applied sciences

MDPI

Article

Hand–Eye Separation-Based First-Frame Positioning and Follower Tracking Method for Perforating Robotic Arm

Handuo Zhang [1], Jun Guo [2,*], Chunyan Xu [2] and Bin Zhang [2]

[1] College of Computer Science and Engineering, Northeastern University, Shenyang 110169, China
[2] Software College, Northeastern University, Shenyang 110169, China
* Correspondence: guojun@mail.neu.edu.cn

Abstract: In subway tunnel construction, current hand–eye integrated drilling robots use a camera mounted on the drilling arm for image acquisition. However, dust interference and long-distance operation cause a decline in image quality, affecting the stability and accuracy of the visual recognition system. Additionally, the computational complexity of high-precision detection models limits deployment on resource-constrained edge devices, such as industrial controllers. To address these challenges, this paper proposes a dual-arm tunnel drilling robot system with hand–eye separation, utilizing the first-frame localization and follower tracking method. The vision arm ("eye") provides real-time position data to the drilling arm ("hand"), ensuring accurate and efficient operation. The study employs an RFBNet model for initial frame localization, replacing the original VGG16 backbone with ShuffleNet V2. This reduces model parameters by 30% (135.5 MB vs. 146.3 MB) through channel splitting and depthwise separable convolutions to reduce computational complexity. Additionally, the GIoU loss function is introduced to replace the traditional IoU, further optimizing bounding box regression through the calculation of the minimum enclosing box. This resolves the gradient vanishing problem in traditional IoU and improves average precision (AP) by 3.3% (from 0.91 to 0.94). For continuous tracking, a SiamRPN-based algorithm combined with Kalman filtering and PID control ensures robustness against occlusions and nonlinear disturbances, increasing the success rate by 1.6% (0.639 vs. 0.629). Experimental results show that this approach significantly improves tracking accuracy and operational stability, achieving 31 FPS inference speed on edge devices and providing a deployable solution for tunnel construction's safety and efficiency needs.

Keywords: hand–eye separation; initial frame positioning; object detection; object tracking

Academic Editor: Suchao Xie

Received: 4 February 2025
Revised: 26 February 2025
Accepted: 28 February 2025
Published: 4 March 2025

Citation: Zhang, H.; Guo, J.; Xu, C.; Zhang, B. Hand–Eye Separation-Based First-Frame Positioning and Follower Tracking Method for Perforating Robotic Arm. *Appl. Sci.* **2025**, *15*, 2769. https://doi.org/10.3390/app15052769

1. Introduction

The advancement of science and technology has facilitated the growth of artificial intelligence and machine vision technology, thereby enabling the development of a vision-guided robot drilling control system [1,2]. The integration of machine vision technology enables tunnel drilling robots to achieve precise positioning and efficient hole drilling, thereby markedly enhancing construction efficiency and quality while satisfying the rigorous standards of modern tunnel construction for superior quality and efficiency [3–5]. Nevertheless, the extant hand–eye integration tunnel drilling robot continues to encounter numerous challenges in practical deployment. The camera is typically situated within the robotic arm used for drilling, which is vulnerable to the disruption caused by dust within the tunnel during the drilling process. This results in a deterioration of image acquisition quality, consequently affecting the stability and accuracy of the visual recognition system.

the lengthy nature of the tunnel project necessitates the drilling of numerous holes within the tunnel area. The conventional approach of utilizing a robotic arm to repeatedly identify and locate these holes is not only time-consuming but also increases the cost and complexity of the operation.

To overcome the aforementioned challenges, a hand–eye separation-based tunnel drilling robot hole location localization system was constructed, based on the first-frame localization and follower tracking method of the drilling robot arm. The optimal configuration for a hand–eye cooperative operation is achieved by mounting a dual robotic arm system with a vision arm (i.e., "eye") and a perforating robot arm (i.e., "hand") on the same mounting surface. The vision robot arm is tasked with acquiring the positional data of the punching robot arm in real-time, thereby enabling the precise tracking and tracing of the punching robot arm through the utilization of a high-precision follower adjustment mechanism. The schematic diagram is illustrated in Figure 1.

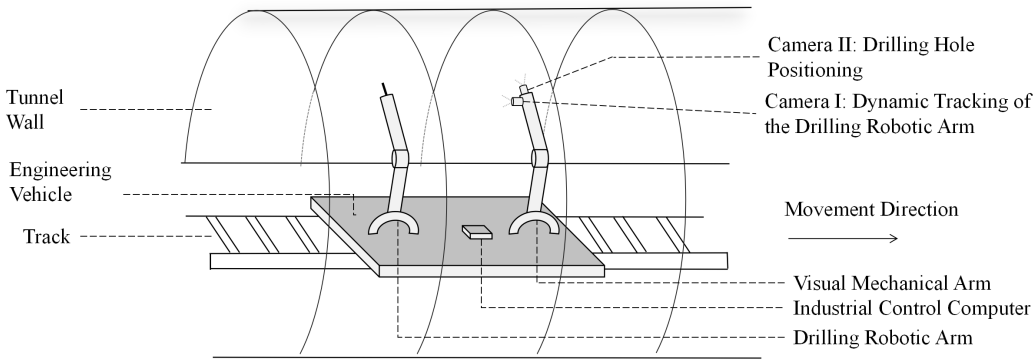


Figure 1. Hand–eye separation schematic.

In the hole localization system of a tunnel boring robot, the accuracy of the initial frame localization has a significant influence on the subsequent follow-up tracking process. An error in the initial frame localization may fail in the subsequent tracking process, thereby negatively impacting the overall quality of the task completion. Additionally, it should be noted that the tunnel drilling robot hole positioning system, which is based on hand–eye separation, must be installed on the edge intelligent terminal industrial control machine. This results in a limitation of system resources. It is therefore important to minimize the computational complexity, memory occupation, and energy consumption problems while ensuring the positioning accuracy of the first frame. To guarantee the precision of identifying and positioning the perforated robotic arm, this study employs a deep learning-based target detection algorithm, RFBNet [6], which modifies its network structure and loss function to enhance the accuracy and efficiency of the first-frame localization. Furthermore, in the context of continuous tracking in complex environments, the tracked target frequently encounters the challenge of occlusion, which may result in the drift phenomenon of the tracker and consequently impair the accuracy and stability of continuous tracking. Furthermore, during the follow-along tracking process of the perforating robotic arm, the vision robotic arm may encounter issues such as target loss when tracking the target for an extended period. Consequently, this study employs the twin network-based SiamRPN [7] tracking and localization method of the perforating robot arm to achieve real-time updating and accurate tracking of the perforating robot arm's position information. Furthermore, the PID follower control algorithm is utilized to facilitate precise adjustment and control of the visual robot arm by the position information obtained from the tracking and positioning. The proposed design not only improves the operational efficiency of the system but also significantly enhances its stability and reliability.

2. Literature Review

2.1. Target Detection

The principal objective of perforating robot target detection techniques is to employ machine vision systems for the precise identification and localization of target objects. The initial target detection methods, including R-CNN [8], Fast R-CNN [9], and Faster R-CNN [10], employ region proposals to identify potential regions for consideration, subsequently performing classification and regression. Despite their enhanced performance, these methods are characterized by elevated computational costs, rendering them unsuitable for real-time applications. To address the issue of real-time performance, Redmon et al. [11] introduced YOLO (You Only Look Once) in 2016. This innovation streamlines target detection into a regression task by leveraging a single convolutional neural network to predict bounding boxes and category probabilities directly over the entire image, thereby negating the necessity for prior region screening. This approach allows for a comprehensive consideration of image information, thereby reducing background misclassification and enhancing detection accuracy. Subsequently, the team proceeded to refine YOLO with several subsequent iterations [12,13], which markedly enhanced the algorithm's accuracy and robustness. Nevertheless, it remained inferior to the two-stage detector in terms of localization precision. In the same year, Liu W et al. [14] proposed SSD (Single-Shot MultiBox Detector), which also achieves rapid and highly accurate detection without the generation of candidate regions. The SSD is capable of responding flexibly to the detection of objects of varying sizes, ranging from large to small. This is achieved through the extraction of features from different layers of the network, which enhances the comprehensiveness and accuracy of the detection process. In SSD, multiple sets of predefined frames are allocated to each feature map unit, with varying dimensions and scales. These are designed to match and predict objects of corresponding sizes. To optimize the performance of the single-stage detector, particularly in the context of multi-scale targets, it is evident that SSD still exhibits shortcomings in the detection of small targets, the handling of complex backgrounds, and the accuracy of prediction. Despite these shortcomings, SSD exhibits notable advantages in real-time target detection. To enhance the performance of single-stage detectors, particularly when dealing with multi-scale targets, RFBNet [6] introduces an innovative Receptive Field Block (RFB) based on SSD. This aims to augment the network's capacity to detect multi-scale targets while maintaining high processing speed and efficiency to meet real-time requirements.

2.2. Target Tracking

The tracking and localization of perforated robotic arms are primarily dependent on sophisticated methodologies within the domain of deep learning, with a particular emphasis on the architectural configuration of twin networks, more specifically Siamese Networks. Twin network trackers have attracted considerable attention due to their capacity to maintain high tracking accuracy while meeting real-time performance requirements. In 2016, Luca B. et al. proposed SiamFC (Siamese Fully Convolutional Network) [15], which learns relevant similarities.

