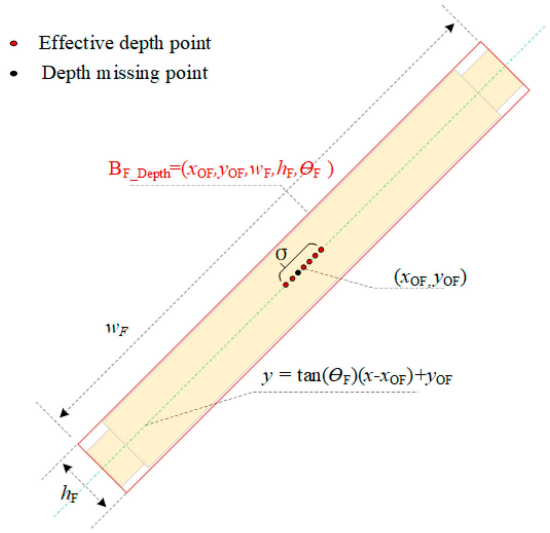


Figure 8. Rules for obtaining $D_{\text{UAV-SS}}$.

Real-time high-precision recognition and distance measurement of the splicing sleeve by the UAV at high altitudes are achieved following the above steps. Algorithm 1 provides the pseudocode for the above process.

Algorithm 1 Fusion K-Means clustering rotating target fine-tuning algorithm

Input:depth_image, rgb_image
Output:rect_new(Fine-tuning rotating target recognition results), depth_value

```
k_means_trimming_functions(rag_image, depth_image)
    result = inference_detector(model, rgb_image); // Rotating target recognition results
key_point = extract_clustering_target_regions(result); // Return the key parameters of the horizontal box area of the rotating target result according to the result.
    if key_point! = 0: // Determine whether there are recognition results that satisfy the conditions
        depth_target_regions = depth_image(key_point); // Cut horizontal box area on depth_image
num_clusters = 3; // Set the initial clustering value. When shooting from high altitude, determine the clustering value based on the complexity of the background.
        cluster_depth_image = K_Means_fuction(num_clusters,depth_target_regions);
// Return clustering results
        min_depth_class = np.argmin([np.mean(depth_target_regions[cluster_depth_image == i]) for i in range(num_clusters)]); // Finding the Depth Minimum Category
        rect = cv2.minAreaRect(points_cv); // Fitting a minimum area rotating frame
       center_x, center_y, width, heigth, angle = rect; // Get the center point coordinates, width, height and rotation angle of the rotated box after fine-tuning.
        depth_value = depth_target_regions[int(center_y), int(center_x)];
// Extract the depth value of the center point of the rotating frame
        rect_new = rect + key_points; // Adjust the spinning frame coordinates to depth_image coordinates
        return (rect_new, depth_value); // Returns the coordinates and depth of the original rotated frame.
    else:
        return (0,0); // Returns the null value
```

3.2. UAV-ASS Coordinate Transformation and Waypoint Planning

The calculation of the splicing sleeve's center point relative to the UAV body coordinates involves four coordinate systems, as shown in Figure 9: the pixel coordinate system O_{xy}, the image coordinate system O_{p_xy}, the camera coordinate system O_{c_xyz}, and the UAV body coordinate system O_{b_xyz}.

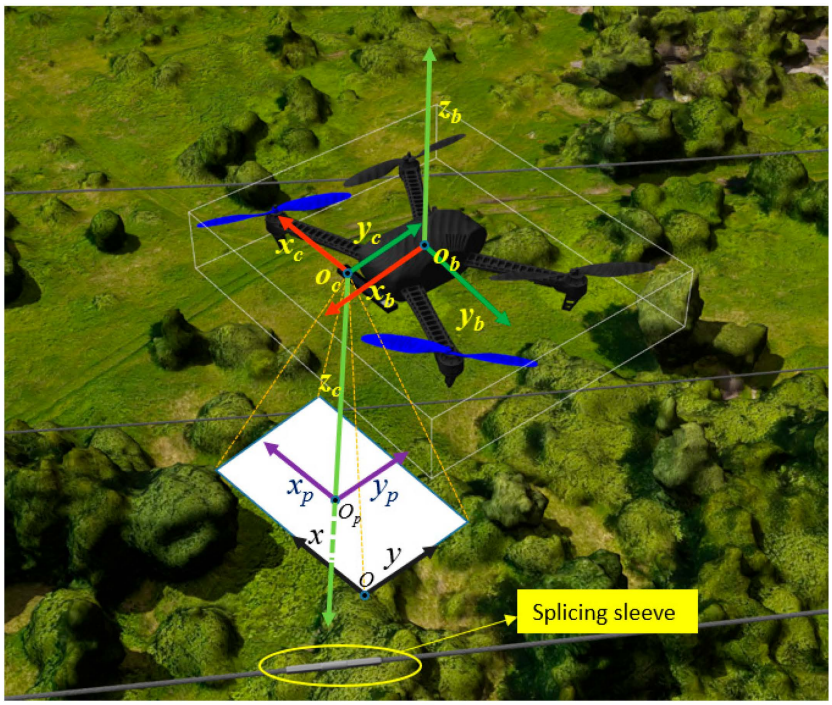


Figure 9. UAV-ASS Coordinate system.

The transformation matrices between the pixel coordinate system O_{xy} and the image coordinate system O_{p_xy}, the image coordinate system O_{p_xy} and the camera coordinate system O_{c_xyz}, and the camera coordinate system O_{c_xyz} and the UAV body coordinate system O_{b_xyz} are denoted as $\mathbf{K_1}$, $\mathbf{K_2}$, and $\mathbf{K_3}$, respectively. The origin O_p of the image coordinate system O_{p_xy} has coordinates (x_{op}, y_{op}) in the pixel coordinate system O_{p_xy}. The pixel sizes along the horizontal axis x_p and vertical axis y_p of the image coordinate system O_{p_xy} are $\mathrm{d}x_p$ and $\mathrm{d}y_p$, respectively, and the camera focal length is f, where $\mathrm{d}x$, $\mathrm{d}y$, and f are the intrinsic parameters of the camera. The rotation and translation matrices from the camera coordinate system O_{c_xyz} to the UAV body coordinate system O_{b_xyz} are ${}^{b}\mathbf{R}_{c3\times3}$ and ${}^{b}\mathbf{T}_{c3\times1}$, respectively, and are primarily determined by the camera's mounting position on the UAV. Thus, $\mathbf{K_1}$, $\mathbf{K_2}$, and $\mathbf{K_3}$ are expressed as:

$$\mathbf{K}_1 = \begin{bmatrix} 1/\mathrm{d}x_p & 0 & x_{op} \\ 0 & 1/\mathrm{d}y_p & y_{op} \\ 0 & 0 & 1 \end{bmatrix}^{-1} ; \mathbf{K}_2 = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \end{bmatrix}^{-1} ; \mathbf{K_3} = \begin{bmatrix} {}^{b}\mathbf{R}_{c3\times3} & {}^{b}\mathbf{T}_{c3\times1} \\ \vec{0} & 1 \end{bmatrix}^{-1} \tag{12}$$

According to Equations (10) and (11), $(x_{\mathrm{OF}}, y_{\mathrm{OF}})$ and $D_{\text{UAV-SS}}$ are obtained, and the relationship of the splicing sleeve's center point relative to the UAV body coordinates (x_b, y_b, z_b) is expressed as follows:

$$\begin{bmatrix} x_b & y_b & z_b & 1 \end{bmatrix}^{\mathrm{T}} = D_{\mathrm{UAV-SS}} \mathbf{K_3} \mathbf{K_2} \mathbf{K_1} \begin{bmatrix} x_{OF} & y_{OF} & 1 \end{bmatrix}^{\mathrm{T}} \tag{13}$$

The splicing sleeve is typically in a horizontal position while in service. When the UAV hovers at high altitude and identifies the splicing sleeve, the position of the splicing sleeve relative to the UAV can be simplified as ${}^{\mathrm{b}}\mathrm{P}_{\mathrm{ss}} = (x_b, y_b, z_b, \theta_b)$, where θ_b is the angle

between the longitudinal axis of the splicing sleeve and the UAV's body axis (i.e., the angle θ_F) between the long side of the splicing sleeve and the *y*-axis of the pixel coordinate system). The criteria for successful docking are that the UAV hovers 1.0 m above the splicing sleeve (this value depends on camera imaging parameters and the length of the equipment suspension line) and that $\theta_b = 0$, as shown in Figure 10. This indicates that the splicing sleeve is centered in the camera's image and that its longitudinal axis is parallel to the *y-axis* of the pixel coordinate system. The final position of the splicing sleeve relative to the UAV can be expressed as ${}^{b}P_{ss}|_{end} = (0, 0, -1.0, 0)$.

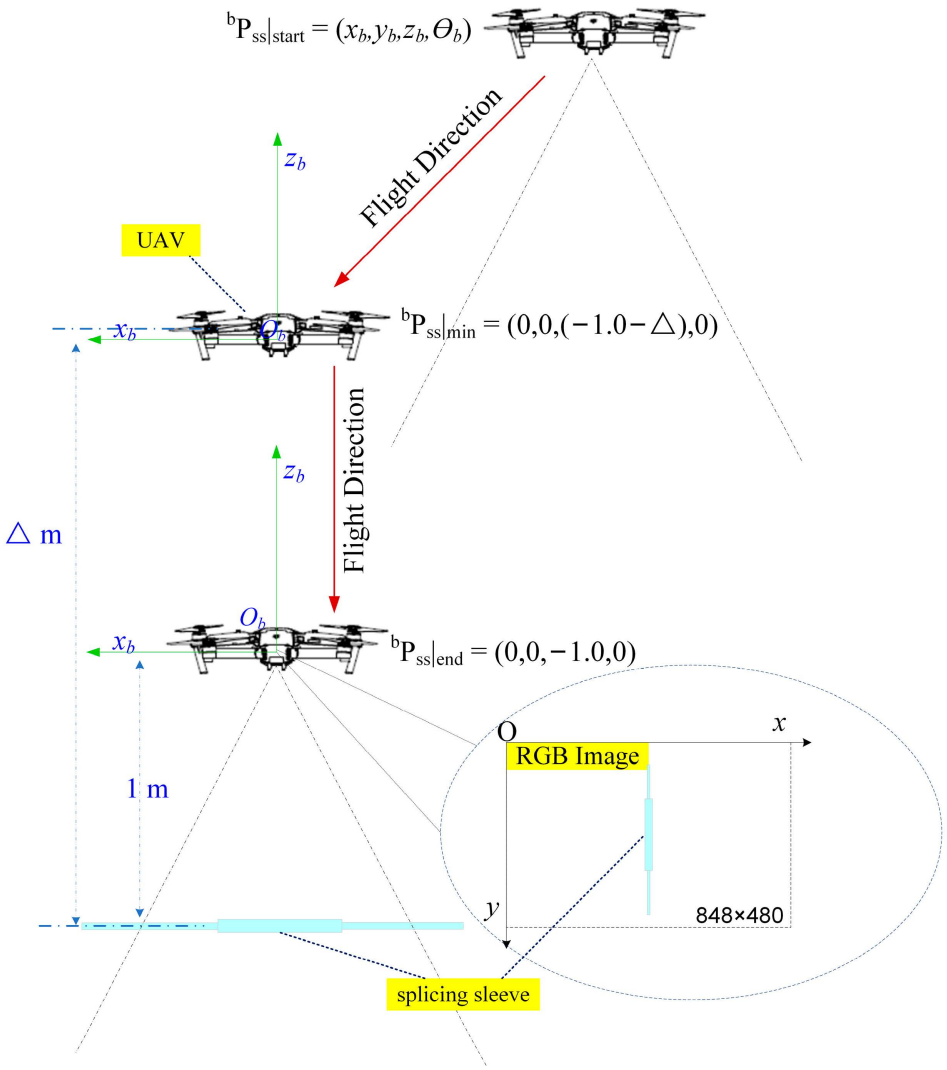


Figure 10. Schematic Diagram of UAV Approaching the Splicing Sleeve.

