Diffusion Model-based Directional Target Detection for Robotic Sorting Task

1st Chaoze Wang
Huazhong University of Science and Technology
School of Artificial Intelligence and Automation
Wuhan, China
wangcz@hust.edu.cn

2nd Gang Peng*
Huazhong University of Science and Technology
School of Artificial Intelligence and Automation
Wuhan, China
penggang@hust.edu.cn

3rd Chaowei Song
Huazhong University of Science and Technology
School of Artificial Intelligence and Automation
Wuhan, China
cwsong@hust.edu.cn

4th Cheng Lai
Huazhong University of Science and Technology
School of Artificial Intelligence and Automation
Wuhan, China
lai_cheng_hust@163.com

5th Mingjun Cong
Huazhong University of Science and Technology
School of Artificial Intelligence and Automation
Wuhan, China
1791169896@qq.com

6th Jiaqi Yang
Huazhong University of Science and Technology
School of Artificial Intelligence and Automation
Wuhan, China
2012574331@qq.com

***Abstract*—In the field of industrial automation, the efficiency and accuracy of robotic arm sorting tasks are crucial for increasing productivity. The core contribution of this study is the proposal of a new method, DiffDDet (Diffusion Model-based Directional Target Detection), which not only achieves target detection but, more importantly, provides directional information of the targets, which is lacking in traditional target detection technologies. DiffDDet improves upon the DiffusionDet target detection method by outputting directional information of targets alongside bounding boxes and target categories, and by considering the cyclic nature of directions, it improves the computation method of the sigmoid focal loss function, making it better adapted to learning directions. Furthermore, to further enhance the efficiency of sorting path planning, we have improved the traditional genetic algorithm and developed a new path planning algorithm. This algorithm optimizes genetic operations by intervening in the initial generation of the population with the nearest neighbor algorithm, significantly improving the speed and adaptability of path planning, making it particularly suitable for robotic arm sorting tasks. The experimental results of this study demonstrate that the advantages of DiffDDet in providing target directional information, combined with the improved genetic algorithm path planning, can significantly enhance the overall performance and efficiency of robotic arm sorting operations.**

***Index Terms*—Robotic Sorting, Directional Target Detection, Diffusion Model**

I. Introduction

With the rapid development of industrial automation, the application of robotic arms in tasks such as component sorting has become increasingly critical. These tasks require the robotic arms not only to accurately recognize and locate target objects but also to understand and predict the direction of the objects to achieve efficient sorting operations. However, traditional object detection technologies often overlook the direction information of the targets, leading to insufficient efficiency and accuracy in robotic arm sorting.

In response to this challenge, this study proposes DiffDDet, an innovative diffusion model-based object detection method that can simultaneously identify target objects and provide their direction information. DiffDDet generates the directionality of targets through the diffusion process, providing key information for the precise grasping and sorting of robotic arms, which is lacking in traditional technologies.

In the field of path planning, although genetic algorithms have shown excellent performance in optimization problems, their efficiency and adaptability in the application of robotic arm sorting path planning still have room for improvement. Therefore, this study has improved the genetic algorithm and developed a new path planning algorithm that significantly improves the speed and adaptability of path planning through the optimization of genetic operations.

The aim of this study is to explore and implement an integrated robotic arm sorting system that incorporates orientation-aware object detection and efficient path planning. Experimental results have demonstrated the advantages of DiffDDet in obtaining target orientation information and the efficiency of the improved genetic algorithm in path planning. The integration of these technologies not only enhances the accuracy of robotic arm sorting but also greatly improves work efficiency.

1) Improve the DiffusionDet object detection method [2], proposing DiffDDet, which outputs target direction information along with the target box and target category.

*Corresponding author: Gang Peng. This work was supported in part by No. HBSNYT202213.

979-8-3315-1056-5/25/$31.00 ©2025 IEEE

2) In response to the cyclic nature of direction, the sigmoid focal loss function has been optimized, enhancing the model's adaptability and effectiveness in learning direction information.
3) The genetic algorithm path planner has been improved by intervening in the initial population with the nearest neighbor algorithm, effectively enhancing the efficiency of path planning for robotic arm sorting tasks.