The measurement function is conducted through a pre-trained deep twin network that passes through a full convolutional layer to evaluate the similarity between targets and potential candidates. This is designed to allow flexible processing of templates and search images of different sizes; however, it is less robust to complex scene processing such as significant appearance changes or occlusion of targets. Furthermore, due to the lack of an online update mechanism in SiamFC, significant changes in target appearance may result in performance degradation over time. In 2018, LIB et al. proposed SiamRPN (Siamese Region Proposal Network) [7], which integrates a region proposal network (Region Pro-

posal Network (RPN)) on top of the base twin network structure. This improvement predicts the target location with greater accuracy and significantly enhances the tracking performance in situations such as target size change and occlusion. Nevertheless, SiamRPN is also deficient in terms of online learning mechanisms, and the long-term stability of tracking performance remains a significant challenge. In the same year, Zheng Z. et al. proposed DaSiamRPN [16], which introduced a mechanism based on SiamRPN that was designed to enhance the model's ability to distinguish between targets and non-targets (interference terms), a process that the authors refer to as "distractor-aware". However, the complex structure of the DaSiamRPN model and its reliance on a substantial quantity of labeled data to identify interfering items restrict its applicability in data-scarce perforated robotic arm scenarios. Despite these algorithms demonstrating considerable advancement in real-time performance and accuracy, further research and optimization are still necessary to achieve enhanced performance and stability when confronted with intricate industrial environments.

3. Materials and Methods

3.1. First-Frame Localization Model Based on Improved RFBNet

3.1.1. Network Infrastructure

The objective of this paper is to optimize the RFBNet-based first-frame localization model through the introduction of techniques derived from ShuffleNet V2 [17] and GIoU [18]. This is to enhance the operational efficiency of the first-frame localization model and the accuracy of the bounding box of the first-frame localization on the industrial control machine. The network structure of the first-frame localization model, which has been enhanced through the application of the aforementioned techniques, is illustrated in Figure 2.

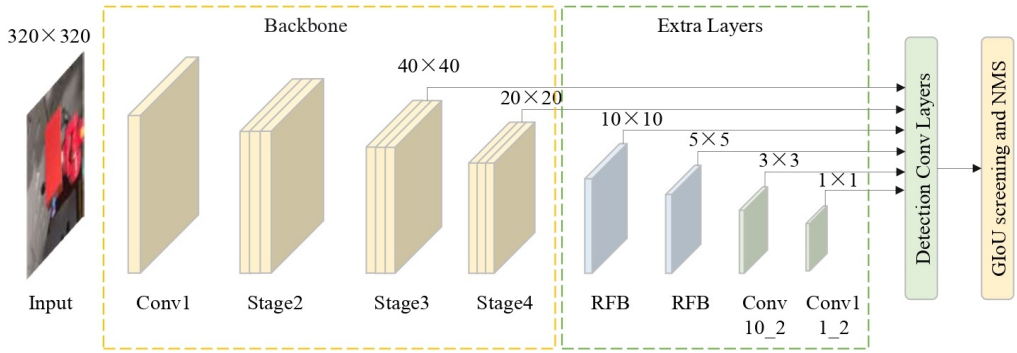
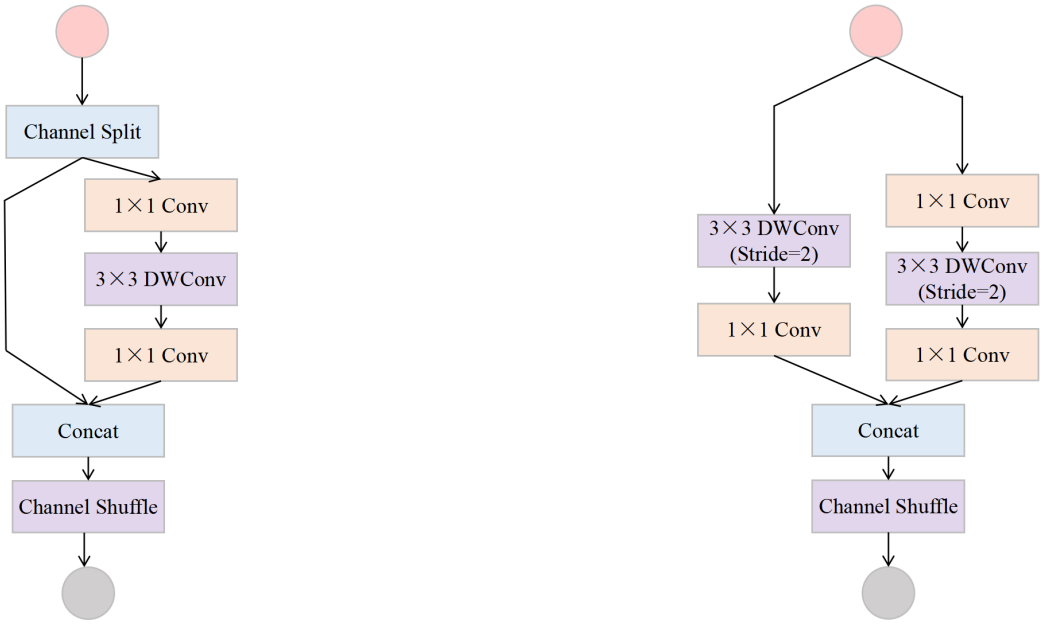


Figure 2. Network structure of initial frame positioning model based on improved RFBNet. Schemes follow the same formatting.

The ShuffleNet V2 network offers several advantages, including a reduced number of network parameters, low computational complexity, rapid operation speed, and high recognition accuracy. Accordingly, the lightweight ShuffleNet V2 network is employed instead of the original VGG16 backbone network, thereby reducing the computational complexity through the introduction of a channel-splitting strategy. This strategy divides the channels of the perforated robotic arm feature map into two parts: direct output and convolutional processing. Following the rearrangement of the convolutionally processed channels, distinct groups of features are combined, thereby enhancing the model's capacity for characterization. Furthermore, point-by-point convolution and depth-separable convolution are employed in ShuffleNet V2 to reduce the number of parameters and computations. Point-by-point convolution can be divided into two categories. The first is

employed to augment the number of channels in the feature map, thereby enabling the model to capture a more nuanced array of information. The second is utilized to reduce the number of channels to the original size, which effectively compresses the information and reduces the number of parameters. Following point-by-point convolution, each input channel is initially convolved channel-by-channel using depth-separable convolution (DWConv). Subsequently, the channel information is mixed by point-by-point convolution. This structure enables ShuffleNet V2 to markedly reduce computational resource consumption while maintaining high performance, thus adapting to the resource constraints of the industrial controllers of the edge intelligent terminals. The structure diagram is shown in Figure 3.



(**a**) Meta (**b**) ShuffleNet

Figure 3. ShuffleNet V2 block.

To enhance the efficacy of the RFBNet first-frame localization model, we optimized the backbone network with four stages, namely, Conv1, Stage 2, Stage 3, and Stage 4, of the ShuffleNet V2 network. In light of the potential impact of the maximum pooling layer on the receptive domain of the feature map, we have elected to remove the initial maximum pooling layer by the specifications of ShuffleNet V2.

In the optimized backbone network, the processing flow of the perforated robotic arm image is as follows: firstly, the Conv1 layer applies a 3 × 3 convolution kernel to the input perforated robotic arm image with a step size of 2, thus effectively reducing the size of the input image to half of the original. Subsequently, the images from Stages 2, 3, and 4 are subjected to feature extraction and transformation through the use of multiple ShuffleNet V2 blocks. In particular, Stage 2 comprises one ShuffleNet V2 block (b) and three ShuffleNet V2 blocks (a). The combination of different types of blocks enables the model to capture both local details and global information about the image. Similarly, Stage 3 contains one ShuffleNet V2 block (b) and seven ShuffleNet V2 blocks (a) for further feature extraction and enhancement, while Stage 4 contains one ShuffleNet V2 block (b) and three ShuffleNet V2 blocks (a) to provide a rich feature representation for the final first frame localization.

From Conv1 to Stage 4, the size of the input image is reduced by a factor of 16 through downsampling, which enables the model to capture the essential information in the image while maintaining a certain level of computational efficiency. Through a series of convolution and block stacking operations, the optimized RFBNet first-frame

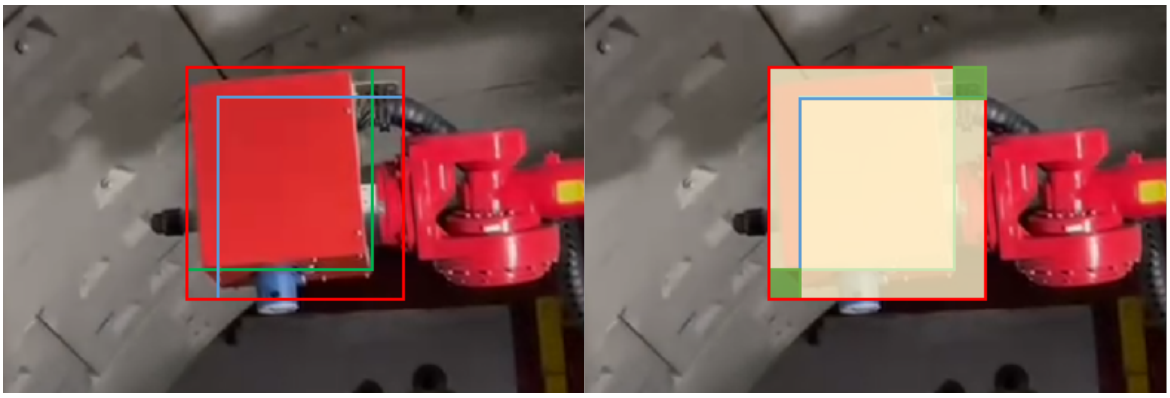
localization model can more accurately determine the position of the perforating robot arm in the image.