To improve docking success rates, an intermediate waypoint ${}^{b}P_{ss}|_{mid} = (0, 0, (-1.0 - \Delta, 0))$ is inserted between the starting point ${}^{b}P_{ss}|_{start}$ and the end point ${}^{b}P_{ss}|_{end}$, where Δ can be adjusted as needed. First, the UAV moves from ${}^{b}P_{ss}|_{start}$ to ${}^{b}P_{ss}|_{mid}$, during which both position and orientation must be adjusted, ensuring that θ_b is achieved upon reaching ${}^{b}P_{ss}|_{mid}$. Next, the UAV moves from ${}^{b}P_{ss}|_{mid}$ to ${}^{b}P_{ss}|_{end}$, which only involves adjusting the UAV's altitude. These two steps generate flight waypoints using linear interpolation of position and orientation, thus enhancing flight smoothness. The UAV's position relative to the world coordinate system is set to be ${}^{w}P_{b}$, and its orientation relative to the world coordinate system be ${}^{w}q_{b}$. The angle between the initial orientation ${}^{w}q_{b}|_{start}$ and the final orientation ${}^{w}q_{b}|_{end}$ is θ, with $t \in [0, 1]$. The UAV's position at any waypoint is calculated by the following equation:

$$^{w}P_{b}(t) = {}^{w}P_{b|start}(1-t) + {}^{w}P_{b|mid}t\ ;\ {}^{w}q_{b}(t) = \frac{\sin((1-t)\theta)}{\sin\theta}{}^{w}q_{b|start} + \frac{\sin(t\theta)}{\sin\theta}{}^{w}q_{b|end} \quad (14)$$

4. Experimental and Results Analysis

This experiment aims to validate the feasibility and effectiveness of a UAV autonomous recognition and approach method for linear target splicing sleeves, integrating deep learning and active stereo vision. The main components include experiments on UAV recognition and localization of splicing sleeves on overhead transmission lines, as well as a series of UAV-ASS visual simulation experiments.

4.1. UAV Recognition and Localization Experiments for Splicing Sleeves on Overhead Transmission Lines

The UAV recognition and localization experiments for splicing sleeves on overhead transmission lines include the construction of the experimental dataset D_{SS}, the RB-YOLOv8n(OBB) rotational object detection rapid localization, as well as the LC rotational object detection fine localization.

4.1.1. Construction of the Experimental Dataset D_{SS}

In UAV inspection tasks, traditional data collection methods require substantial human and material resources, especially when capturing images of splicing sleeves installed on high-altitude transmission lines, which entails high costs and safety risks. Moreover, the complex and variable weather conditions and terrain make it extremely challenging to acquire sufficient and representative data in real-world scenarios [35]. To address the issue of limited samples in the real splicing sleeve dataset D_{real}, this study combines the real dataset D_{real}, a virtual reality scene dataset D_{vr}, and data augmentation techniques to construct a splicing sleeve dataset capable of meeting diverse scene requirements.

In this study, the large-scale 3D modeling software Blender 3.6 was used to create a virtual transmission line environment (including splicing sleeves) featuring typical scenarios such as farmland, villages, and forests. By varying perspectives, depths of field, low visibility (e.g., fog), and high noise conditions, diverse virtual reality scenes were generated, and the built-in camera tool in the software was employed to capture the scene data (as shown in Figure 11). This virtual scene augmentation method provides controllability and diversity of data samples, allowing for the simulation of various extreme conditions. It not only achieves low-cost and efficient data acquisition but also compensates for the high cost and difficulty associated with collecting complex scene data in real environments. Furthermore, it significantly enhances the robustness and generalization capabilities of the model. Through virtual scene augmentation, a total of 13,500 splicing sleeve image samples were generated in this study.

Moreover, the real splicing sleeve dataset, D_{real}, consists of 200 aerial images of 10 splicing sleeves from 6 high-voltage 220 kV power transmission lines in 3 regions of southern China. Based on the D_{real} and D_{vr}, data augmentation techniques such as color enhancement/weakening, contrast enhancement/weakening, perspective transformation, distortion, elastic deformation, and scaling [36–38] were applied to create the dataset D_{au}, with a total of 7900 images. The purpose of data augmentation is to simulate image distortions such as warping and blurring caused by UAV vibrations during actual aerial photography, enabling the splicing sleeve deep learning model to better adapt to real-world applications. If DA represents the data augmentation function, the augmented dataset D_{au} is expressed as:

$$D_{au} = DA(D_{real} + D_{vr}) \quad (15)$$

By combining D_{real}, D_{vr}, and D_{au}, a dataset D_{SS} with 21,600 images was constructed. During model training, D_{SS} was divided into training, validation, and test sets in a 7:2:1 ratio.

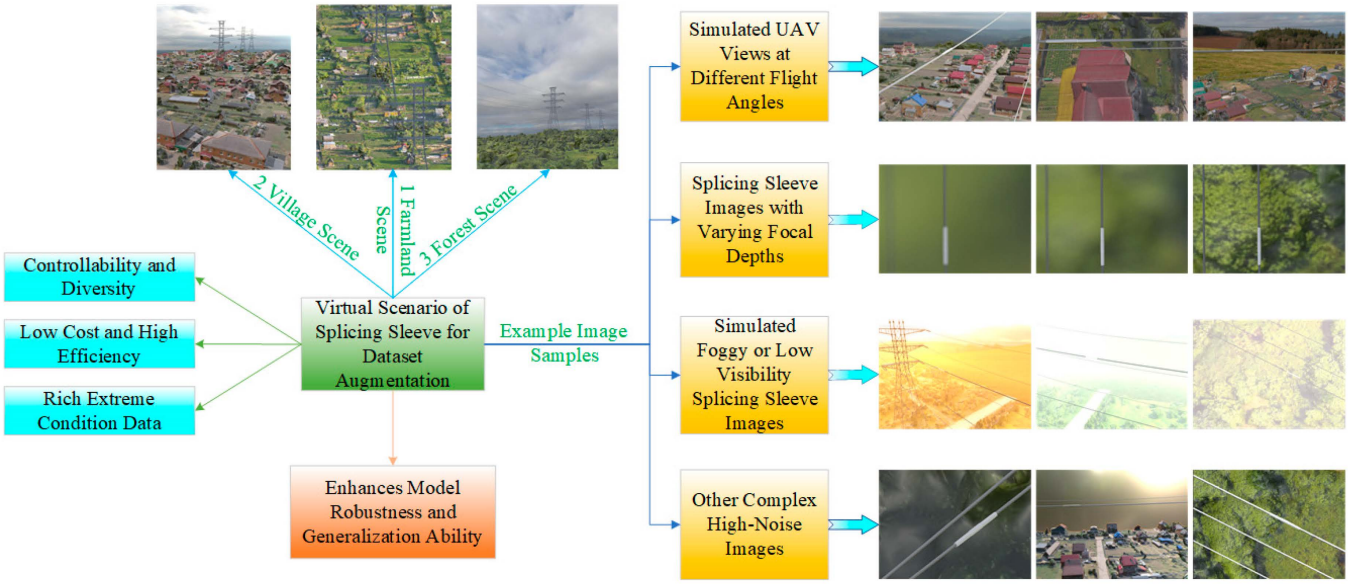


Figure 11. Virtual Scenario of Splicing Sleeve for Dataset Augmentation.

4.1.2. RB-YOLOv8n(OBB) Rotational Object Detection Rapid Localization Experiment

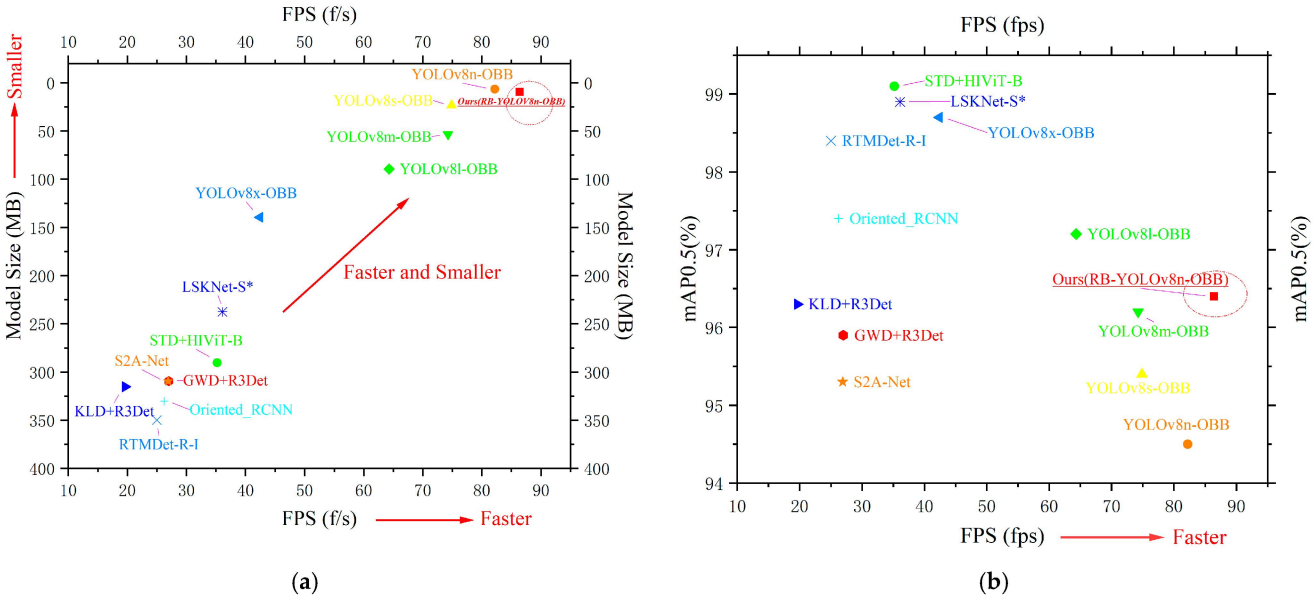
This section describes the rapid localization performance of RB-YOLOv8n(OBB) using the D_{SS} dataset relative to 12 mainstream rotational object detection algorithms, such as YOLOv8n/s/m/l/x-OBB, STD+HIViT-B [39], LSKNet-S* [40], RTMDet-R-I [41], KLD+R3Det [42], GWD+R3Det [43], S2anet [44], and Oriented_rcnn [45]. The key performance metrics used for comparison include mean average precision (mAP0.5), inference time per image (Spend, ms/img), inference speed (FPS), and model size (ModelSize).