II. Related Work

A. Application of Diffusion Models in Object Detection

The application of diffusion models in the field of object detection has been a hot topic in recent years. DiffDet4SAR [1] is a study that introduces diffusion models into the field of SAR image aircraft target detection, mapping the SAR aircraft target detection task to a denoising diffusion process of bounding boxes, effectively adapting to the large variations in aircraft size.

Another study, DiffusionDet [2], proposed a new object detection framework that models object detection as a denoising diffusion process from noise boxes to target boxes. This framework allows the target boxes to diffuse from real bounding boxes to random distributions during the training phase, and the model learns to reverse this noise process, providing a new perspective on the application of diffusion models in object detection. Additionally, it has led to the development of 3D object detection [3] and RGBD-based object detection [4].

B. Improvements of Genetic Algorithms in Path Planning

A study [5] proposed the use of genetic algorithms to optimize point-to-point trajectory planning for a robotic arm. The objective function of this study aims to minimize travel time and space while ensuring that the maximum predefined torque is not exceeded and avoiding collisions with any obstacles in the workspace.Another study [6] employed genetic algorithms to optimize the path planning of a robotic arm, with a particular emphasis on minimizing the energy consumed by the actuator and travel time.

A recent study [7] on the optimal motion planning energy optimization for dual-arm industrial robots. The objective function of this study is based on the execution time and total energy consumption of the robot arm configuration performing pick-and-place operations in the workspace. The study first uses a PID controller to achieve optimal parameters and then fine-tunes the PID parameters using genetic algorithms to create more precise robot motion trajectories, achieving energy-saving robot configurations.

III. Methodology

A. Diffusion Model

Diffusion models are a type of generative model that generates data by simulating the gradual denoising process of the data. DDIM (Denoising Diffusion Implicit Models) [8] is a variant of diffusion models, which allows for non-Markovian sampling processes, thus enabling faster sampling speeds. DDIM derivation process is summarized as follows:

1) Forward Process of DDIM: The forward process of DDIM is the same as that of DDPM (Denoising Diffusion Probabilistic Models) [9], which can be expressed as:
$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t}\epsilon_t \tag{1}$$
where x_t is the noisy image at time step t, x_0 is the original noise-free image, ϵ_t is noise sampled from the standard normal distribution $N(0,1)$, and α_t is the cumulative product of noise levels.
2) Reverse Sampling Process of DDIM: The key to DDIM lies in its reverse sampling process, which allows for non-Markovian sampling steps. For given x_t and x_0, DDIM defines the conditional distribution $q(x_{t-1}|x_t, x_0)$:
$$p(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \kappa_t x_t + \lambda_t x_0, \sigma_t^2 I) \tag{2}$$
where κ_t and λ_t are coefficients to be determined, and σ_t^2 is the variance.
3) Determination of Mean and Variance: By solving for the coefficients a, b, and σ_t, the distribution of $q(x_{t-1}|x_t, x_0)$ can be determined. Specifically, we have:
$$\kappa_t = \sqrt{\frac{1-\alpha_{t-1}-\sigma_t^2}{1-\alpha_t}} \tag{3}$$
$$\lambda_t = \sqrt{\alpha_{t-1}} - \sqrt{\frac{1-\alpha_{t-1}-\sigma_t^2}{1-\alpha_t}}\sqrt{\alpha_t} \tag{4}$$
The determination of these coefficients allows us to predict from x_t directly to x_0, accelerating the sampling process.
4) DDIM Sampling Formula: DDIM allows skipping intermediate time steps and directly predicting from x_t to x_0, thus accelerating the sampling process. The sampling steps can be expressed as:
$$x_0 = \frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \tag{5}$$
where $\epsilon_\theta^{(t)}(x_t)$ is the noise predicted by the model.

B. Directional Object Detection

As shown in Fig. 1, the DiffDDet framework describes the object detection task as a denoising diffusion process from noise boxes to target boxes. During the training phase, the target boxes diffuse from the true boxes to a random distribution, and the model learns to reverse this process, i.e., recovering the true annotated boxes from the noise. At the same time, the model trains two classification heads: one for category classification and the other for direction classification. During the inference phase, the model gradually refines a set of randomly generated target boxes and finally outputs results containing target boxes, categories, and directions.