To enhance the model's capacity to comprehensively understand the perforated robotic arm, from local texture to overall semantics, we adopt the strategy of detecting target objects from multi-scale feature maps. In total, six layers of features are extracted from the deep structure of the backbone network, which encompasses a comprehensive range of information, from the minutiae to the global information. In particular, the middle- and high-level features are captured by utilizing Stages 3 and 4 of the backbone network, which correspond to feature maps of sizes 40 × 40 and 20 × 20, respectively. The aforementioned feature maps contain more detailed spatial information about the perforated robotic arm and certain semantic information, which is of great importance for the process of localizing the target object. Subsequently, to extract higher-level semantic features, two RFB (Receptive Field Block) modules are introduced following Stage 4. The RFB modules capture broader contextual information through carefully designed convolution and pooling operations. These modules use dilated convolutions with different dilation rates, effectively expanding the receptive field and enabling the model to capture multi-scale contextual information. After processing by the RFB modules, we obtained two smaller but semantically richer feature maps, sized 10 × 10 and 5 × 5. This multi-scale feature aggregation compensates for the resolution degradation caused by the lightweight backbone design, preserving fine local details while enhancing the expression of high-level semantic features. Finally, to further enhance the discriminative power and adaptability of the features, two additional convolution layers are added at the end of the network, namely, Conv10-2 and Conv11-2. The function of Conv10-2 is to further fuse local spatial information, while the function of Conv11-2 is to compress the number of channels and generate the final feature representation through a 1 × 1 convolution layer. The multi-scale feature fusion and progressive deepening of the network structure enable our model to comprehensively comprehend the local texture and overall semantics of the perforating robotic arm, thereby facilitating more accurate tracking and localization.

The conventional IoU is constrained in its ability to address scenarios where the bounding boxes exhibit minimal or no overlap. This limitation hinders its capacity to accurately represent the disparities in the position and dimensions between the two boxes. In contrast, GIoU evaluates the similarity between two boxes more comprehensively by introducing the concept of an enclosing box. This takes into account not only the overlapping regions of the predicted and real bounding boxes but also the non-overlapping regions between them. To enhance the precision of the target bounding box prediction for the perforated robotic arm, this study employs GIoU (Generalized Intersection over Union) as an evaluation metric, utilizing the RFBNet model in place of the conventional IoU (Intersection over Union). During the training process, GIoU is employed as a criterion for the selection of positive and negative samples about the predicted bounding box. Should the GIoU value of the predicted and actual bounding boxes exceed the pre-established positive sample threshold, the former is deemed a positive sample, thereby contributing to the positive training of the model. Conversely, if the GIoU value is below the negative sample threshold, the predicted bounding box is classified as a negative sample, which is employed for the negative training of the model.

The GIoU schematic is shown in Figure 4.

As illustrated in Figure 4a, the red box represents the smallest outer bounding box that encompasses both the prediction box and the true box. In Figure 4b, the green shading denotes the area of the smallest outer bounding box, minus the concatenation of the true box and the prediction box.



(**a**) GIoUa (**b**) GIoUb

Figure 4. Comparison of tracking results of the proposed method with the baseline algorithm and the classical algorithm.

3.1.2. Loss Function

To enhance the efficacy of the model and augment the performance of the lightweight model for first-frame localization, we pursue optimization based on the SSD loss function. The expression is as follows:

$$L(x, c, l, g) = \frac{1}{N}(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)) \tag{1}$$

where $L(x, c, l, g)$ denotes the loss function, N denotes the number of matches to the default box, x denotes the input sample, c denotes the confidence level, l denotes the prediction box, g denotes the true box, and a denotes the weight coefficients for classification and regression.

The loss function is the sum of the confidence loss and the position loss. The confidence loss is $L_{\text{conf}}(x, c)$. To make the perforation machine arm more distinguishable from the tunnel background, the focal loss is introduced and the positive and negative sample weight coefficients are increased. The focal loss formula is

$$FL(p_t) = -\alpha_t(1 - p_t)^{\gamma} \log(p_t) \tag{2}$$

Confidence losses are specified as follows:

$$L_{\text{conf}}(x, c) = -\sum_{i \in \text{Pos}}^{N} \omega_p x_{ij}^p (1 - \hat{c}_i^p)^{\gamma} \log(\hat{c}_i^p) - \beta \sum_{i \in \text{Neg}} \log(\hat{c}_i^0) \tag{3}$$

$$\hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \tag{4}$$

where $L_{\text{conf}}(x, c)$ is the confidence loss, $\hat{c}_i^p$ is the probability value of the category predicted by the first prediction box, c_i^p is the confidence prediction value of the category predicted by the first prediction box, x_{ij}^p is the match value of the first prediction box and the first true box of the category p, where taking 1 means that the first prediction box matches the first true box and taking 0 means that it does not match, i is the serial number of the prediction box, j is the serial number of the true box, p is the serial number of the category, Pos is a positive sample, Neg is a negative sample, ω_p is the weight of the category p, and β is the weight coefficient between positive and negative samples.

The position loss is $L_{\text{loc}}(x, l, g)$. To increase the sensitivity of the punching machine arm target frame to errors and to more accurately locate the position of the punching

machine arm target frame, the position loss is rooted for the width and height values of the prediction frame:

$$L_{\text{loc}}(x,l,g) = \sum_{i\in\text{Pos}}^{N} x_{ij}^{p}\Bigg(\begin{aligned}&\text{smooth}_{L1}(l_i^{cx} - \hat{g}_j^{cx}) \\ &+ \text{smooth}_{L1}(l_i^{cy} - \hat{g}_j^{cy}) \\ &+ \text{smooth}_{L1}\left(\sqrt{w_i^l} - \sqrt{\hat{g}_j^{w}}\right) \\ &+ \text{smooth}_{L1}\left(\sqrt{h_i^l} - \sqrt{\hat{g}_j^{h}}\right)\Bigg)\end{aligned} \tag{5}$$

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{6}$$

$$\hat{g}_j = \begin{bmatrix} \hat{g}_j^{cx} \\ \hat{g}_j^{cy} \\ \hat{g}_j^{w} \\ \hat{g}_j^{h} \end{bmatrix} = \begin{bmatrix} \left(\frac{g_j^{cx} - d_i^{cx}}{d_i^{w}}\right) \\ \left(\frac{g_j^{cy} - d_i^{cy}}{d_i^{h}}\right) \\ \log\left(\frac{g_j^{w}}{d_i^{w}}\right) \\ \log\left(\frac{g_j^{h}}{d_i^{h}}\right) \end{bmatrix} \tag{7}$$

where $L_{\text{loc}}(x,l,g)$ denotes the positional loss, l_i^{cx} denotes the horizontal coordinate of the center of the i-th prediction box, l_i^{cy} denotes the vertical coordinate of the center of the i-th prediction box, g_j^{cx} denotes the horizontal coordinate of the center of the j-th truth box, g_j^{cy} denotes the vertical coordinate of the center of the j-th truth box; $\hat{g}_j^{cx}$ denotes the coded value of the horizontal coordinate of the center of the j-th truth box, $\hat{g}_j^{cy}$ denotes the coded value of the vertical coordinate of the center of the j-th truth box; w_i^l denotes the width of the i-th prediction box, h_i^l denotes the height of the i-th prediction box; g_j^{w} denotes the width of the j-th truth box, g_j^{h} denotes the height of the j-th truth box; $\hat{g}_j^{w}$ denotes the coded value of the width of the j-th truth box, $\hat{g}_j^{h}$ denotes the coded value of the height of the j-th truth box; d_i^{w} denotes the width of the i-th a priori box, and d_i^{h} denotes the height of the i-th a priori box.

3.2. SiamRPN-Based Tracking and Localization Method for Punching Robot Arm

In the perforated robot arm tracking and localization study, we selected the deep learning-based SiamRPN algorithm [7] as the core framework, which combines the power of twin networks for perforated robot arm feature extraction and the precise efficiency of region proposal networks for perforated robot arm localization.

The twin network contains two branches, each with the same structure and weight. Its main purpose is to compute the similarity of the two input perforated robot arm images. In the context of perforated robot arm tracking, one branch processes the perforated robot arm samples in the initial (or previous) frame (known as the template image), while the other branch processes the current frame (known as the search image). With these two branches, the twin network can extract the feature representations of the template image and the search image. Next, a cross-correlation operation is performed on the feature map of the search image using the feature map of the template image to generate a response map. This response map encodes the similarity of each location in the search image to the template image, with higher response values corresponding to regions that are more similar to the template image.