All models were trained in the deep learning framework PyTorch under the Ubuntu 20.04.6 LTS operating system, using hardware that includes an NVIDIA GeForce RTX 3090 GPU, an AMD Ryzen 9 5950X 16-core 3.40 GHz processor, and 64.0 GB of memory. When evaluating the inference speed (FPS) for images of size 848 × 480, the GPU is pre-warmed before calculating the inference time.

Table 3 presents detailed values of precision, recall, mean average precision (mAP0.5%), inference time per image (Spend, ms/img), inference speed (FPS, f/s), and model size (Model Size, MB) for various models. Figure 12 provides a visual comparison of model size, mAP, and FPS through a two-dimensional scatter plot. As shown in Table 3 and Figure 12, our model (RB-YOLOv8n(OBB)) achieves an mAP0.5 of 96.4%, an inference time (Spend) of 11.57 ms, and a processing speed of 86.41 FPS. Compared to the original YOLOv8n-OBB model, although the model size increased by 2.8 MB, mAP0.5 increased by 2% inference time decreased by 0.6 ms, and FPS improved by 4.24 fps. This indicates that in RB-YOLOv8n(OBB), replacing the original cf2 module with the RepBlock module, and using different network structures for training and inference, where multiple branches are reparameterized into a single branch during inference, leads to superior mAP0.5, Spend, and FPS performance.

Table 3. Comparison of Key Performance Indicators for Different Models on the Splicing Sleeve Dataset.

Rotating Target Detection Model	Precision (%)	Recall (%)	mAP0.5 (%)	Spend (ms/img)	FPS (f/s)	Model Size (MB)
Ours	96.7	95.7	96.4	11.57	86.41	9.4
YOLOv8n-OBB [46]	98.8	88.2	94.5	12.17	82.17	6.6
YOLOv8s-OBB [46]	95.8	87.8	95.4	13.35	74.91	23.3
YOLOv8m-OBB [46]	96.2	92.5	96.2	13.47	74.24	53.3
YOLOv8l-OBB [46]	98.4	93.9	97.2	15.54	64.35	89.5
YOLOv8x-OBB [46]	98.8	94.6	98.7	23.58	42.41	139.5
KLD+R3Det t [42]	95.8	93.5	96.3	50.92	19.64	315.3
GWD+R3Det [43]	95.2	90.1	95.9	37.04	27.00	309.5
S2A-Net [44]	94.3	93	95.3	37.17	26.90	309.6
Oriented_RCNN [45]	98.3	92.6	97.4	38.14	26.22	330.3
STD+HIViT-B [39]	99.0	98.5	99.1	28.41	35.20	290.5
LSKNet-S* [40]	97.2	96.5	98.9	27.70	36.10	237.6
RTMDet-R-I [41]	93.4	95.2	98.4	39.97	25.02	350.1



(**a**) (**b**)

Figure 12. Comparison of Model Size, mAP0.5, and Speed for Different Rotational Object Detection Models. (**a**) Model Size (MB) vs. FPS; (**b**) mAP0.5 (%) vs. FPS.

The results in Table 3 and Figure 12 also show that comparison with other models in the YOLOv8-OBB series (such as YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x), RB-YOLOv8n(OBB) demonstrates significant advantages in terms of Spend, FPS, and Model Size. Although the mAP0.5 of YOLOv8m-OBB is 96.2%, which is very close to the 96.4% achieved by RB-YOLOv8n(OBB), the Model Size of YOLOv8m-OBB is 53.3 MB, much larger than the 9.4 MB of RB-YOLOv8n(OBB). Although the mAP0.5 values are comparable, RB-YOLOv8n(OBB) exhibits much better results in terms of Spend, FPS, and Model Size.

Compared to other models such as KLD+R3Det, GWD+R3Det, S2A-Net, Oriented_RCNN, STD+HIViT-B, LSKNet-S*, and RTMDet-R-I, the RB-YOLOv8n(OBB) model has a similar mAP0.5, ranking at a medium level. However, in terms of Spend, FPS, and Model Size, RB-YOLOv8n(OBB) significantly outperforms these models, as shown in Figure 12b.

To verify the practical effectiveness and general applicability of the RB-YOLOv8n(OBB) model, performance tests were conducted in various hazy static, real/simulated static, and UAV aerial dynamic scenarios. Figure 13 shows the recognition results of RB-YOLOv8n(OBB) in various environments, with (a)–(c), (d)–(f), and (g)–(i) corresponding to the hazy static, real/virtual, and UAV aerial dynamic scenarios, respectively. The dynamic recognition

results are shown in the Supplementary Video S0. The results demonstrate that RB-YOLOv8n(OBB) has high general applicability and allows for effective recognition of splicing sleeves in various scenarios.

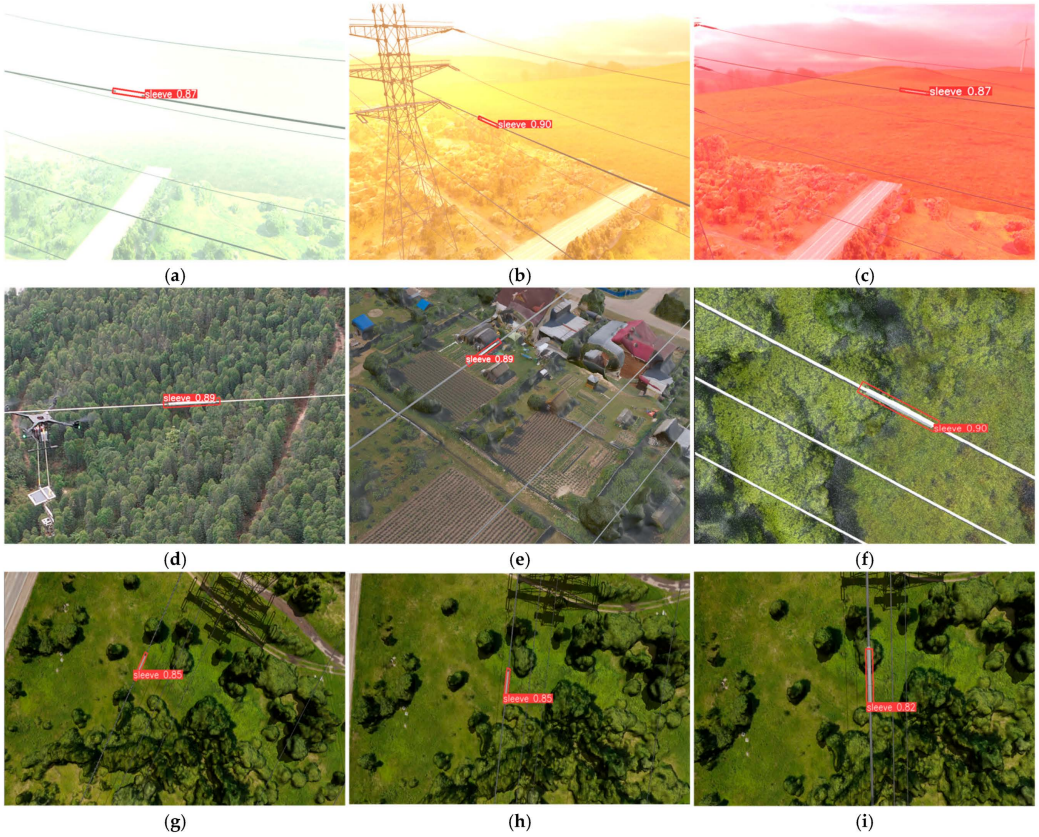


Figure 13. Different scene recognition effect diagrams, with (**a**–**c**), (**d**–**f**), and (**g**–**i**) corresponding to the hazy static, real/virtual, and UAV aerial dynamic scenarios, respectively.

In summary, by replacing the original cf2 module with the RepBlock module and using reparameterization to convert the multi-branch network structure into a single-branch structure, RB-YOLOv8n(OBB) achieved performance improvements: inference speed reached 86.41 (f/s), mean average precision (mAP0.5) was 96.4%, and the model size was only 9.4 MB. These results fully demonstrate the effectiveness of the improvements to YOLOv8n(OBB). RB-YOLOv8n(OBB) strikes an excellent balance between compact model size and outstanding inference speed, making it particularly suitable for real-time applications.

4.1.3. LC Rotational Object Detection Fine Localization Experiment

To quantitatively analyze the differences between the rapid localization method for rotating targets and the precise localization method incorporating local clustering, we collected 500 sets of target images containing RGB and depth images using a depth camera under various angles, distances, depths of field, and scene conditions in both a real laboratory environment and a virtual scene created in Blender 3.6.

The position of the splicing sleeve is represented by five parameters: x, y, w, h, and *angle*. The splicing sleeves in 500 RGB images were annotated with nominal values (Nom) using the rotational object annotation software roLabelImg (version 3.0). The RB-YOLOv8n(OBB) model could predict the rapid localization of the splicing sleeves in the RGB images, denoted as B_{C_RGB}. Subsequently, the LC rotational object detection fine localization algorithm was applied to build the LC-RB-YOLOv8n(OBB) method, providing the precise position of the splicing sleeves in the RGB images, denoted as B_{F_RGB}. Figure 14 illustrates the positions of Nom, B_{C_RGB}, and B_{F_RGB} for the same target.

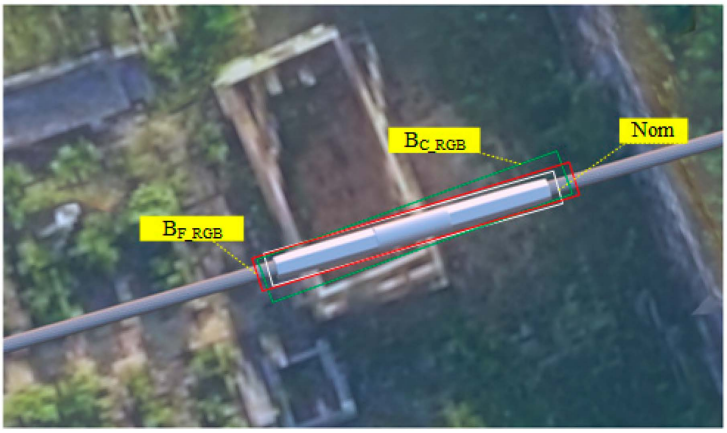


Figure 14. Diagram of Localization Using Three Methods.

In this study, four evaluation metrics were used to measure the differences between Nom, B_{C_RGB}, and B_{F_RGB}: mean absolute error (MAE), mean relative error (MRE), root mean squared error (RMSE), and Spearman's rank correlation coefficient ρ. The Spearman correlation coefficient ρ is determined by the following formula:

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{16}$$

where d_i is the rank difference of each parameter in the data points, and n = 500.