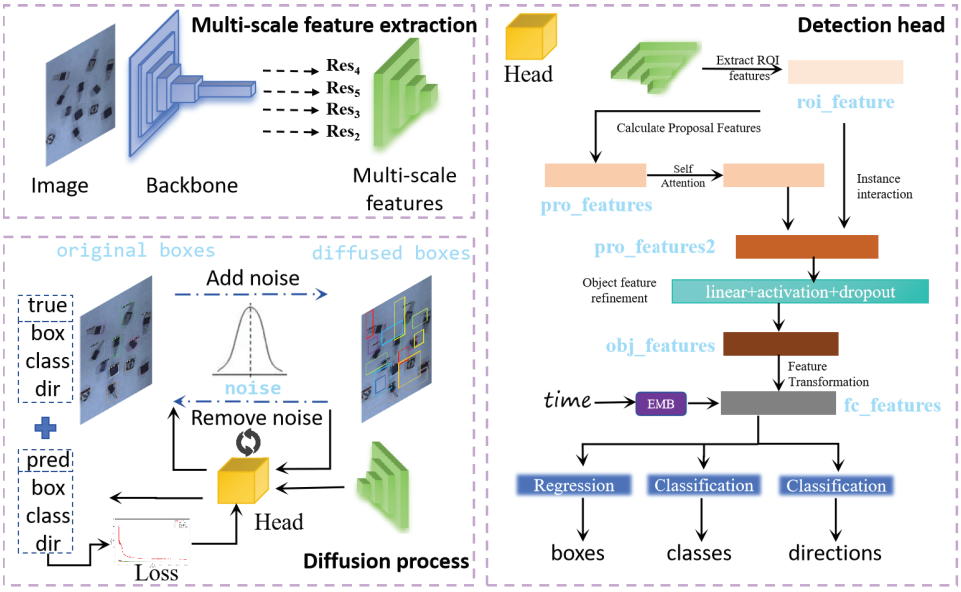

Fig. 1. DiffDDet Overall Framework

Similar to DiffusionDet [2], DiffDDet takes the original image as input and uses the detection head to extract high-level features. The framework employs convolutional neural networks (e.g., ResNet [10]) and Transformer-based models (e.g., Swin [12]) to implement it. Feature pyramid networks are used to generate multi-scale feature maps for the ResNet and Swin backbone networks. This study selects ResNet50 as the backbone network, extracting features from the res2, res3, res4, and res5 layers.

During the denoising process, DiffDDet does not use the Unet structure [13] but instead adopts the idea of R-CNN [14]. The detection head receives a set of proposal boxes as input, crops RoI features from different scale feature maps, and goes through a series of computational steps, including proposal features, self-attention mechanisms, instance interaction enhancement, object feature refinement, temporal embedding, and feature transformation, to obtain the final features. These features are then sent to the regression classifier to obtain box regression results, label classification results, and direction classification results.

C. Design of Direction Loss Function

When dealing with direction classification problems, there is a challenge different from standard label loss calculation, which is the cyclic nature of direction. For example, the difference between 0° and 135° should be considered the same as the difference between 0° and 45° in loss calculation. To address this issue, we introduce a dynamic penalty term α to consider the cyclic nature of direction. Specifically, we adjust the penalty term based on the cyclic distance between each prediction and the true direction, magnifying the loss for predictions far from the true category and making the loss relatively smaller for predictions close to the true category. Moreover, we integrate the principles of Sigmoid Focal Loss [15] to diminish the impact of easily classified samples on the loss and amplify the impact of challenging samples, thus improving the model's discriminative power.