We overlay the twin network with the Region Proposal Network (RPN). The RPN quickly generates high-quality target proposal frames in the input image (in this case, the feature map of the search image). The RPN uses a series of fixed anchor frames (anchors), which are distributed throughout the feature map and have different sizes and aspect ratios. The RPN outputs two results for each anchor frame: a score (confidence level) that the anchor box contains the perforated robot arm, and four real values (bounding box regression parameters) that fine-tune the anchor box to better fit the perforated robot arm target. In SiamRPN, RPN has two output headers: a classification header, which is used to determine whether a particular anchor frame contains a perforated robot arm, and a regression header, which is used to fine-tune the position and size of the anchor frame to better fit the perforated robot arm target.

To solve the problem of tracking accuracy and stability of the perforating robot arm during the tracking process, we predict the next frame position of the perforating robot arm using the target prediction method based on Kalman filtering [19] and predict the motion trend of the perforating robot arm by the relocation method based on the lost judgment and target detection. The process is shown in Figure 5.

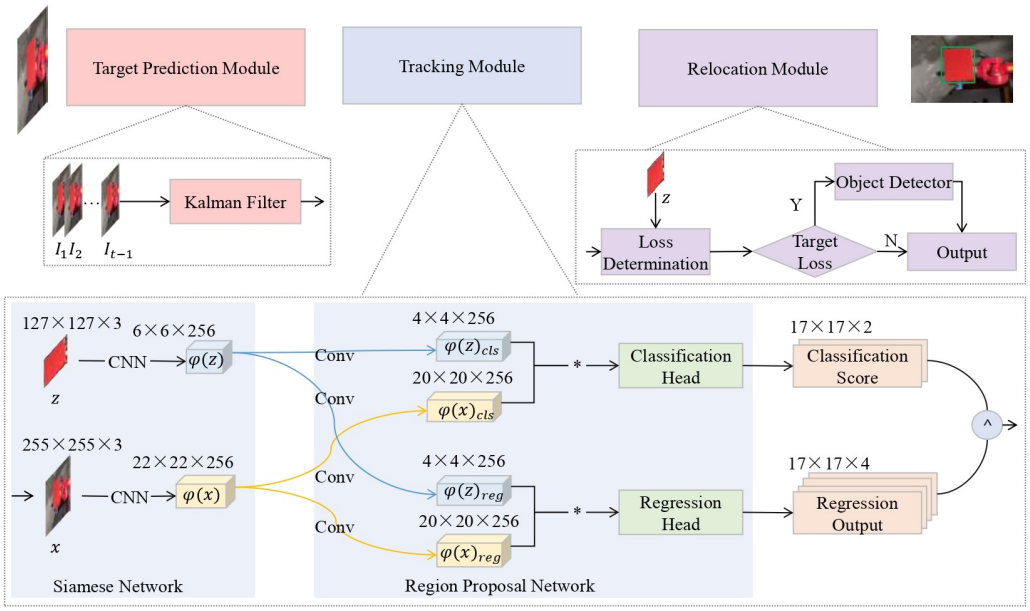


Figure 5. Tracking and positioning process diagram of drilling robot arm based on SiamRPN.

3.2.1. Target Prediction Algorithm Based on Kalman Filtering

To improve the accuracy of tracking results in occlusion scenarios, this study proposes a target prediction algorithm based on Kalman filtering as detailed in Algorithm 1. Firstly, the Kalman filter is used to predict the motion trajectory of the perforating robot arm, and secondly, the results of the Kalman filter are used to optimize the positioning of the perforating robot arm and to correct the selection of the search area of the perforating robot arm. The Kalman filter predicts the possible position of the perforating robot arm in the next frame, which helps to reduce the search space that SiamRPN has to handle, which in turn improves performance and optimizes the allocation of computational resources.

The prediction step of Kalman filtering can predict the next instantaneous state based on the current estimated state (position and velocity) of the perforating robot arm. This prediction can provide an expected target position of the perforating robot arm, which helps SiamRPN to search for the perforating robot arm more accurately in subsequent frames. Whenever SiamRPN locates the robot arm in a new frame, it receives an observation that can be used in the Kalman filter update step, where the Kalman filter combines the previous

prediction with the current observation to generate a more accurate state estimate of the robot arm. The Kalman filter calculates the error of a current state estimate and a gain (called the Kalman gain) that allows a trade-off between prediction and observation to correct the state of the robot arm.

The state vector of the perforating machine arm can be expressed as $\mathbf{x} = [x, y, v_x, v_y]$, where x denotes the position of the perforating machine arm in the x direction, y denotes the position of the perforating machine arm in the y direction, v_x denotes the speed of the perforating machine arm in the x direction, and v_y denotes the speed of the perforating machine arm in the y direction. The equation of state of the perforating machine arm is as follows:

$$X_k = F_k X_{k-1} + e_k \tag{8}$$

$$F_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

$$e_k \sim N(0, E) \tag{10}$$

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

where X_{k-1} denotes the position of the perforating machine arm and the speed of the perforating machine arm at the instant $k-1$, X_k denotes the position of the perforating machine arm and the speed of the perforating machine arm at the instant k, F_k denotes the state transfer matrix of the system from the instant $k-1$ to the instant k, e_k denotes the state noise vector of the system which satisfies the normal distribution $N(0, E)$, and E denotes the covariance matrix of the state noise during the movement of the perforating machine arm. At present, the state equation of the perforating machine arm is as follows:

$$\begin{bmatrix} x(k) \\ y(k) \\ v_x(k) \\ v_y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1) \\ y(k-1) \\ v_x(k-1) \\ v_y(k-1) \end{bmatrix} + e_k \tag{12}$$

The observation vector can be expressed as $\mathbf{x} = [x, y]$, where x denotes the observation position of the perforating robot arm in the direction, y denotes the observation position of the perforating robot arm in the y direction, and the observation equation of the perforating robot arm is as follows:

$$Z_k = H_k X_k + m_k \tag{13}$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{14}$$

$$m_k \sim N(0, M) \tag{15}$$

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{16}$$

where Z_k denotes the observed position of the instantaneous punch press arm, H_k denotes the observation matrix, m_k denotes the observation noise vector satisfying the normal distribution $N(0, M)$, and M denotes the covariance matrix of the observation noise during

the movement of the punch press arm. The observation equation of the instantaneous punch press arm is as follows:

$$Z_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ y(k) \\ v_x(k) \\ v_y(k) \end{bmatrix} + m_k \tag{17}$$

The prediction equation for the state of the perforating machine arm is as follows:

$$\hat{X}_k^- = F_{k-1}\hat{X}_{k-1} \tag{18}$$

where $\hat{X}_{k-1}$ denotes the instantaneous state estimate and $\hat{X}_k^-$ denotes the instantaneous prediction of the state of the punch press arm.

The prediction error covariance matrix is calculated as follows:

$$P_k^- = F_{k-1}P_{k-1}F_{k-1}^T + E \tag{19}$$

where P_{k-1} is the moment-to-moment error covariance matrix estimate and P_k^- is the moment-to-moment error covariance matrix prediction.

The Kalman gain matrix is calculated as follows:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + M)^{-1} \tag{20}$$

The estimated value of the state of the punching machine arm at the moment k is calculated as follows:

$$\hat{X}_k = \hat{X}_k^- + K_k(Z_k - H_k\hat{X}_k^-) \tag{21}$$

The estimation error covariance matrix is calculated by the following procedure:

$$P_k = (I - K_k H_k)P_k^- \tag{22}$$

where I is the unit matrix.

Algorithm 1 Target Prediction Algorithm Based on Kalman Filter

Input: Bounding frame $\hat{l}_0$ of the perforated robot arm labeled in frame 1.
Output: Predicted perforated robot arm bounding box $\hat{l}_{k-1}$ in frame k.

Initialize the Kalman filter:
    Set the initial state vector $\mathbf{x} = [x, y, v_x, v_y]$ based on the bounding box $\hat{l}_0$ in frame 1.
    Set the initial observation vector $\mathbf{z} = [x, y]$.
for $i = 2, 3, \ldots, k$ **do**
    (1) Predict the state of the perforated robot arm in frame $i - 1$:
        Compute the predicted state $\hat{\mathbf{x}}_k^-$ and the prediction error covariance matrix P_k^-.
    (2) Update the Kalman filter based on the observation in frame k:
        Calculate the observation $\mathbf{z}_k$ and the Kalman gain K_k.
    (3) Refine the state and covariance estimates:
        Compute the state estimate $\hat{\mathbf{x}}_k$ and the error covariance P_k.
    (4) Obtain the predicted bounding box $\hat{l}_k$ from the state estimate $\hat{\mathbf{x}}_k$.
end for

3.2.2. Re-Localization Method for Punching Robotic Arm Based on Target Detection

By combining the tracking ability of SiamRPN and the prediction ability of Kalman filtering, useful dependencies can be established between different frames to improve the overall tracking performance of the perforating robot arm. However, during the whole

tracking process of the perforating robot arm, due to the long tracking time and the large rotation of the robot arm, the target loss of the perforating robot arm will occur, resulting in the tracking interruption. This study proposes a re-localization method for perforating robot arms based on target detection to improve continuous tracking stability. Firstly, the tracking is based on the SiamRPN algorithm and Kalman filtering; secondly, the current perforating robot arm target is lost based on the loss detection module before the end of each frame—if the perforating robot arm is not lost, the perforating robot arm position is output to continue the tracking, and if it is determined that the perforating robot arm target is lost, the target detection module is based on the target detection module to re-detect and localize the perforating robot arm position in the global image.