Smaller values of MAE, MRE, and RMSE indicate a smaller difference between the localization results and Nom, reflecting higher localization accuracy. When ρ approaches 1, it suggests a higher consistency between the localization method and the Nom annotation method. Table 4 provides a comparison of difference metrics for 500 sets of B_{C_RGB}, B_{F_RGB}, and Nom, detailing the differences across four evaluation metrics (Els) for the five coordinate parameters x, y, w, h, and *angle*. Figure 15 visually illustrates the differences in MAE, MRE, RMSE, and ρ between the two methods.

Table 4. Comparison Table of B_{C_RGB}, B_{F_RGB}, and Nom Difference Indicators for 500 Data Sets.

Model	B_{C_RGB}				B_{F_RGB}			
Els / Parameters	MAE	MRE	RMSE	ρ	MAE	MRE	RMSE	ρ
x	40.55	9.05%	51.25	0.95	15.28	3.39%	19.47	0.99
y	40.38	9.34%	51.58	0.95	16.07	3.65%	19.88	0.99
w	23.66	12.99%	28.77	0.85	7.49	4.21%	9.51	0.98
h	2.47	12.7%	3.06	0.66	0.77	3.96%	0.95	0.95
angle	7.94	12.97%	9.86	0.83	2.33	3.83%	2.97	0.98

As shown in Table 4 and Figure 15, the localization results of B_{F_RGB} using LC-RB-YOLOv8n(OBB) demonstrate superior performance across four evaluation metrics. For the five coordinate parameters x, y, w, h, and *angle*, the MAE values for B_{F_RGB} decreased from 40.55, 40.38, 23.66, 2.47, and 7.94 to 15.28, 16.07, 7.49, 0.77, and 2.33, respectively, showing a substantial improvement in localization precision. The MRE values for B_{F_RGB} decreased significantly, from 9.05%, 9.34%, 12.99%, 12.7%, and 12.97% to 3.39%, 3.65%, 4.21%, 3.96%, and 3.83%, respectively, thereby improving localization accuracy. The RMSE values of B_{F_RGB} for x, y, w, h, and *angle* ranged from 0.95 to 19.88, achieving relative improvements over the RMSE values of B_{C_RGB} by 62.01%, 61.46%, 66.94%, 68.95%, and 69.88%, respectively. This indicates that the introduction of the clustering-based LC rotational object detection refinement algorithm into RB-YOLOv8n(OBB) significantly enhances

the stability of the *x*, *y*, *w*, *h*, and *angle* parameters. Additionally, Figure 15d shows a radar plot illustrating improvements in the correlation coefficients for B_{F_RGB}, with increases of 0.04, 0.04, 0.13, 0.26, and 0.15 for the parameters *x*, *y*, *w*, *h*, and *angle*, reaching values of 0.99, 0.99, 0.98, 0.95, and 0.98, respectively. These results indicate that LC-RB-YOLOv8n(OBB) achieves high consistency with manual annotation in rotating target localization.

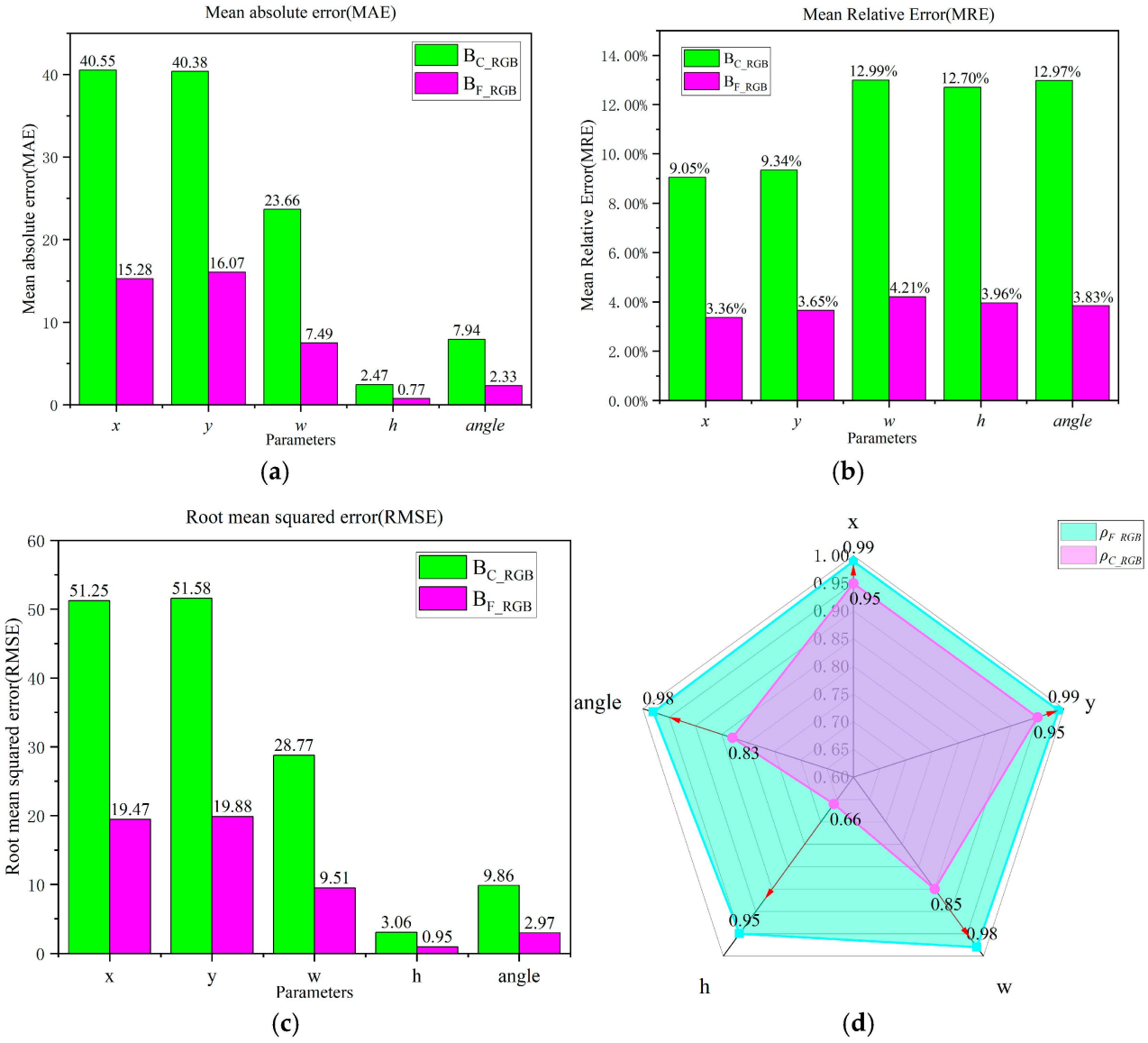


Figure 15. Comparison of Five Coordinate Parameters Using Different Methods Across Various Metrics. (**a**) MAE of Parameters; (**b**) MRE of Parameters; (**c**) RMSE of Parameters; (**d**) ρ of Parameters.

Figure 16 shows the positioning results of B_{C_RGB}, B_{F_RGB}, and B_{F_Depth}. Panels (a)–(c), as well as (d)–(f), represent the B_{C_RGB}, B_{F_RGB}, and the B_{F_Depth} results when the distance between the UAV and the splicing sleeve is 4.8 m and 1.2 m, respectively. Panels (g)–(i) display the positioning results of B_{C_RGB}, B_{F_RGB}, and B_{F_Depth} using an Intel D455 depth camera in a laboratory environment. The figures clearly show that the B_{F_RGB} bounding box better fits the splicing sleeve, with significant improvements in the accuracy of angle, width, and height localization.

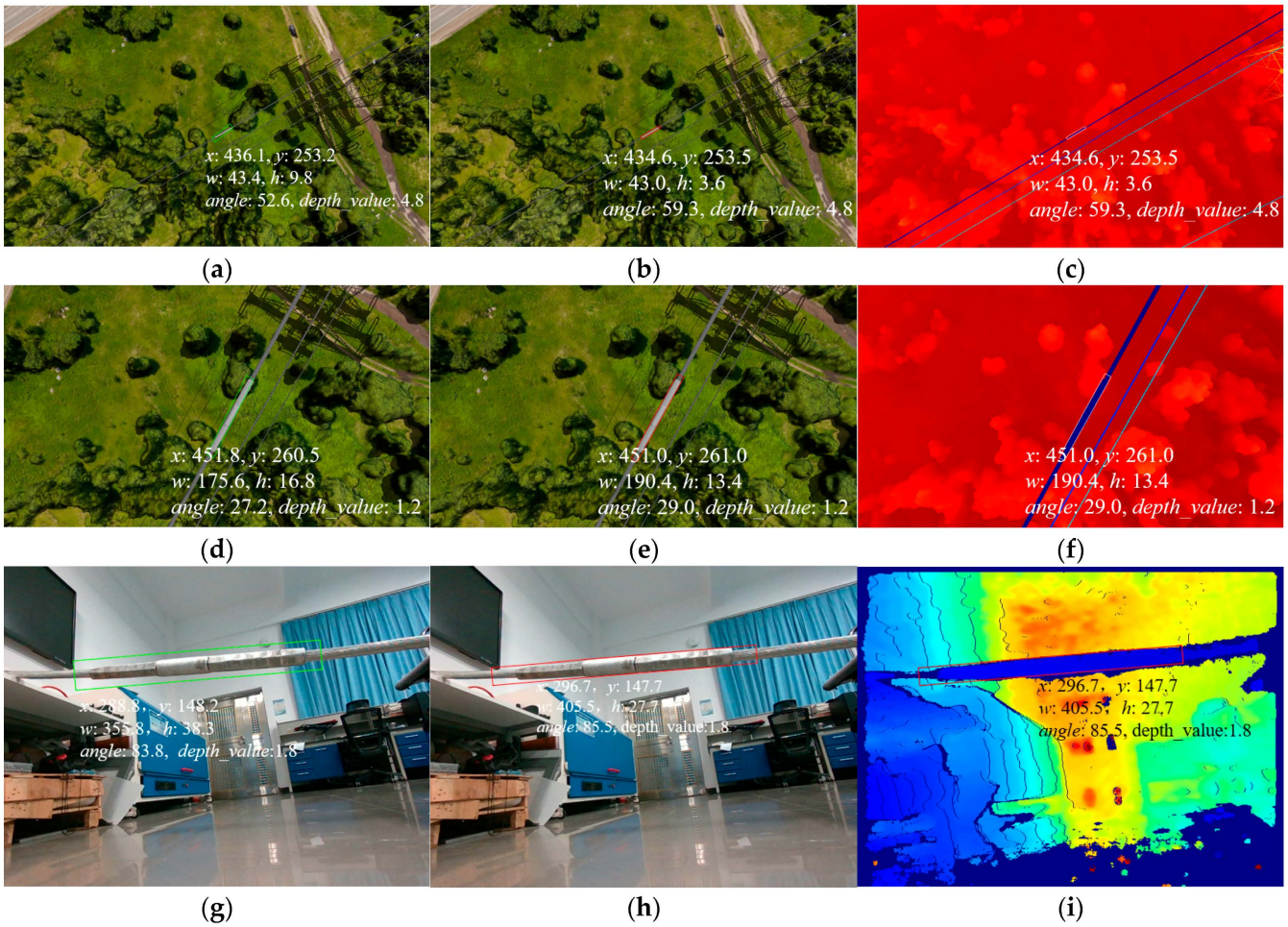


Figure 16. Positioning results of B_{C_RGB} and B_{F_RGB}, B_{F_Depth}. Panels (**a**–**c**), as well as (**d**–**f**), represent the B_{C_RGB}, B_{F_RGB}, and the B_{F_Depth} results when the distance between the UAV and the splicing sleeve is 4.8 m and 1.2 m, respectively. Panels (**g**–**i**) display the positioning results of B_{C_RGB}, B_{F_RGB}, and B_{F_Depth} using an Intel D455 depth camera in a laboratosry environment. The resolution of the images in panels (**a**–**f**) is 848 × 480, while the resolution in panels (**g**–**i**) is 640 × 480.