To clearly articulate the loss function, the pseudo code for the Sigmoid Focal Circle Loss (SFCL) is provided below:

Algorithm 1 Sigmoid Focal Circle Loss

procedure SFCL($inputs, targets, \gamma, num\_dirs$)
  sig $\leftarrow$ sigmoid($inputs$)
  losses $\leftarrow$ zeros($batch\_size$)
  for $i = 0$ to $batch\_size - 1$ **do**
    pos_cls $\leftarrow$ argmax($targets[i]$)
    pred_cls $\leftarrow$ argmax(sig$[i]$)
    distances $\leftarrow$ cal_cyclic_dis(num_dirs, pos_cls, pred_cls)
    $\alpha\_values \leftarrow$ calculate_alpha(distances, num_dirs)
    sample_loss $\leftarrow 0$
    for $j = 0$ to $num\_classes - 1$ **do**
      ce $\leftarrow -(targets[i][j] \cdot \log(\text{sig}[i][j]) + (1 - targets[i][j]) \cdot \log(1 - \text{sig}[i][j]))$
      term1 $\leftarrow (1 - \text{sig}[i][j])^{\gamma}$
      term2 $\leftarrow (\text{sig}[i][j])^{\gamma}$
      fl $\leftarrow \alpha\_values[j] \cdot$ term1 $\cdot$ ce $+ (1 - \alpha\_values[j]) \cdot$ term2 $\cdot$ ce
      sample_loss $\leftarrow$ sample_loss + fl
    end for
    losses$[i]$ $\leftarrow$ sample_loss
  end for
  return mean(losses)
end procedure

The `cal_cyclic_dis` function computes the cyclic distance between the predicted direction and the true direction,

considering the periodic nature of the directions.

$$\text{cyclic\_distance} = \min\Big(|\text{pred\_cls} - \text{pos\_cls}|, \; num\_dirs - |\text{pred\_cls} - \text{pos\_cls}|\Big) \tag{6}$$

num_dirs: The total number of direction classes. pos_cls: The index of the true direction class. $pred\_cls$: The index of the predicted direction class.

The `calculate_alpha` function dynamically computes penalty weights for each direction class based on their cyclic distances from the true direction. Using `cal_cyclic_dis`, it assigns higher penalty weights to predictions further from the true direction.

$$\alpha_{\text{value}}[j] = \frac{1}{1 + \text{distance}(j)} \tag{7}$$

The Sigmoid Focal Circle Loss (SFCL) for each class i is given by Eq 8:

$$\begin{aligned}\text{SFCL}_i = &-\alpha_i \cdot (1 - p_i)^\gamma \cdot \log(p_i) \\ &- (1 - \alpha_i) \cdot p_i^\gamma \cdot \log(1 - p_i)\end{aligned} \tag{8}$$

Where p_i is the probability predicted by the model for class i, α_i is the dynamic penalty term, and γ is the focusing parameter.

As given in Eq 9, the total SFCL is the sum of the losses for all classes:

$$\text{Total SFCL} = \sum_{i=1}^{4} \text{SFCL}_i \tag{9}$$

D. Improved Genetic Algorithm

When exploring the path planning problem for robotic arm sorting, this study adopts a real-number encoding strategy to replace traditional binary encoding. Real-number encoding represents potential solutions as vectors, simplifying crossover and mutation operations and reducing encoding errors.

The research process is as follows:

1) Initial Population: Random number sequences are generated as genetic encodings, e.g., 7 3 1 6 4 9 8 0 2 5 or 2 7 4 6 3 0 8 1 5 9.
2) Fitness Evaluation: Fitness is defined as the inverse of the total path length executed by the robotic arm. The path length includes steps such as grasping components and moving to sorting boxes. Distance is calculated as:

$$\begin{aligned}\text{Distance} = \sum_{i=1}^{n} \Big(&\sqrt{(x_i - x_{B_i})^2 + (y_i - y_{B_i})^2} \\ &+ \sqrt{(x_{B_i} - x_{i+1})^2 + (y_{B_i} - y_{i+1})^2}\Big)\end{aligned} \tag{10}$$

where (x_i, y_i) is the position of the i-th component, and (x_{B_i}, y_{B_i}) is the sorting position. Fitness is calculated as:

$$\text{Fitness} = \frac{1}{\text{Distance}} \tag{11}$$

3) Selection Mechanism: The roulette wheel selection method is used, where individuals are selected based on their fitness values.
4) Crossover Operation: Two chromosomes are selected for crossover by exchanging gene segments.
5) Mutation Operation: Some gene encodings are randomly changed with a predetermined probability.