In the context of target detection, this study employs a first-frame localization model based on an enhanced RFBNet approach. Regarding loss determination, this study incorporates a target loss determination mechanism. If no discernible peak is present in the response map, the tracker assigns a score of 0 to the current frame, indicating a perforated robot arm target tracking failure. Consequently, the current frame is designated as being in a perforated robot arm target loss state. The judgment strategy is as follows:

$$F = \begin{cases} \max(f(z,s)), & f(z,s) > \alpha \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

$$f(z,s) = \varphi(z) * \varphi(s) + b_1 \tag{24}$$

where F denotes the maximum response value, $f(z,s)$ denotes the feature comparison formula, z denotes the perforated robotic arm target template, s denotes the current frame, b_1 denotes the bias term for the values taken in the score map, and φ denotes the feature extraction method.

3.3. PID-Based Control Method for Punching Machine Arm Follower

In practice, a soft-body robotic arm is employed for the vision robotic arm, the tracking robotic arm, and the punching robotic arm. However, the jittering of the robot arm, the inertia of the motion, and the effects of gravity and elasticity in the soft robot arm may result in the robot arm moving away from or failing to maintain an accurate following position. It is therefore necessary to implement a control method to achieve the desired control effect. In the control concept, the PID controller is capable of comparing the current actual speed with the desired control speed in real-time, and making corresponding adjustments according to the differences, to achieve the optimal control effect. The PID control comprises three components, namely, Proportional, Integral, and Derivative. The PID control adjusts the control amount to reduce the deviation between the current state of the system and the target state by adjusting the control quantity. The PID control model is illustrated in Figure 6.

The proportional term modulates the change in the control amount by the magnitude of the current deviation. Consequently, a larger deviation results in a larger adjusted control amount, thereby facilitating the expeditious reduction in the initial deviation in the position of the robotic arm. However, it is challenging to eradicate the deviation through the use of proportional control alone. The integral term considers the cumulative impact of deviations, whereby a persistent deviation results in further adjustments to the control amount to eliminate long-term steady-state errors. This helps to eliminate the static errors of the system and ensures the accuracy of the robotic arm's follower position. However, it may also lead to over-adjustment and oscillations in the system. The differential term predicts the future trend of the deviation and makes control adjustments by taking into account the rate of change of the deviation. This helps to reduce or eliminate overshooting

of the system and improve the speed of response, reduce vibration at the beginning and end of the robotic arm follower, and optimize the smoothness and accuracy of the robotic arm follower process.

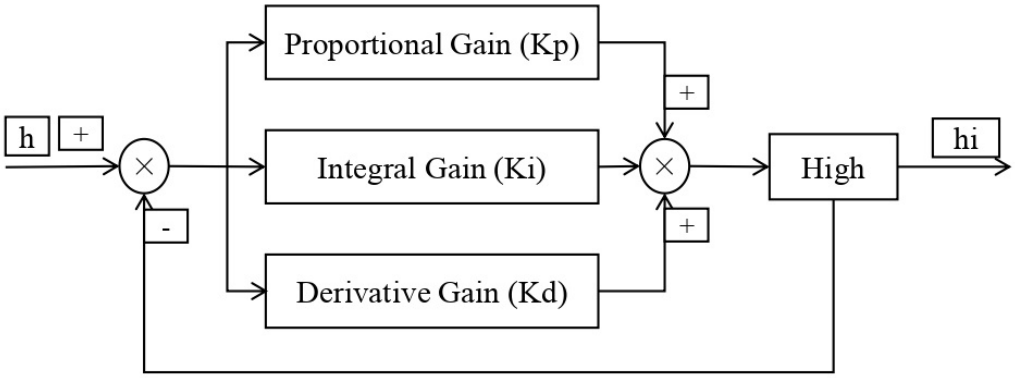


Figure 6. PID control system block diagram.

To reduce the deviation during the rotation of the flexible robotic arm, it is necessary to reduce the jitter when the flexible robotic arm rotates and stops. Furthermore, the influence of gravity and the elasticity of the flexible robotic arm on the rotation position of the flexible robotic arm must be reduced. A control algorithm has been designed for this purpose:

$$u(t) = \begin{cases} K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de(t)}{dt} \\ +K_g \sin(e(t)) + K_f \left(\frac{1}{1-e(t)} - 1\right), & e(t) < 0 \\ K_p e(t) + K_i \int_0^t e(t)\,dt + K_d \frac{de(t)}{dt} \\ +K_g \sin(e(t)) + K_f \left(\frac{1}{e(t)+1} - 1\right), & e(t) \geq 0 \end{cases} \tag{25}$$

where $u(t)$ denotes a control function, $e(t)$ denotes a coordinate deviation of the center of the target of the perforating robotic arm from the center point of the image of the current frame in one dimension, t denotes a discrete time, and K_p, K_i, K_d, K_g, and K_f are the proportional parameter, integral parameter, differential parameter, sinusoidal parameter, and power parameter, respectively.

When the coordinates of the center point of the target of the perforating robot arm deviate from the center point of the image, the flexible robot arm is affected by two factors: the weight of the flexible robot arm itself and the elasticity of the flexible arm's central flexible shaft in the process of rotation. To ensure greater accuracy in the rotation of the flexible robot arm during the tracking of the target, it is essential to take into account the combined effect of the weight of the flexible robot arm and the elasticity of the flexible arm's central flexible axis on the rotation of the robot arm. It is also necessary to consider the weight of the flexible robot arm and the elasticity of the flexible arm's central axis on the rotation of the robot arm. Therefore, these three terms $K_g \sin(e(t))$, $K_f \left(\frac{1}{1-e(t)} - 1\right)$, and $K_f \left(\frac{1}{e(t)+1} - 1\right)$ are introduced to correct for the effects of the flexible robot arm's gravity and the flexible arm's central flexible axis.

When the soft-body mechanical arm rotates downward, the gravity of the soft-body mechanical arm itself will lead to the actual angle of rotation of the soft-body mechanical arm being greater than the actual angle of rotation needed. When the soft-body mechanical arm rotates upward, the gravity of the soft-body mechanical arm itself will instead lead to the actual angle of rotation of the soft-body mechanical arm being less than the actual angle of rotation needed. $\sin(e(t))$ is used to indicate the changing trend of the actual angle of

rotation needed in the process of the soft-body mechanical arm rotation and the angle of deviation from the actual rotation of the soft-body mechanical arm. When $0 \leq e(t) \leq 1$, that is, when the soft-body mechanical arm rotates downward, $e(t)$ and $\sin(e(t))$ are positively correlated. When $-1 \leq e(t) < 0$, that is, when the soft-body mechanical arm rotates upward, $e(t)$ and $\sin(e(t))$ are negatively correlated.

Since the center flexible shaft of the flexible mechanical arm has a certain elasticity, when the flexible mechanical arm rotates downward, the elasticity of the center flexible shaft will cause the actual rotation angle of the flexible mechanical arm to be smaller than the actual required rotation angle. When the flexible mechanical arm rotates upward, the elasticity of the center flexible shaft will cause the actual rotation angle of the flexible mechanical arm to be larger than the actual required rotation angle. When $0 \leq e(t) \leq 1$, i.e., the soft mechanical arm rotates downward, the change of the actual required rotation angle and the deviation of the actual rotation angle of the soft mechanical arm is $\left(\frac{1}{e(t)+1} - 1\right)$. When $-1 \leq e(t) < 0$, i.e., the soft mechanical arm rotates upward, the change of the actual required rotation angle of the soft mechanical arm and the deviation of the actual rotation angle of the soft mechanical arm is $\left(\frac{1}{1-e(t)} - 1\right)$.

4. Results

4.1. Experimental Environment and Data Preprocessing

In this experiment, an industrial camera is employed to capture the image of the perforating robot arm and perform the image acquisition operation within this simulated tunnel environment. To ensure the accuracy and reproducibility of the experiment, the parameters within the environment were strictly configured, and the specific details of the environmental parameters involved in the experiment are presented in Table 1.

Table 1. Experimental software and hardware environment.

Category	Description
GPU Type	NVIDIA GeForce GTX 1650 GPU
Display Size	8 GB
Memory Capacity	64 GB
Operating System	Ubuntu 22.04.1
Acquisition Tool	Industrial Camera
Programming Language	Python 3.8
Experiment Tool	PyCharm

The dataset employed in this experiment was constructed ad hoc and comprised images captured in a simulated tunnel environment. The photos encompass a diverse range of lighting conditions, backgrounds, viewing angles, and target sizes. In this manner, a series of images and videos were gathered. The images were partially labeled with the assistance of the LabelImg tool and divided by the VOC format. Due to the limited number of samples in the original collection, data enhancement techniques were employed to increase sample diversity. These enhanced image datasets were then used to train the first-frame localization model. For the video component, the annotation was conducted using the ViBAT (Video Tracking and Behaviour Annotation Tool) [20], and the annotated perforated robotic arm video data were incorporated into the OTB-2015 dataset [21]. This phase was intended to facilitate the training of the target tracking model using the bespoke dataset, which markedly enhanced the model's performance in tracking the perforated robotic arm. The drilling details, including the Drilling Robot Detail Image, Hole Positioning Recognition Video, and Drilling Operation Video, can be accessed at https://anonymous.4open.science/r/Supplementary-File-18EF.

