






A Novel Framework for Improved Grasping of Thin and Stacked Objects

Zhangli Zhou , Shaochen Wang , *Graduate Student Member, IEEE*, Ziyang Chen ,
Mingyu Cai , and Zhen Kan , *Senior Member, IEEE*

Abstract—In this study, we propose a novel top-down grasping approach for robots that combines a deep high-resolution convolutional neural network (DHRNet) and a multiview perception-based trajectory planning controller (MP-PC). Unlike the traditional encoder-decoder architecture, DHRNet preserves the high-resolution feature maps and integrates feature maps at different scales to ensure maximum retention of spatial information and its fusion with high-level semantic information. The MP-PC continuously adapts the trajectory planning of the robotic end-effector based on the outputs of the DHRNet in order to respond to situations with low confidence in the grasping detection. As a result, it improves the smoothness and accuracy of the trajectory and avoids entering singularities. We evaluate the performance of the DHRNet on the Cornell and Jacquard grasping datasets, achieving accuracies of 99.50% and 94.80%, respectively. In addition, the framework outperformed other methods in real-world experiments using a 7 degrees of freedom Franka Emika Panda robot, especially in scenarios with stacked and thin objects. The codes for this approach are available at <https://github.com/USTCzll/DHRNet-MP-PC/tree/master>.

Impact Statement—Robot grasping technology plays a pivotal role in achieving robot intelligence and autonomy. It has great potential in enhancing the efficiency of intelligent manufacturing and alleviates human workload. However, existing approaches do not perform well in grasping thin and stacked objects. It is arguable that such limitations stem mainly from the compromised accuracy in spatial detection, which leads to unnecessary exploration of motion directions, and thereby increases the possibility of encountering singularity. To address these challenges, we propose a novel grasping framework that exceeds current state-of-the-art methods in most scenarios and effectively tackles the issues associated with grasping thin and stacked objects. The efficacy of our proposed framework has been validated through extensive experiments.

Index Terms—Grasp detection, motion planning, robotic grasping.

Manuscript received 20 May 2023; revised 12 September 2023 and 1 November 2023; accepted 7 December 2023. Date of publication 12 December 2023; date of current version 21 June 2024. This work was supported in part by the National Key R&D Program of China under Grant 2022YFB4701400/4701403 and in part by the National Natural Science Foundation of China under Grant U2013601. This article was recommended for publication by Associate Editor Letizia Marchegiani upon evaluation of the reviewers' comments. (*Corresponding author: Zhen Kan.*)

Zhangli Zhou, Shaochen Wang, Ziyang Chen, and Zhen Kan are with the Department of Automation, University of Science and Technology of China, Hefei 230022, China (e-mail: zll1215@mail.ustc.edu.cn; samwang@mail.ustc.edu.cn; chen ziyang19981220@126.com; zkan@ustc.edu.cn).

Mingyu Cai is with the Department of Mechanical Engineering, University of California, Riverside, CA 92521 USA (e-mail: mingyu.cai@ucr.edu).
Digital Object Identifier 10.1109/TAI.2023.3341833

I. INTRODUCTION

ROBOTIC grasping is a fundamental research area in robotics, with a critical role in enabling robots to perform versatile and complex tasks [1], [2]. The ultimate goal of this field is to empower robots with the ability to grasp and manipulate objects with precision, dexterity, and reliability, which are essential skills for various real-world applications such as product assembly and object sorting [3], [4]. The initial stage of robotic grasping involves detecting the graspable regions of an object, and the precision of this step is a crucial factor that can determine the success or failure of the entire grasping task. Therefore, researchers have been exploring various approaches to improve the accuracy of grasp detection, such as using deep learning techniques, multisensor fusion, and visual servoing. These methods aim to extract relevant features from objects and the surrounding environment to determine optimal trajectories and grasp poses.