The experimental results indicate that the LC-RB-YOLOv8n(OBB) method significantly outperforms RB-YOLOv8n(OBB) in terms of localization accuracy and consistency with annotations. The optimization algorithm based on local clustering significantly reduces MAE, MRE, and RMSE values for rotational targets, improving the stability of angle and size parameters. This improvement is crucial for high-precision UAV docking tasks, providing a solid foundation for accurate docking of linear targets.

4.2. UAV-ASS Visual Simulation Experiment Series

The experimental results discussed above verified that the LC-RB-YOLOv8n(OBB) model effectively recognizes and locates splicing sleeves. However, after recognition, relative positional information is still needed to guide the UAV-mounted equipment for DR inspection of the splicing sleeve. The UAV-ASS visual simulation series includes the construction of the UAV-ASS visual simulation platform, studies on the effect of UAV rotation on $D_{UAV\text{-}SS}$, and studies of the UAV-ASS waypoint planning and docking.

4.2.1. UAV-ASS Visual Simulation Platform

This study developed the UAV-ASS visual simulation system to accelerate the validation of the scientifically founded and effective UAV-based recognition and approach algorithms, which are based on LC-RB-YOLOv8n(OBB). Figure 17 shows the structural diagram of the UAV-ASS visual simulation system built on the Ubuntu 20.04.6 LTS platform. The system consists of several units, including the splicing sleeve visual recognition and positioning module LC-RB-YOLOv8n(OBB), the UAV control system, the communication layer, and the GAZEBO simulation module. In the visual recognition and positioning module, nodes are used to read RGB images and depth maps from the depth camera. The LC-RB-YOLOv8n(OBB) is invoked to identify and locate the splicing sleeve. The results of the recognition and positioning are then transferred through a topic communication mechanism. The UAV control system unit, using the open-source PX4 flight controller, implements functions such as UAV altitude, position, and speed control, while also adding coordinate transformation and waypoint planning. The GAZEBO simulator, with its high-fidelity physical simulation capabilities, integrates the UAV and environmental models. The environmental models were created using Blender 3.6 to generate realistic 3D scenes of high-voltage power lines and splicing sleeves. The communication layer enables data transmission between ROS and the UAV, and the control command transmission according to the MAVLink and MAVROS protocol.

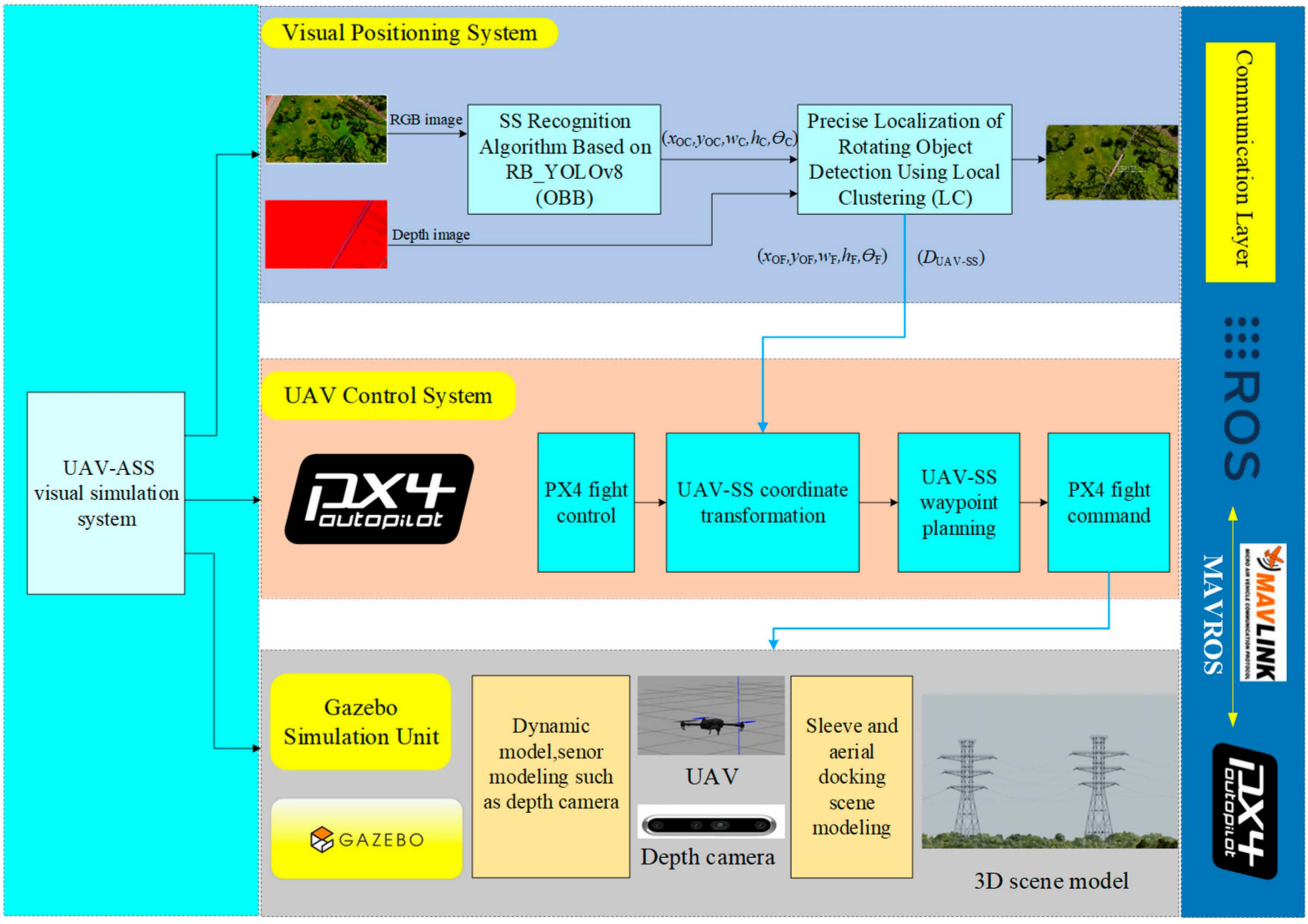


Figure 17. UAV-ASS visual simulation system.

Figure 18 shows the interface of the UAV-ASS visual simulation system. Figure 18a displays the main interface of the UAV-ASS simulation. Figure 18b,c show the resulting RGB images and the depth map of the UAV's visual recognition and localization of the splicing sleeve, respectively. The output results include x_{OF}, y_{OF}, w_F, h_F, θ_F, $D_{UAV\text{-}SS}$. Please refer to the Supplementary Video S1, for more details.

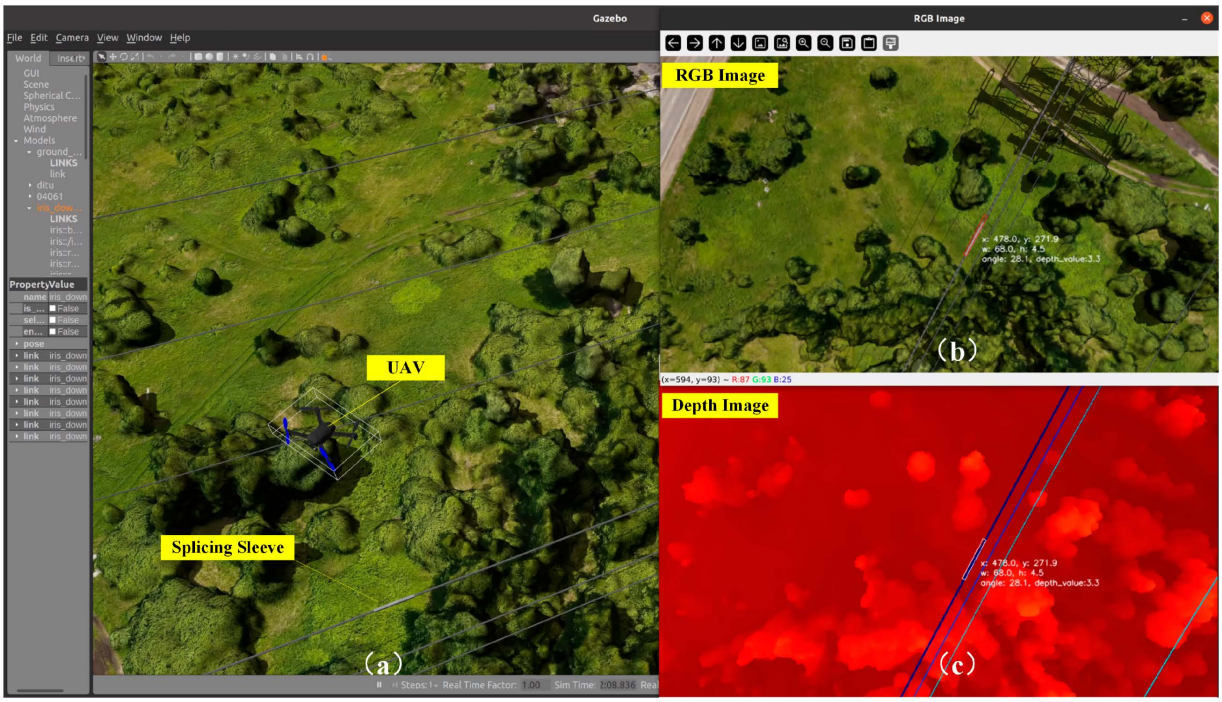


Figure 18. UAV-ASS visual simulation system interface. (**a**) Main interface of the UAV-ASS simulation; (**b**) resulting RGB image of the UAV's visual recognition and localization of the splicing sleeve; (**c**) depth map of the UAV's visual recognition and localization of the splicing sleeve.

4.2.2. UAV Fixed-Point Rotation Impact on $D_{UAV\text{-}SS}$ Experiment

In the UAV-ASS visual simulation system, the UAV is set to maintain a fixed height above the splicing sleeve while rotating intermittently. RGB images and depth maps are captured at a rate of 20 frames per second over a given time, and two localization methods are used to obtain B_{C_RGB} (including B_{C_Depth}) and B_{F_RGB} (including B_{F_Depth}). The depth value $D_{UAV\text{-}SS}$ is then calculated according to Equation (11). Figure 19 shows the changes in $D_{UAV\text{-}SS}$ determined with the B_{C_Depth} (Figure 19a, and B_{F_Depth} (Figure 19b methods, with the horizontal axis representing time (s) and the vertical axis representing distance (m).