4.2. Evaluation and Analysis of First-Frame Localization Effects

The primary objective of the initial frame localization model in this study is to accurately and expeditiously localize the punching robot arm based on RFBNet. Consequently, the original RFBNet model is subjected to a comparative and analytical evaluation with the enhanced RFBNet in this study. The findings are presented in Table 2.

Table 2. Model parameter comparison.

Algorithm	Average Precision (AP)	FPS	Model Size (MB)
Improved RFBNet	0.94	31	135.5
RFBNet	0.91	24	146.3

The tabular data demonstrate that the enhancements made to the RFBNet model in this study markedly enhance the average accuracy and frames per second (FPS) while simultaneously reducing the model size. The average accuracy of the enhanced model is 0.94, representing a 3.3% improvement over the original RFBNet model. Additionally, the FPS has reached 31. These outcomes demonstrate that the detection precision and speed of the punching robotic arm can be significantly enhanced through the integration of ShuffleNet V2 and GIoU into the conventional RFBNet network architecture and the optimization of the loss function.

To observe the comparative effect of the initial frame localization model based on the enhanced RFBNet before and after the incorporation of each improvement strategy, the model was subjected to an ablation experiment, the results of which are presented in Table 3 below.

Table 3. Comparison results of model ablation experiments.

Model Name	Average Precision (AP)	FPS	Model Size (MB)
RFBNet	0.91	24	146.3
RFBNet + ShuffleNet V2	0.89	30	131.3
RFBNet + ShuffleNet V2 + GIoU	0.92	31	132.1
RFBNet + ShuffleNet V2 + GIoU + loss	0.94	31	135.5

In this study, when ShuffleNet V2 was adopted as the backbone network, the initial average precision (AP) dropped to 0.89, primarily due to the reduced channel capacity in its architecture. ShuffleNet V2 is designed as a lightweight model, utilizing channel splitting and depthwise separable convolutions to reduce computational complexity and parameter count. However, while this design improves computational efficiency and reduces model size, it also limits the network's capacity to represent complex details and multi-scale features, leading to an initial decline in AP. To compensate for this loss, the Receptive Field Block (RFB) module was introduced, incorporating dilated convolutions. Dilated convolutions expand the receptive field, allowing the network to capture a broader range of contextual information, particularly for multi-scale feature extraction. Through these extended convolution operations, the RFB module enhances the network's ability to perceive objects across different scales, effectively restoring multi-scale feature extraction capability. Thus, although ShuffleNet V2 reduces computational complexity, the integration of the RFB module mitigates the loss of multi-scale information and improves detection accuracy. Furthermore, to optimize bounding box regression, the study introduces the Generalized Intersection over Union (GIoU) loss function. Unlike traditional IoU loss, GIoU incorporates the concept of an enclosing box to provide a more comprehensive evaluation of the similarity between the predicted and ground truth boxes. GIoU considers not only the overlapping region but also the non-overlapping areas, leading to more precise

gradients and addressing the gradient vanishing issue encountered in IoU when bounding box overlap is low. With the introduction of GIoU loss, the model's AP further increased to 0.92, demonstrating its crucial role in refining bounding box regression and enhancing localization accuracy. Finally, by integrating Focal Loss, the model's precision was further improved, ultimately reaching an AP of 0.94. Focal Loss assigns greater weight to hard-to-classify samples, enabling the model to focus more on challenging instances, thereby improving performance in complex scenarios.

The preceding discussion collectively demonstrates that optimizing the RFBNet model not only preserves the accuracy of the initial frame localization model but also reduces its complexity. This indicates that under the same hardware configuration, the optimized model can perform the first-frame localization task of the perforating robot arm more rapidly. Figure 7 presents a comparison image of the effect of the original RFBNet and the improved RFBNet on the first-frame localization model for detecting the perforated robotic arm, which visually demonstrates the effectiveness and efficiency of the improved model.

(**a**) Original RFBNet (**b**) Improved RFBNet

Figure 7. Comparison of initial frame positioning model effect of improved RFBNet.

Figure 7a illustrates the detection efficacy of the original RFBNet, whereas Figure 7b depicts the detection efficacy of the initial frame localization model with enhanced RFBNet. The analysis demonstrates that the optimized first-frame localization model of RFBNet produces more precise target frames in locating the perforation machine arm. This establishes a robust foundation for the subsequent processing steps of target tracking.

4.3. Evaluation and Analysis of Follow-Up Effects

4.3.1. Evaluation Criteria

To determine the performance of the SiamRPN-based perforated robotic arm tracking and localization method in this study, this section quantitatively analyzes the algorithmic model by employing the evaluation criteria, i.e., the accuracy and success rate, of the OTB dataset. The validation experiments for the specific implementation were conducted through the One-Pass Evaluation (OPE) model.

Precision: Calculate the position offset between the center of the predicted target bounding box and the center of the real bounding box obtained from the SiamRPN-based tracking and localization method of the perforated robotic arm and set a threshold value of 20 pixels. A position offset less than this threshold is considered to be an accurate prediction, and one more than this threshold is considered to be a prediction failure. By calculating the position offsets of all frames in the test set, the ratio of the number of frames with accurate prediction to the total number of frames is taken as the accuracy rate, where the center position error is calculated as follows:

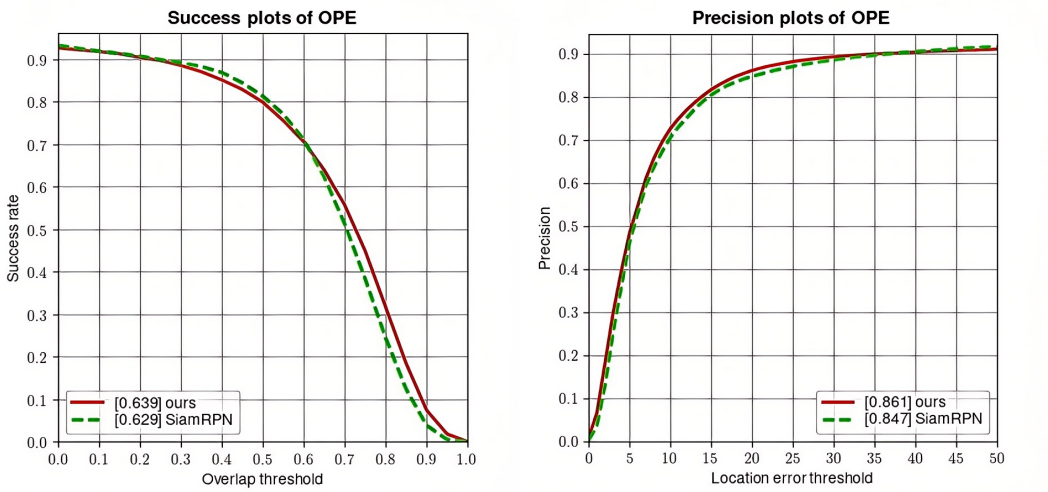
$$\|C_P - C_G\|_e \leq T_P \tag{26}$$

where C_P denotes the center position of the tracking result, C_G denotes the center position of the real labeled bounding box, T_P is the set threshold value, and $\|\cdot\|_e$ denotes the calculated Euclidean distance.

Success: The intersection ratio between the predicted target bounding box and the real target bounding box obtained from the SiamRPN-based tracking and localization method for the perforated robotic arm is calculated and the threshold is set to 0.5. If the IoU value exceeds this threshold, the prediction is recognized as correct, and if it is lower than this threshold, it is considered a prediction failure. The ratio of the number of correctly predicted frames to the total number of frames is taken as the success rate by evaluating the intersection and concurrency ratio between the prediction results and the real target for all frames in the entire test set.

4.3.2. Experimental Results

Firstly, the proposed SiamRPN-based tracking and localization method for punching the robotic arm in this study is compared with the original benchmark algorithm, SiamRPN, and the obtained success and accuracy rates are shown in Figures 8a and 8b, respectively. Both in terms of the success rate and the accuracy rate, the method of this study has a better result compared to the benchmark algorithm, with an increase of the success rate by 1.6% and the accuracy rate by 1.7%.