After several generations, the population stabilizes, and the optimal solution is selected for decoding to obtain the shortest path sorting plan. For scenarios with a large number of components, traditional genetic algorithms may face long optimization times and cannot guarantee finding the global optimal solution. Therefore, this study proposes an improved strategy: introducing a better individual obtained by the nearest neighbor algorithm into the initial population. This approach helps speed up convergence and improve solution quality.

IV. Experimental Results and Analysis

A. Annotation of Directional Dataset

In this study, considering the design characteristics of the robotic arm gripper, which is a vertically downward double-finger gripper, there is no need to precisely control the direction of the object when grasping. It is only necessary to recognize the discrete direction of the component to calculate the appropriate grasping angle. Therefore, as shown an Fig. 2, the dataset in this study uses discrete values to represent the direction of the components: 0 represents the component is horizontally placed, 1 represents the component is tilted at 45°, 2 represents the component is vertically placed, and 3 represents the component is tilted at 135°. The dataset consists of 60 actual collected images, covering three types of components: stabilivolt (144 instances), drcode (124 instances), and inductance (185 instances).

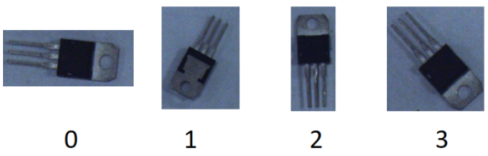

Fig. 2. Example of directional dataset annotation

B. Experimental Platform

Hardware Facilities: As shown as Fig. 3, the experimental platform selected in this study includes the DOBOT Magician robotic arm, which is the world's first desktop collaborative robot independently developed by Yuejiang Robot. This robotic arm provides strong hardware support for the experiment with its flexibility and precision. For the camera platform, we chose the industrial camera plA2400-17gc with high resolution and fast imaging capabilities, and used the Basler_pylon software for configuration to achieve high-speed data transfer through the GigE interface.

Software Configuration: To achieve precise control and monitoring of the robotic arm, we developed a grasping operation software based on the QT framework. As shown as Fig. 4, the software provides a user-friendly interface, allowing

Fig. 3. Robic arm and camera platform

operators to easily control the robotic arm to perform grasping tasks and monitor its working status in real-time.

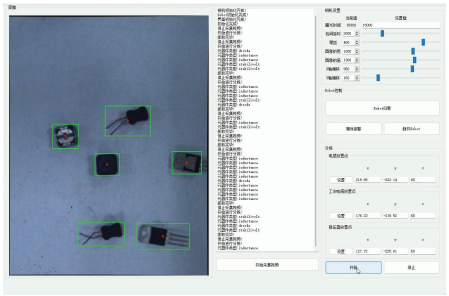

Fig. 4. Robotic arm sorting operation software

Table I shows the environmental configuration used in DiffDDet model.

TABLE I
MODEL ENVIRONMENT CONFIGURATION

Parameter Name	Parameter Settings
sys.platform	linux
Python	3.8.16
detectron2	0.6
Compiler	GCC 7.5
CUDA compiler	CUDA 11.0
PyTorch	1.10.1+cu111
GPU 0	NVIDIA TITAN Xp
cv2	4.7.0

C. DiffDDet Experiment

In this study, we explored the application of direction detection in object detection tasks and compared the performance of YOLO v8 [16], DiffusionDet [2], and the method proposed in this paper, DiffDDet. Since YOLO v8 and DiffusionDet do not inherently support direction detection, we adopted an indirect approach to achieve this functionality: by combining labels with direction, for example, marking a stabilivolt with an direction of 3 as stabilivolt3, thus allowing the model to learn orientation information during training. Table II shows the training results of these three methods, where the performance metrics of DiffDDet are significantly higher than the other two methods, indicating the effectiveness of our approach.