Figure 19b shows that the $D_{UAV\text{-}SS}$ values obtained using the B_{C_Depth} method fluctuate significantly due to UAV rotation, which causes inaccurate fitting of the splicing sleeve's bounding box by the B_{C_RGB} and B_{C_Depth} methods. In contrast, Figure 19c shows that the $D_{UAV\text{-}SS}$ values obtained using the B_{F_Depth} method remain stable at 3.4 m, even when the UAV rotates. The B_{F_RGB} (including B_{F_Depth}) method accurately fits the splicing sleeve despite the UAV rotation. This indicates that the B_{F_RGB} method effectively mitigates the effects of UAV rotation, improving the stability of distance measurement. The Supplementary Video S2, provides more details.

This experiment further validated the robustness and stability of the LC-RB-YOLOv8n(OBB) method, particularly its potential applications in dynamic scenarios, providing a solid technical foundation for precise depth measurement of UAVs in complex rotational and dynamic environments.

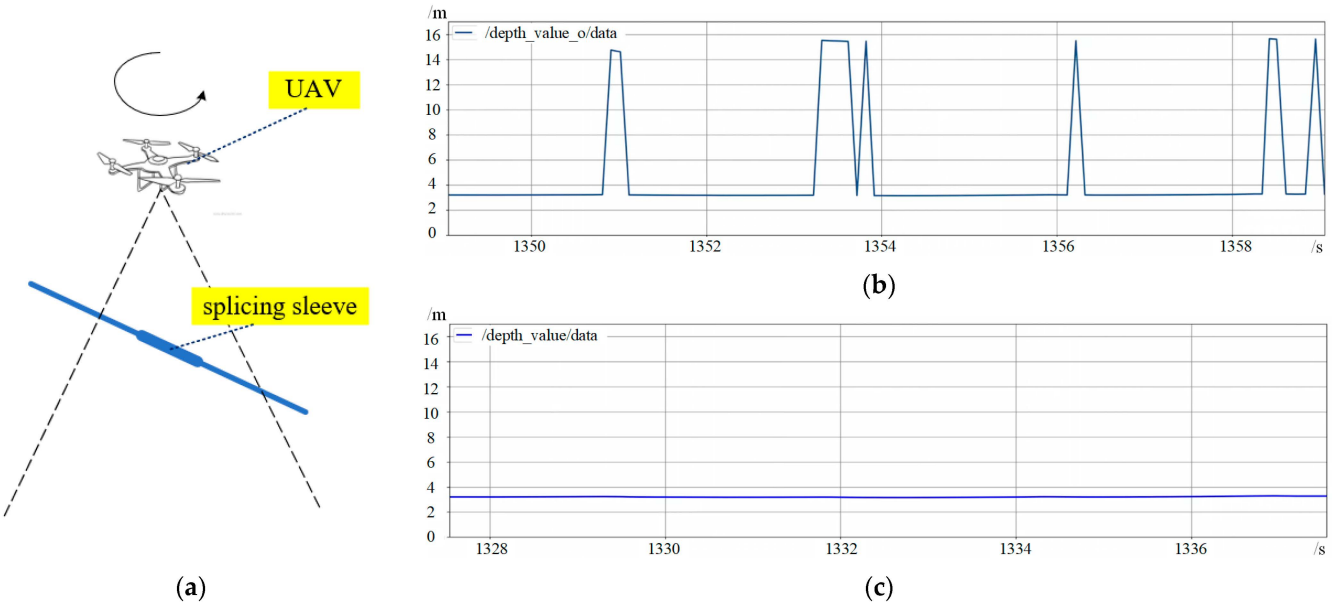


Figure 19. UAV calculating $D_{\text{UAV-SS}}$ using B_{C_Depth} and B_{F_Depth} for splicing sleeve localization. (**a**) UAV Fixed-Point Rotation; (**b**) $D_{\text{UAV-SS}}$ extraction using B_{C_Depth} localization; (**c**) $D_{\text{UAV-SS}}$ extraction using B_{F_Depth} localization.

4.2.3. UAV-ASS Waypoint Planning and Docking Experiment

According to the UAV-ASS approach planning described in Section 2.2, the splicing sleeve's final position relative to the UAV's body in the simulation system is set as ${}^{b}P_{ss}|_{end} = (0, 0, -1, 0)$, which serves as the criterion for determining successful docking. After the UAV identifies the splicing sleeve during the high-altitude search, the ROS system retrieves the UAV's world coordinates at the starting and end points through coordinate transformation and generates flight waypoints according to Equation (14). The simulated process of final identification and docking is divided into three stages: hovering and identification, position and altitude adjustments.

Figure 20 shows video screenshots of the UAV's body coordinate trajectory and the B_{F_RGB}-located splicing sleeve position (including position and depth) during the recognition and docking process. In frames 1 to 7, the UAV is in the position adjustment stage; from frame 8 onwards, it enters the altitude adjustment stage, and by frame 11, the UAV is hovering approximately 1.0 m above the splicing sleeve. The Supplementary Video S3, shows the detailed steps. Figure 21 illustrates the corresponding changes in the UAV's position adjustment process during the approach and docking, as seen in Figure 20. During the position adjustment stage, the angle θ between the splicing sleeve and the UAV's body axis (i.e., the UAV's yaw angle) adjusts from 60° to 0°. At the final position, the UAV's world coordinates are (14.5 m, −14.3 m, 12.2 m).

To verify the robustness of the approach algorithm, UAV recognition and approach to the splicing sleeve from different initial positions were studied. The UAV's initial yaw angle was set to 30°, and the end recognition and docking algorithm was executed from four positions on the same plane above the splicing sleeve: (14.5 m, −12.5 m, 15.5 m), (15.5 m, −14.5 m, 15.5 m), (14.5 m, −16.5 m, 15.5 m), and (13.5 m, −14.5 m, 15.5 m). The world coordinates of the splicing sleeve's center point in GAZEBO were set to (14.5 m, −14.5 m, 11.1 m), with the splicing sleeve's axis aligned with the X-axis of the world coordinate system. Figure 22 shows the UAV's four different initial positions, and Table 5 presents the resulting errors in the recognition and approach experimental data for different starting positions.

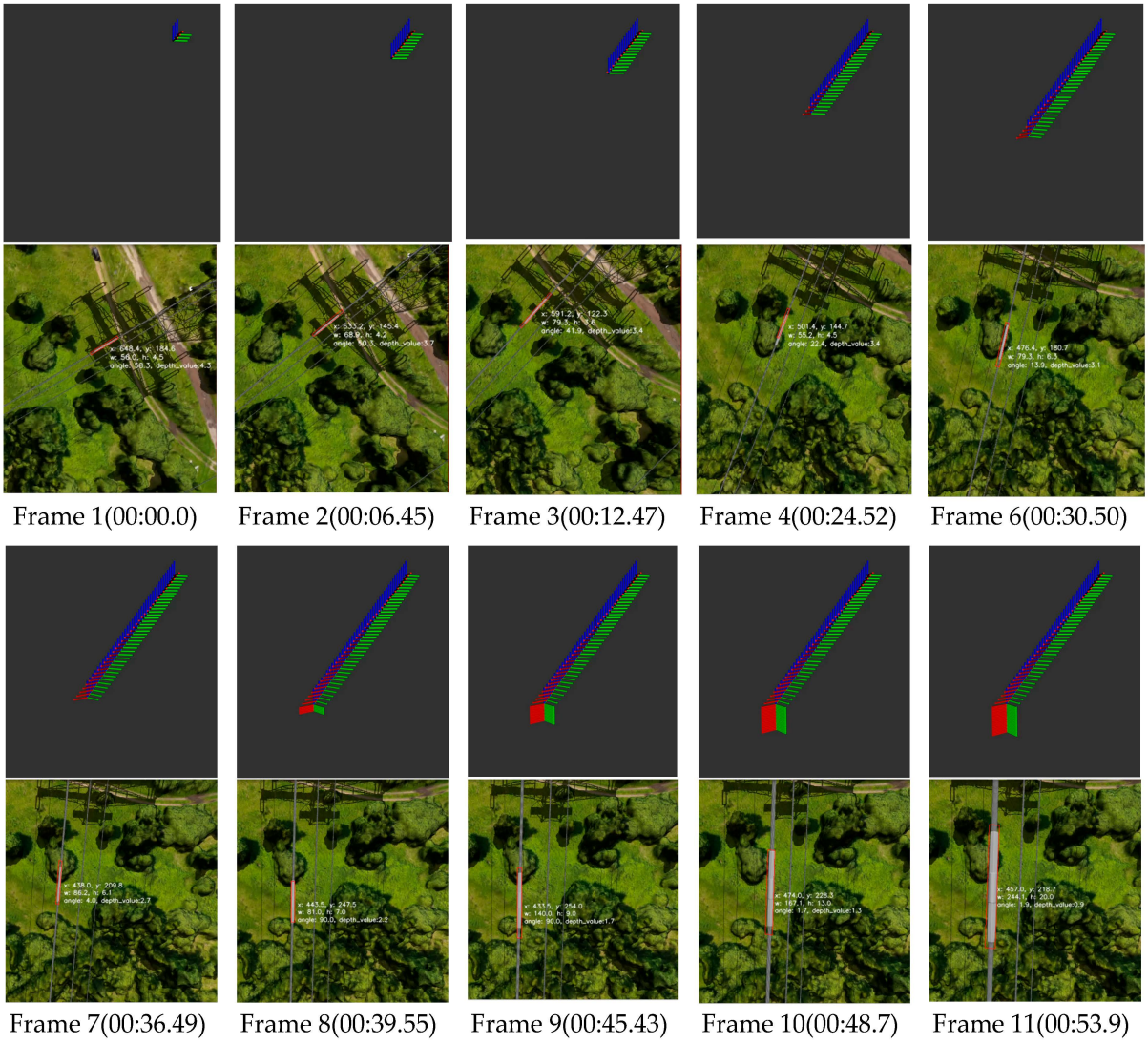


Figure 20. Video screenshots of the UAV body coordinate trajectory and B_{F_RGB}-located splicing sleeve position during the UAV recognition and docking process.