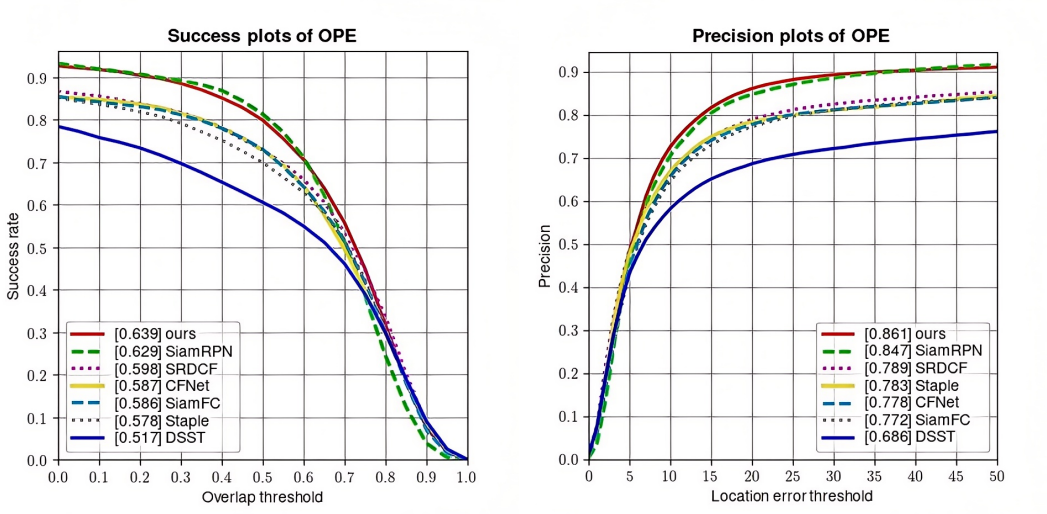
(**a**) Success Plots of OPE (**b**) Precision Plots of OPE

Figure 8. Comparison of success rate and accuracy rate between proposed algorithm and baseline algorithm.

Not only the improved target tracking method of this study is compared with the original SiamRPN algorithm, but also several classical target tracking algorithms related to the method of this study are selected for comparison, which includes the breakthrough algorithms SiamFC using a twin network, Staple [22] using deep convolutional features and DSST [23] based on correlation filtering, SRDCF [24], and CFNet [25]. As shown in Figure 9, this research algorithm still outperforms other classical target tracking algorithms; Table 4 details the comparison data of the proposed method with the benchmark algorithm and other classical algorithms in terms of success rate and accuracy, where the success rate of this research method reaches 0.639 and the accuracy rate is 0.861.

Table 4. Comparison of tracking performance between proposed algorithm and classical algorithms.

Algorithm	Success	Precision
Ours	0.639	0.861
SiamRPN	0.629	0.847
SiamFC	0.586	0.772
Staple	0.578	0.783
DSST	0.517	0.686
SRDCF	0.598	0.789
CFNet	0.587	0.778

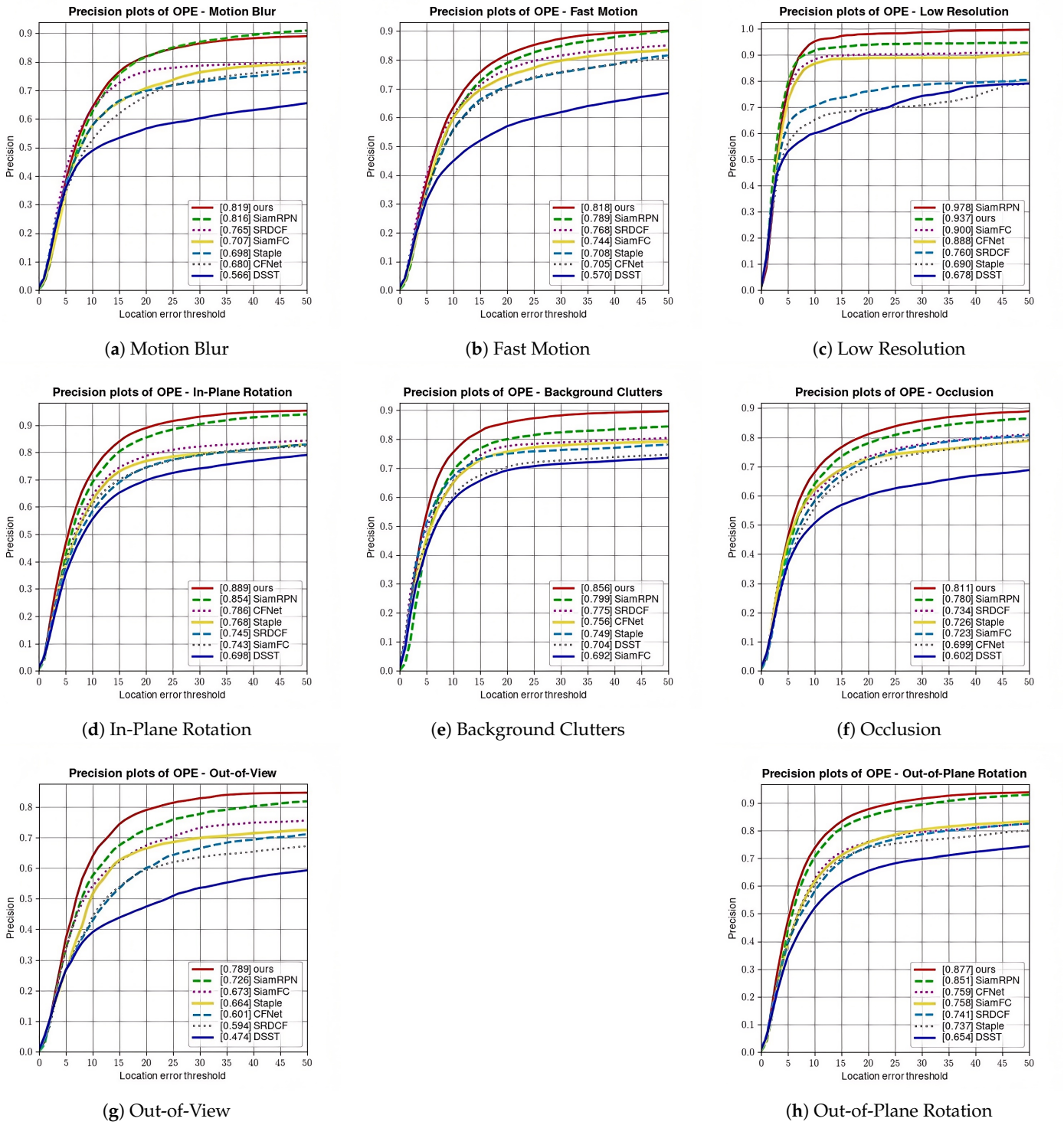


(**a**) Success Plots of OPE (**b**) Precision Plots of OPE

Figure 9. Comparison of success rate and accuracy rate between proposed algorithm and classical algorithm.

The execution of the tracking task of the perforating robotic arm is carried out in a dynamic environment, which is characterized by a high degree of uncertainty. The perforating robotic arm itself is in a constant state of flux, and a variety of unanticipated challenges may arise at any time. A series of experimental trials were conducted to comprehensively examine the adaptability and efficacy of the SiamRPN-based perforating robotic arm tracking and localization methodology employed in this investigation within a variable and intricate environment. The tests addressed a series of scenarios: blur, fast motion, low resolution, occlusion, in-plane and out-of-plane rotation, out-of-field, out-of-view, and background clutter. The results of these tests were then compared, and the resulting accuracy and success curves are presented in Figures 10 and 11 for visual comparison and analysis of performance.

The experimental results demonstrate that the method proposed in this study demonstrates superior performance to classical algorithms when confronted with a variety of complex scenarios, including motion blur, fast movement, background clutter, occlusion, and out-of-plane rotation. In particular, there is a significant improvement of 4.0% in accuracy and a 4.4% increase in success rate in scenes with occlusion. In scenes outside the field of view, there was an 8.7% increase in accuracy and a 7.6% increase in success rate. These results validate the efficacy of the Kalman filter-based target prediction algorithm and the target detection-based repositioning method for the perforating robotic arm used in this study, demonstrating an improvement in system determination efficiency and accuracy when dealing with occlusion and out-of-field-of-view scenes. While the success rate is not the highest in low-resolution and in-plane rotation scenes, it is not far from the optimal method.