TABLE II
COMPARISON OF DIFFERENT MODELS

Model	mAP50(%)
YOLO v8	92.627
DiffusionDet	92.372
DiffDDet (Ours)	95.540

Fig. 5 shows the changes in total loss, bounding box loss, label loss, and direction loss during the model training process, revealing the gradual improvement in model performance. Fig. 6 compares the average precision (AP) of different models on various components, with DiffDDet performing the best in all categories.

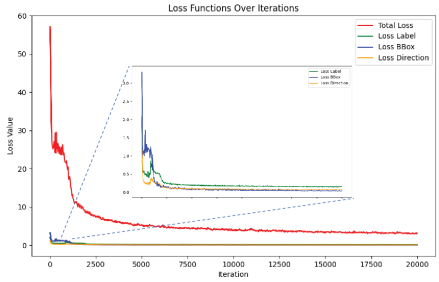

Fig. 5. Changes in Various Losses During Model Training

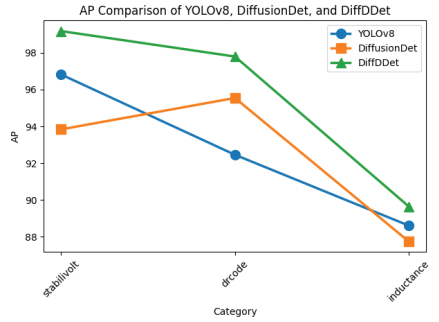

Fig. 6. AP of Different Models on Different Components

Fig. 7's visualization results show how the DiffDDet model uses orientation information to predict the robotic arm's grasping angle, thereby improving the accuracy and efficiency of the sorting task.

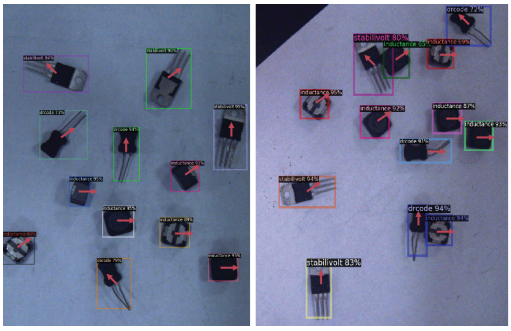

Fig. 7. Model Inference Results Including Direction Information

D. Improved Genetic Algorithm Experiment

In path planning research, Fig. 8 illustrates that the improved genetic algorithm achieves shorter path lengths and faster convergence to the optimal solution compared to the standard version.

Fig. 9 depicts the iterative process of genetic algorithm path planning, where different colored dashed boxes indicate the predetermined placement positions of the components, and the numbers indicate the order of grasping.

To reduce the impact of randomness in genetic algorithms, this study applied both algorithms to 21 samples, conducting

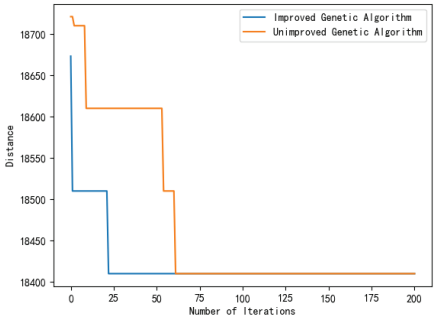

Fig. 8. Comparison of Improved and Unimproved Genetic Algorithms

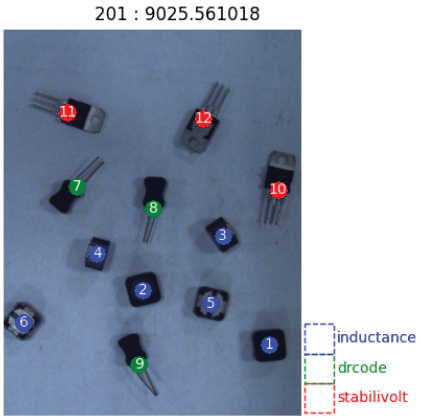

Fig. 9. Iterative Results of Genetic Algorithm

10 independent experiments for each. In data analysis, we excluded the extreme values of the number of iterations in each experiment and took the average. The results are shown in Fig. 10, where the improved genetic algorithm reduced the number of iterations in path planning by an average of 3 cycles. When the number of targets is large, the advantage of this improved algorithm is more significant, saving more than 15 iterations. These objective data reflect the significant effect of the improved algorithm in improving efficiency and performance.