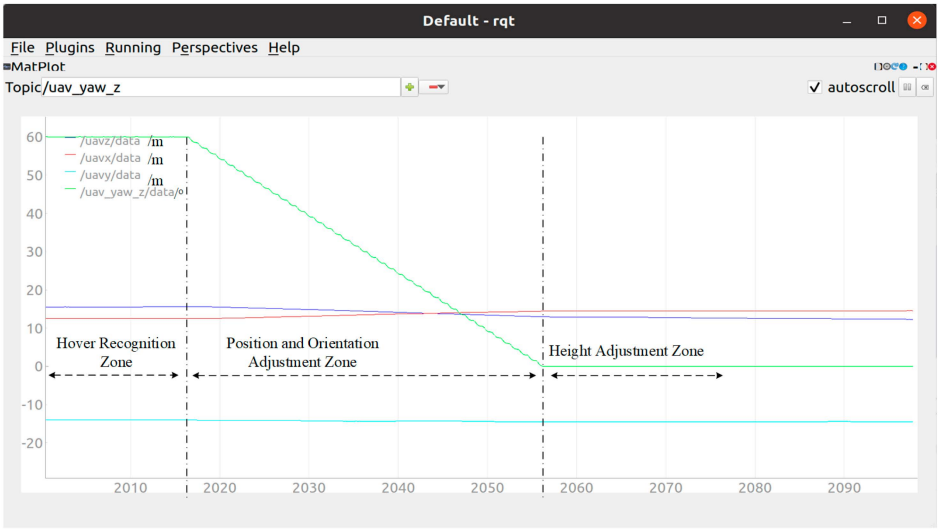


Figure 21. Changes in the UAV Pose Adjustment Process during Approach and Docking.

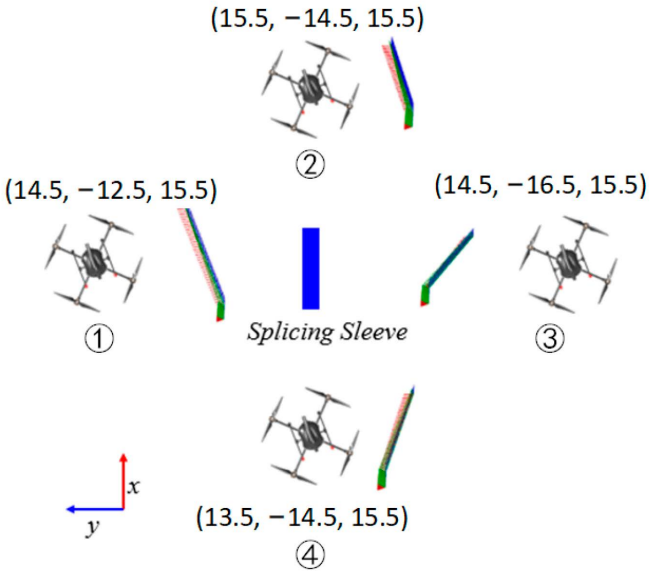


Figure 22. UAV Initial Positions at Different Starting Points.

Table 5. Errors in Recognition and Approach to Splicing Sleeve from Different Starting Positions in Simulations.

Location Error / Starting Position	X-Axis Error/m	Y-Axis Error/m	Z-Axis Error/m	Angle Error/°
① (14.5 m, −12.5 m, 15.5 m)	−0.05	+0.06	−0.07	2
② (15.5 m, −14.5 m, 15.5 m)	+0.03	+0.03	−0.06	3
③ (14.5 m, −16.5 m, 15.5 m)	−0.05	+0.06	+0.04	−4
④ (13.5 m, −14.5 m, 15.5 m)	+0.07	+0.05	−0.06	2
MAE	0.050	0.050	0.058	2.750

The Table 5 shows that the UAV, starting from four different positions, flew to the target with an average absolute error (MAE) of 0.050 m, 0.050 m, 0.058 m in the X-, Y-, and Z-axes, respectively, and an angular error of 2.75°. This is comparable to the X-axis and Y-axis positioning errors of 0.050 m and 0.0650 m described in Ma's method [47] and Gong's 5 Points method [8] for UAV autonomous aerial refueling using binocular depth cameras. However, Ma's method and Gong's 5 Points method did not mention errors in the Z-axis or angular direction. The angular error of 2.75° is acceptable for mechanical docking in real-world environments when using ⌒- or Λ-shaped clamps [47,48].

The experiment validated the robustness and applicability of the UAV-ASS way-point planning and docking algorithm under various initial conditions. The results demonstrated that the algorithm achieved high-precision localization along the X-, Y-, and Z-axes while providing accurate angular measurements, with angular errors within the acceptable range for mechanical docking. This study further highlights the practical potential of the UAV-ASS algorithm in complex and dynamic environments, offering essential technical support for precise recognition and docking of linear tar-gets in UAV missions.

5. Conclusions and Future Works

To achieve autonomous recognition and approach control of linear target splicing sleeves by unmanned aerial vehicles (UAVs), this study proposes an approach that integrates deep learning and active stereo vision for UAV recognition and approach of linear targets, with experimental validation providing the following insights:

(1) An algorithm based on the LC-RB-YOLOv8n(OBB) framework was designed specifically for linear target splicing sleeves. This algorithm employs a two-stage localization strategy, achieving rapid and precise localization of splicing sleeves with high reliability. The recognition model within this algorithm attains an average detection accuracy (mAP0.5) of 96.4% and an image inference speed of 86.41 f/s, meeting the real-time and lightweight requirements for high-altitude UAV detection. Additionally, the algorithm integrates depth images of splicing sleeves and performs local clustering analysis of depth values to further enhance the accuracy and reliability of sleeve localization. This algorithm provides a valuable reference for UAV autonomous recognition and docking with linear targets.

(2) Typical virtual scenarios of splicing sleeves containing linear targets were constructed using the 3D modeling software Blender 3.6 to expand the dataset. This approach addresses the high cost and safety risks associated with capturing images of high-altitude transmission line splicing sleeves. It also improves the robustness and generalization ability of the model, providing a valuable reference for constructing deep learning datasets for aerial key targets in the power industry.

(3) Utilizing the open-source PX4 UAV flight control platform, the Robot Operating System (ROS), and the Gazebo physics simulation platform, a UAV-ASS visual simulation platform was developed to quickly validate high-risk algorithms for UAV autonomous recognition and approach of overhead transmission line splicing sleeves. This platform provides an efficient simulation validation tool for high-altitude UAV operations in the power industry, effectively reducing testing costs and safety risks.

In summary, these methods and insights provide essential support for the development of UAV autonomous recognition and docking technologies in complex environments. Future work will focus on the following directions: firstly, exploring vibration compensation and error correction methods, such as Kalman filtering or machine learning models; secondly, investigating the stability of the docking process by employing robust control and adaptive feedback mechanisms; thirdly, conducting in-depth research on 6D pose estimation for non-parallel splicing sleeves; and finally, addressing obstacle avoidance and path optimization during the approach phase, as well as studying approaches to splicing sleeves under live-line conditions.

Supplementary Materials: The following supporting information can be downloaded at: https://zenodo.org/records/14101117, accessed on 8 December 2024, Video S0–S3.

Author Contributions: G.Z.: Conceptualization, Methodology, Software, Data curation, Writing—original draft, Visualization. G.L.: Writing—review and editing, Supervision, Project administration. F.Z.: Feasibility analysis, Resources, Funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Southern Power Grid Guangdong Yue Dianke Testing and Inspection Technology Co., Ltd. under Grant No. GDYDKKJ2023-02.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors would like to thank the anonymous reviewers and members of the editorial team for their comments and contributions.

Conflicts of Interest: Author Fei Zhong was employed by the company South Power Grid Technology Co., 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.

Abbreviations

Abbreviation	Full Form/Explanation
UAV	Unmanned Aerial Vehicle
ASS	Approach Splicing Sleeve
SS	Splicing Sleeve
OBB	Oriented Bounding Box
ROS	Robot Operating System
UAV-ASS	UAV approach splicing sleeve
mAP	mean Average Precision
MRE	Mean Relative Error
DR	Digital Radiography
GPS	Global Positioning System
INS	Inertial Navigation System
LiDAR	Light Detection and Ranging
IMU	Inertial Measurement Unit
UWB	Ultra Wideband
GNSS	Global Navigation Satellite System
LC	Local Clustering
RB	Reparameterization Block
RB-YOLOv8(OBB)	A fast rotational object detection model for SS, after reparameterization with the RB module
LC-RB-YOLOv8(OBB)	RB-YOLOv8(OBB) model integrated with depth information and local clustering (LC) method
RepBlock	Reparameterization Block
PAN	Path Aggregation Network
FPN	Feature Pyramid Network
BN	Batch Normalization
Conv	Convolutional Layer
$\mathrm{Conv}_{\mathrm{eq}}$	Equivalent Convolutional Layer
D_{real}	The real splicing sleeve dataset
D_{vr}	The virtual reality scene splicing sleeve dataset
D_{au}	The augmented splicing sleeve dataset
D_{SS}	The splicing sleeve dataset
FPS	Frames Per Second
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
Els	Evaluation Metrics
$\mathrm{B}_{\mathrm{C_RGB}}$	The detection results of splicing sleeves using the RB-YOLOv8(OBB) model in RGB images
$\mathrm{B}_{\mathrm{R_Depth}}$	The detection results of splicing sleeves using clustering and minimum area rectangle in local areas (rotation box fitting)
$\mathrm{B}_{\mathrm{F_RGB}}$	The detection results of splicing sleeves using LC-RB-YOLOv8(OBB) in RGB images
$\mathrm{B}_{\mathrm{F_Depth}}$	The detection results of splicing sleeves using LC-RB-YOLOv8(OBB) in depth images
$\mathrm{B}_{\mathrm{C_Depth}}$	The detection results of splicing sleeves using RB-YOLOv8(OBB) in depth images
$\rho_{\mathrm{C_RGB}}$	The correlation coefficient between the detection results using RB-YOLOv8(OBB) and manually labeled results in RGB images
$\rho_{\mathrm{F_RGB}}$	The correlation coefficient between the detection results using LC-RB-YOLOv8(OBB) and manually labeled results in RGB images
$D_{\mathrm{UAV\text{-}SS}}$	UAV's relative distance to the splicing sleeve