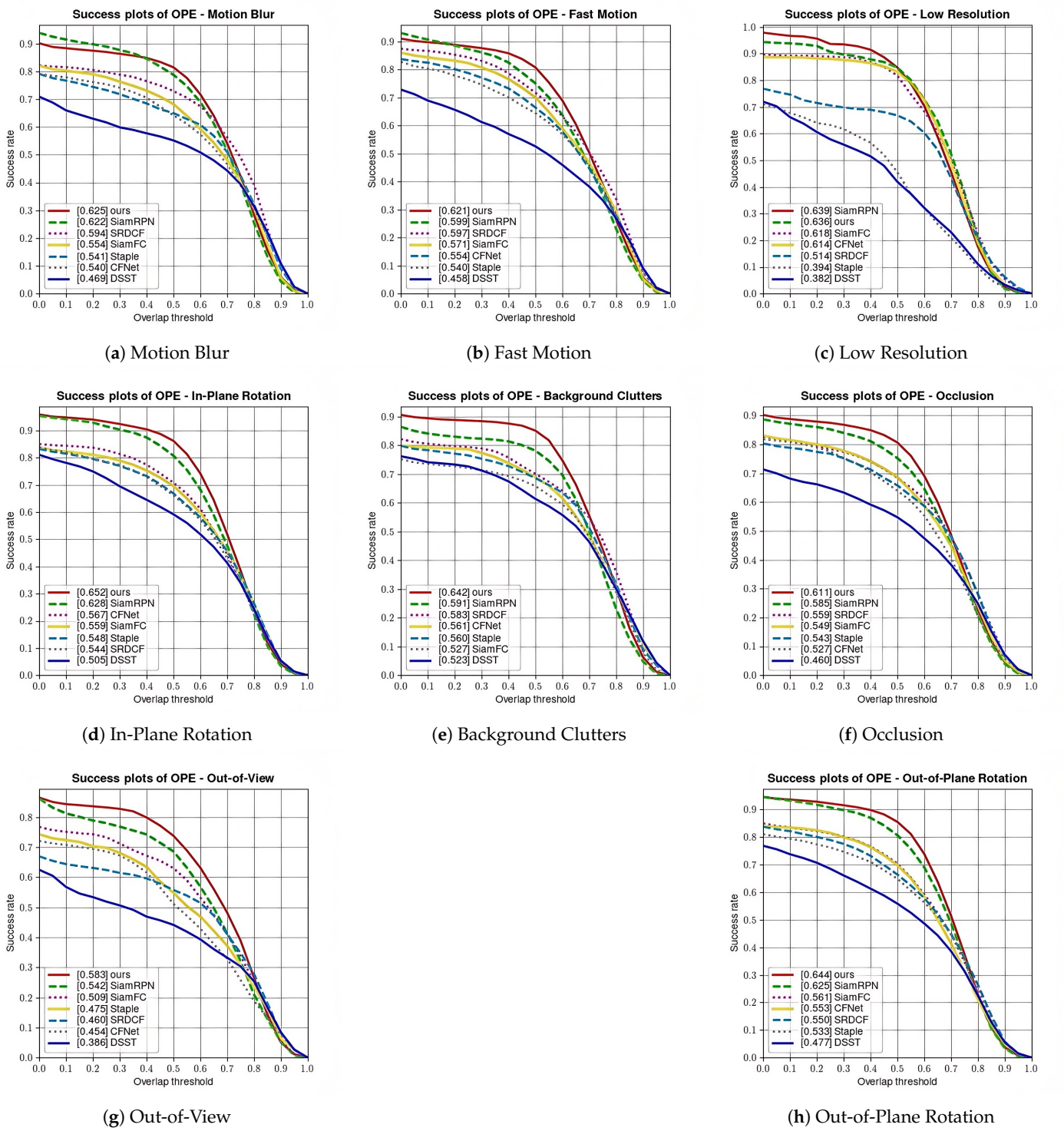


(**a**) Motion Blur (**b**) Fast Motion (**c**) Low Resolution

(**d**) In-Plane Rotation (**e**) Background Clutters (**f**) Occlusion

(**g**) Out-of-View (**h**) Out-of-Plane Rotation

Figure 10. Comparison of accuracy rates between the proposed method and the classical algorithm. Figures are arranged in three columns per row.



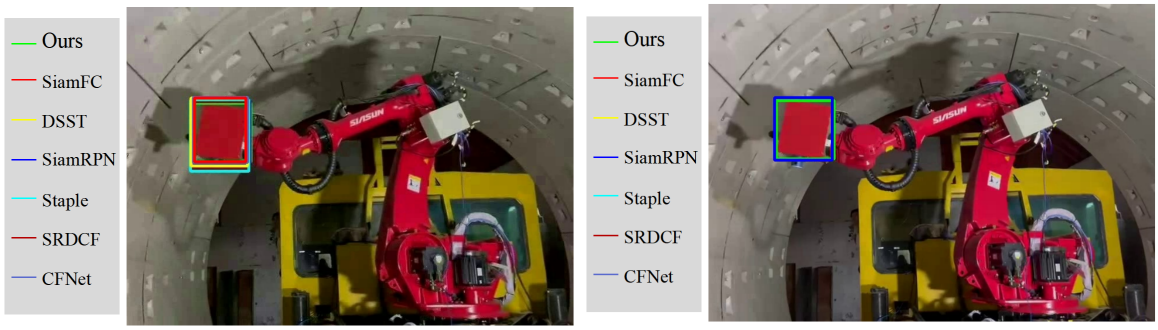
(**a**) Motion Blur (**b**) Fast Motion (**c**) Low Resolution

(**d**) In-Plane Rotation (**e**) Background Clutters (**f**) Occlusion

(**g**) Out-of-View (**h**) Out-of-Plane Rotation

Figure 11. Comparison of accuracy rates between the proposed method and the classical algorithm. Figures are arranged in three columns per row.

To visually express the tracking results of the proposed SiamRPN-based tracking and localization method for punching the robotic arm in this study, the tracking results of the 276th frame in the video sequence are presented in Figure 12a. The figure illustrates the tracking outcomes of the proposed method in conjunction with the original SiamRPN algorithm, whereas Figure 12b depicts the tracking outcomes of the proposed method in conjunction with the classical algorithm.



(**a**) Success Plots of OPE (**b**) Accuracy Plots of OPE

Figure 12. Comparison of tracking results of the proposed method with the baseline algorithm and the classical algorithm.

5. Conclusions

This study proposes a hand–eye separation-based robot initial frame localization and follower tracking method to address the issues of image quality degradation, limited computational resources, and complex environmental interference in tunnel drilling operations. First, by improving the RFBNet model, ShuffleNet V2 is used as a lightweight alternative to VGG16 for the backbone network. This, combined with channel splitting and depthwise separable convolutions, reduces the model's parameters by 15 MB and boosts inference speed to 31 FPS, significantly adapting to the resource constraints of edge devices. Additionally, the GIoU loss function is introduced to optimize bounding box regression. By calculating the minimum enclosing box, it resolves the gradient vanishing issue in traditional IoU, improving average precision (AP) by 3.3%. For continuous tracking, the SiamRPN algorithm is combined with Kalman filtering for prediction, along with a target detection-based re-localization strategy. This effectively addresses occlusions and target loss, improving the success rate and accuracy by 1.6% and 1.7%, respectively. Furthermore, by integrating a PID controller with dynamic compensation terms, tracking jitter caused by gravity and elastic deformation in the flexible robotic arm is suppressed, further enhancing system stability. Experimental results demonstrate that the proposed method shows excellent localization accuracy and tracking robustness in simulated tunnel environments. The improved initial frame localization model achieves an average precision of 0.94 in complex scenarios such as occlusions, rapid motion, and background interference. The tracking algorithm achieves a success rate and accuracy of 0.639 and 0.861, respectively, on the OTB-2015 dataset, significantly outperforming traditional SiamRPN and other classic algorithms. This research provides an efficient, low-energy, and deployable solution for intelligent drilling operations in tunnel construction, addressing the challenges posed by more extreme industrial environments.

Author Contributions: Conceptualization, H.Z., J.G. and B.Z.; methodology, J.G., H.Z. and C.X.; validation, C.X.; investigation, J.G. and H.Z.; data curation, H.Z. and C.X.; writing—original draft preparation, H.Z.; writing—review and editing, J.G.; supervision, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The OTB-2015 datasets can be obtained from http://www.cs.toronto.edu/~dross/ivt/.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Fu, Q.; Wang, S.T.; Wang, J.; Liu, S.N.; Sun, Y.B. A lightweight eagle-eye-based vision system for target detection and recognition. *IEEE Sens. J.* **2021**, *21*, 26140–26148. [CrossRef]
2. Nasrabadi, N.M. Deeptarget: An automatic target recognition using deep convolutional neural networks. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 2687–2697. [CrossRef]
3. Lei, M.; Liu, L.; Shi, C.; Tan, Y.; Lin, Y.; Wang, W. A novel tunnel-lining crack recognition system based on digital image technology. *Tunn. Undergr. Space Technol.* **2021**, *108*, 103724. [CrossRef]
4. Li, M.; Tian, W.; Hu, J.; Wang, C.; Liao, W. Study on shear behavior of riveted lap joints of aircraft fuselage with different hole diameters and squeeze forces. *Eng. Fail. Anal.* **2021**, *127*, 105499. [CrossRef]
5. Liu, H.; Zhu, W.; Ke, Y. Pose alignment of aircraft structures with distance sensors and CCD cameras. *Robot. Comput.-Integr. Manuf.* **2017**, *48*, 30–38. [CrossRef]
6. Deng, L.; Yang, M.; Li, T.; He, Y.; Wang, C. RFBNet: Deep multimodal networks with residual fusion blocks for RGB-D semantic segmentation. *arXiv* **2019**, arXiv:1907.00135.
7. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with Siamese region proposal network. In Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.
8. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
9. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
10. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NIPS), Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
11. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
12. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
13. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
14. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
15. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully convolutional Siamese networks for object tracking. In Proceedings of the 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016; pp. 850–865.
16. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware Siamese networks for visual object tracking. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Volume 11213, pp. 103–119.
17. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. ShuffleNet v2: Practical guidelines for efficient CNN architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
18. Rezatofighi, H.; Tsoi, N.; Gwak, J.Y.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
19. Yang, B. Research on Single-Target Tracking Algorithm Based on Deep Siamese Neural Networks. Master's Thesis, Dalian University of Technology, Dalian, China, 2024.
20. Biresaw, T.A.; Nawaz, T.; Ferryman, J.; Dell, A.I. ViTBAT: Video tracking and behavior annotation tool. In Proceedings of the 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Colorado Springs, CO, USA, 23–26 August 2016; pp. 295–301. [CrossRef]
21. Wu, Y.; Lim, J.; Yang, M.H. Object Tracking Benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
22. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H. Staple: Complementary learners for real-time tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.

23. Danelljan, M.; Khan, F.S.; Hager, G. Discriminative scale space tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *39*, 1561–1575. [CrossRef] [PubMed]
24. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
25. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.