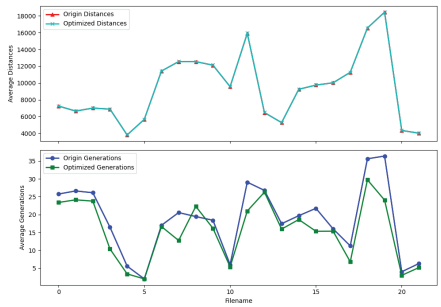

Fig. 10. Comparison of Two Methods for Multiple Planning Tasks

V. Conclusion

In this study, we propose an innovative solution to address efficiency and accuracy issues in robotic arm sorting tasks within industrial automation. The DiffDDet method is developed to detect target objects and provide accurate directional information, thereby enhancing sorting precision. This method optimizes the target detection algorithm by improving the Sigmoid Focal Loss function, effectively capturing directional characteristics and overcoming limitations of traditional techniques. Additionally, we enhance genetic algorithms through a new path planning algorithm that optimizes the initial population using the nearest neighbor approach, improving planning speed and adaptability. Experiments confirm the effectiveness of these methods. DiffDDet shows significant advantages in obtaining target direction, and when integrated with the improved genetic algorithm, it significantly boosts the overall performance and efficiency of robotic arm sorting.

Acknowledgment

This work was supported by Hubei Province Core Technology Application Research Project for Agricultural Machinery Equipment(No. HBSNYT202213).

References

[1] Zhou, Jie, et al. "DiffDet4SAR: Diffusion-based aircraft target detection network for SAR images." *IEEE Geoscience and Remote Sensing Letters* (2024).

[2] Chen, Shoufa, et al. "Diffusiondet: Diffusion model for object detection." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[3] Ho, Cheng-Ju, et al. "Diffusion-ss3d: Diffusion model for semi-supervised 3d object detection." In *Advances in Neural Information Processing Systems*, vol. 36, pp. 49100-49112, 2023.

[4] Orfaig, Eliraz, Inna Stainvas, and Igal Bilik. "Enhanced Automotive Object Detection via RGB-D Fusion in a DiffusionDet Framework." *arXiv preprint arXiv:2406.03129* (2024).

[5] Kazem, Bahaa Ibraheem, Ali Ibrahim Mahdi, and Ali Talib Oudah. "Motion planning for a robot arm by using genetic algorithm." *Jmie* 2.3 (2008): 131-136.

[6] Sharma, Gouri Shankar, Mandeep Singh, and Tejinder Singh. "Optimization of energy in robotic arm using genetic algorithm." *International Journal of Computer Science and Technology* 2.2 (2011): 315-317.

[7] Nonoyama, K.; Liu, Z.; Fujiwara, T.; Alam, M.M.; Nishi, T. "Energy-Efficient Robot Configuration and Motion Planning Using Genetic Algorithm and Particle Swarm Optimization." *Energies* 15, 2074 (2022).

[8] Song, Jiaming, Chenlin Meng, and Stefano Ermon. "Denoising diffusion implicit models." *arXiv preprint arXiv:2010.02502* (2020).

[9] Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." In *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840-6851, 2020.

[10] He, Kaiming, et al. "Deep residual learning for image recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[11] Vaswani, A. "Attention is all you need." In *Advances in Neural Information Processing Systems* (2017).

[12] Liu, Ze, et al. "Swin transformer: Hierarchical vision transformer using shifted windows." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[13] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III*, vol. 18. Springer International Publishing, 2015.

[14] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[15] Lin, T. "Focal Loss for Dense Object Detection." *arXiv preprint arXiv:1708.02002* (2017).

[16] Swathi, Y., and Manoj Challa. "YOLOv8: Advancements and Innovations in Object Detection." In *International Conference on Smart Computing and Communication*. Singapore: Springer Nature Singapore, 2024.