References

1. Liu, Z.; Wu, G.; He, W.; Fan, F.; Ye, X. Key target and defect detection of high-voltage power transmission lines with deep learning. *Int. J. Electr. Power Energy Syst.* **2022**, *142*, 108277. [CrossRef]
2. Wong, S.Y.; Choe, C.W.C.; Goh, H.H.; Low, Y.W.; Cheah, D.Y.S.; Pang, C. Power Transmission Line Fault Detection and Diagnosis Based on Artificial Intelligence Approach and its Development in UAV: A Review. *Arab. J. Sci. Eng.* **2021**, *46*, 9305–9331. [CrossRef]
3. Liu, K.; Li, B.; Qin, L.; Li, Q.; Zhao, F.; Wang, Q.; Xu, Z.; Yu, J. Review of application research of deep learning object detection algorithms in insulator defect detection of overhead transmission lines. *High Volt. Eng.* **2023**, *49*, 3584–3595. [CrossRef]
4. Qin, W.; Yu, G.; Yu, C.; Zhu, K.; Liang, J.; Liu, T. Research on Electromagnetic Interference Protection of X-ray Detecting Device for Tension Clamp of Transmission Line. In Proceedings of the 2022 7th Asia Conference on Power and Electrical Engineering (ACPEE), Hangzhou, China, 15–17 April 2022; pp. 1436–1440.
5. Liu, Y.; Zhao, P.; Qin, X.; Liu, Y.; Tao, Y.; Jiang, S.; Li, Y. Research on X-ray In-situ Image Processing Technology for Electric Power Strain Clamp. In Proceedings of the Conference on AOPC—Optical Sensing and Imaging Technology, Beijing, China, 20–22 June 2021.
6. Li, J.; Chen, D.; Li, J.; Zeng, C. Live detection method of transmission line piezoelectric tube defects based on UAV. In Proceedings of the Third International Conference on Artificial Intelligence and Computer Engineering (ICAICE 2022), Wuhan, China, 4–6 November 2022; pp. 41–46.
7. Luo, D.; Shao, J.; Xu, Y.; Zhang, J. Docking navigation method for UAV autonomous aerial refueling. *Sci. China Inf. Sci.* **2018**, *62*, 10203. [CrossRef]
8. Gong, K.; Liu, B.; Xu, X.; Xu, Y.; He, Y.; Zhang, Z.; Rasol, J. Research of an Unmanned Aerial Vehicle Autonomous Aerial Refueling Docking Method Based on Binocular Vision. *Drones* **2023**, *7*, 433. [CrossRef]
9. Bacelar, T.; Madeiras, J.; Melicio, R.; Cardeira, C.; Oliveira, P. On-board implementation and experimental validation of collaborative transportation of loads with multiple UAVs. *Aerosp. Sci. Technol.* **2020**, *107*, 106284. [CrossRef]
10. Miao, Y.; Tang, Y.; Alzahrani, B.A.; Barnawi, A.; Alafif, T.; Hu, L. Airborne LiDAR Assisted Obstacle Recognition and Intrusion Detection Towards Unmanned Aerial Vehicle: Architecture, Modeling and Evaluation. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4531–4540. [CrossRef]
11. Huang, J.; He, W.; Yao, Y. Multifiltering Algorithm for Enhancing the Accuracy of Individual Tree Parameter Extraction at *Eucalyptus* Plantations Using LiDAR Data. *Forests* **2023**, *15*, 81. [CrossRef]
12. Wang, Z.-H.; Chen, W.-J.; Qin, K.-Y. Dynamic Target Tracking and Ingressing of a Small UAV Using Monocular Sensor Based on the Geometric Constraints. *Electronics* **2021**, *10*, 1931. [CrossRef]
13. Meng, X.; Xi, H.; Wei, J.; He, Y.; Han, J.; Song, A. Rotorcraft aerial vehicle's contact-based landing and vision-based localization research. *Robotica* **2022**, *41*, 1127–1144. [CrossRef]
14. Luo, S.; Liang, Y.; Luo, Z.; Liang, G.; Wang, C.; Wu, X. Vision-Guided Object Recognition and 6D Pose Estimation System Based on Deep Neural Network for Unmanned Aerial Vehicles towards Intelligent Logistics. *Appl. Sci.* **2022**, *13*, 115. [CrossRef]
15. Li, D.; Sun, X.; Elkhouchlaa, H.; Jia, Y.; Yao, Z.; Lin, P.; Li, J.; Lu, H. Fast detection and location of longan fruits using UAV images. *Comput. Electron. Agric.* **2021**, *190*, 106465. [CrossRef]
16. Wang, G.; Qiu, G.; Zhao, W.; Chen, X.; Li, J. A real-time visual compass from two planes for indoor unmanned aerial vehicles (UAVs). *Expert Syst. Appl.* **2023**, *229*, 120390. [CrossRef]
17. Rueda-Ayala, V.P.; Peña, J.M.; Höglind, M.; Bengochea-Guevara, J.M.; Andújar, D. Comparing UAV-Based Technologies and RGB-D Reconstruction Methods for Plant Height and Biomass Monitoring on Grass Ley. *Sensors* **2019**, *19*, 535. [CrossRef]
18. Cao, Z.; Kooistra, L.; Wang, W.; Guo, L.; Valente, J. Real-Time Object Detection Based on UAV Remote Sensing: A Systematic Literature Review. *Drones* **2023**, *7*, 620. [CrossRef]
19. Marelli, D.; Bianco, S.; Ciocca, G. IVL-SYNTHSFM-v2: A synthetic dataset with exact ground truth for the evaluation of 3D reconstruction pipelines. *Data Brief* **2019**, *29*, 105041. [CrossRef] [PubMed]
20. Jia, Z.; Ouyang, Y.; Feng, C.; Fan, S.; Liu, Z.; Sun, C. A Live Detecting System for Strain Clamps of Transmission Lines Based on Dual UAVs' Cooperation. *Drones* **2024**, *8*, 333. [CrossRef]
21. Ma, Y.; Li, Q.; Chu, L.; Zhou, Y.; Xu, C. Real-Time Detection and Spatial Localization of Insulators for UAV Inspection Based on Binocular Stereo Vision. *Remote. Sens.* **2021**, *13*, 230. [CrossRef]
22. Haque, A.; Elsaharti, A.; Elderini, T.; Elsaharty, M.A.; Neubert, J. UAV Autonomous Localization Using Macro-Features Matching with a CAD Model. *Sensors* **2020**, *20*, 743. [CrossRef] [PubMed]
23. Daramouskas, I.; Meimetis, D.; Patrinopoulou, N.; Lappas, V.; Kostopoulos, V.; Kapoulas, V. Camera-Based Local and Global Target Detection, Tracking, and Localization Techniques for UAVs. *Machines* **2023**, *11*, 315. [CrossRef]
24. Lu, K.; Xu, R.; Li, J.; Lv, Y.; Lin, H.; Liu, Y. A Vision-Based Detection and Spatial Localization Scheme for Forest Fire Inspection from UAV. *Forests* **2022**, *13*, 383. [CrossRef]
25. Arafat, M.Y.; Alam, M.M.; Moh, S. Vision-Based Navigation Techniques for Unmanned Aerial Vehicles: Review and Challenges. *Drones* **2023**, *7*, 89. [CrossRef]
26. Cheng, C.; Li, X.; Xie, L.; Li, L. Autonomous dynamic docking of UAV based on UWB-vision in GPS-denied environment. *J. Frankl. Inst.* **2022**, *359*, 2788–2809. [CrossRef]

27. Li, Z.; Tian, Y.; Yang, G.; Li, E.; Zhang, Y.; Chen, M.; Liang, Z.; Tan, M. Vision-Based Autonomous Landing of a Hybrid Robot on a Powerline. *IEEE Trans. Instrum. Meas.* **2022**, *72*, 1–11. [CrossRef]
28. Chen, C.; Chen, S.; Hu, G.; Chen, B.; Chen, P.; Su, K. An auto-landing strategy based on pan-tilt based visual servoing for unmanned aerial vehicle in GNSS-denied environments. *Aerosp. Sci. Technol.* **2021**, *116*, 106891. [CrossRef]
29. Zhou, Y.; Zhang, D.; Ma, X. Distribution network insulator detection based on improved ant colony algorithm and deep learning for UAV. *iScience* **2024**, *27*, 110119. [CrossRef] [PubMed]
30. Chang, Y.; Cheng, Y.; Manzoor, U.; Murray, J. A review of UAV autonomous navigation in GPS-denied environments. *Robot. Auton. Syst.* **2023**, *170*, 104533. [CrossRef]
31. Feng, S.; Huang, Y.; Zhang, N. An Improved YOLOv8 OBB Model for Ship Detection through Stable Diffusion Data Augmentation. *Sensors* **2024**, *24*, 5850. [CrossRef]
32. Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-scale Dataset for Object Detection in Aerial Images. In Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 3974–3983.
33. Li, R.; Shao, Z.; Zhang, X. Rep2former: A classification model enhanced via reparameterization and higher-order spatial interactions. *J. Electron. Imaging* **2023**, *32*, 053002. [CrossRef]
34. Lloyd, S.P. Least squares quantization in PCM. *IEEE Trans. Inf. Theory* **1982**, *28*, 129–137. [CrossRef]
35. Ren, Z.; Lin, T.; Feng, K.; Zhu, Y.; Liu, Z.; Yan, K. A Systematic Review on Imbalanced Learning Methods in Intelligent Fault Diagnosis. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–35. [CrossRef]
36. Meng, Q.; Zheng, W.; Xue, P.; Xin, Z.; Liu, C. Research and application of CNN-based transmission line hazard identification technology. In Proceedings of the Annual Meeting of CSEE Study Committee of HVDC and Power Electronics (HVDC 2023), Nanjing, China, 22–25 October 2023; pp. 296–300.
37. Liu, J.; Jia, R.; Li, W.; Ma, F.; Wang, X. Image Dehazing Method of Transmission Line for Unmanned Aerial Vehicle Inspection Based on Densely Connection Pyramid Network. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 8857271. [CrossRef]
38. Dong, C.; Zhang, K.; Xie, Z.; Shi, C. An improved cascade RCNN detection method for key components and defects of transmission lines. *IET Gener. Transm. Distrib.* **2023**, *17*, 4277–4292. [CrossRef]
39. Yu, H.; Tian, Y.; Ye, Q.; Liu, Y. Spatial Transform Decoupling for Oriented Object Detection. *arXiv* **2024**, arXiv:2308.10561. [CrossRef]
40. Li, Y.; Hou, Q.; Zheng, Z.; Cheng, M.M.; Yang, J.; Li, X. Large Selective Kernel Network for Remote Sensing Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision 2023, Paris, France, 2–3 October 2023; pp. 16748–16759.
41. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Zhang, S.; Chen, K. RTMDet: An Empirical Study of Designing Real-Time Object Detectors. *arXiv* **2022**, arXiv:2212.07784.
42. Yang, X.; Yang, X.; Yang, J.; Ming, Q.; Wang, W.; Tian, Q.; Yan, J. Learning High-Precision Bounding Box for Rotated Object Detection via Kullback-Leibler Divergence. In Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS), Electr Network, Online, 6–14 December 2021.
43. Xue, Y.; Junchi, Y.; Qi, M.; Wentao, W.; Xiaopeng, Z.; Qi, T. Rethinking Rotated Object Detection with Gaussian Wasserstein Distance Loss. *arXiv* **2021**, arXiv:2101.11952.
44. Jiaming, H.; Jian, D.; Jie, L.; Gui-Song, X. Align deep features for oriented object detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5602511.
45. Xie, X.; Cheng, G.; Wang, J.; Yao, X.; Han, J. Oriented R-CNN for Object Detection. In Proceedings of the 18th IEEE/CVF International Conference on Computer Vision (ICCV), Electr Network, Online, 11–17 October 2021; pp. 3500–3509.
46. Jocher, G.; Chaurasia, A.; Oiu, J. YOLO by Ultralytics. Available online: https://github.com/ultralytics/ultralytics (accessed on 4 July 2024).
47. Ruiz, F.; Arrue, B.C.; Ollero, A. SOPHIE: Soft and Flexible Aerial Vehicle for Physical Interaction with the Environment. *IEEE Robot. Autom. Lett.* **2022**, *7*, 11086–11093. [CrossRef]
48. Zufferey, R.; Barbero, J.T.; Talegon, D.F.; Nekoo, S.R.; Acosta, J.A.; Ollero, A. How ornithopters can perch autonomously on a branch. *Nat. Commun.* **2022**, *13*, 7713. [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.