Recent research has made significant strides in grasp detection, with approaches based on convolutional neural networks (CNNs) garnering considerable attention [5], [6], [7], [8], [9], [10], [11]. Such methods have exhibited remarkable performance on various grasping datasets, including Cornell [12], Jacquard [13], and GraspNet-1Billion [14]. Meanwhile, advancements in grasp detection models and computational hardware have significantly reduced the time required for visually detecting grasps. Previously, detection time could take tens of seconds [12], but with recent improvements, it can now take less than a second [15], or even a fraction of a second [9]. As a consequence, grasp detection is no longer a major bottleneck for the robotic arm during gripping execution, enabling grasp detection from multiple viewpoints with little impact on overall execution time. Now, the reaching action has become an essential component of the grasping pipeline, rather than just a mechanical requirement. Previous research [8], [15] has demonstrated that this approach is an effective solution to the challenge of grasping objects in cluttered environments. However, we found that existing methods do not perform well when dealing with stacked and thin objects.

As illustrated in Fig. 1, on the left side, multiple objects are stacked together, and due to prediction bias, the end-effector collides with other objects, leading to a failed grasp. On the right side, the object is thin and has a color similar to the background in the initial stage, which makes it challenging for the grasp detection algorithm to provide effective informative data.

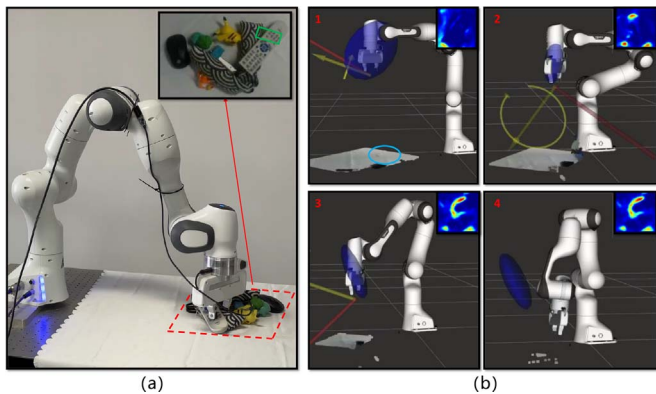


Fig. 1. Two common cases that existing grasping methods fail. (a) Stacked objects: stacked objects can lead to irrational grasping predictions (green rectangles represent predicted grasping positions and poses). (b) Thin objects: the thin object is hard to be detected in the initial phase, leading to the issue of entering a singularity and thus unable to continue completing the grasp. The blue ellipses reflect the ability of the end-effector to move in different directions in the current configuration space. The end-effector outside the blue ellipse means the arm is entering the singularity. The top-right subplot is a real-time heat map of the quality of the grasp.

Consequently, during unconfined exploration, the robotic arm may experience singularities. This could lead to difficulties in achieving the desired target position necessary to successfully grasp an object. We speculate that the main reason for these two cases is the loss of spatial resolution of the feature map due to downsampling, and the subsequent upsampling process may not accurately compensate for the lost spatial information, leading to failure in the grasping process. In addition, the end-effector of the robot arm is far from the target object at the initial position and cannot give reasonable advice for trajectory planning, so the robot arm explores freely, which easily leads to the robot arm activating the protection mechanism or entering the singularity point.

In this study, we introduce a novel neural network architecture that maintains a high-resolution feature map throughout the entire process. Our parallel-branch structure preserves a high-resolution representation while frequently exchanging information between resolutions. Our approach not only achieves state-of-the-art performance on several mainstream datasets (e.g., 99.5% in Cornell¹ [12] and 94.8% in Jacquard [13]) but also exhibits superior performance in various grasping-related complex applications in physical environments. In addition, we developed a new multiview perceptual planning controller (MP-PC), which can dynamically adjust the speed of end execution according to the information gain obtained in the grasping pipeline, improve the smoothness of the trajectory, and effectively avoid singularities. Experiments in physical environments have confirmed that our controller can effectively improve the success rate of grasping thin, stacked objects.

The main contribution of this article can be summarized as follows.

- 1) We present a novel network architecture DHRNet designed for thin and stacked object planar grasping tasks.

¹Cornell dataset: http://pr.cs.cornell.edu/grasping/rect_data/data.php.

Our approach is novel in its utilization of high-resolution feature maps, making it one of the first attempts to incorporate such maps into the field of robotic grasping.

- 2) We propose a MP-PC in the grasping pipeline, which can improve trajectory smoothness and accuracy and reduce the risk of entering singularities.
- 3) Physical experiments have been conducted to validate the superiority of our proposed model in terms of performance. Our results indicate that our method outperforms existing approaches in terms of accuracy and robustness.

II. RELATED WORKS

Grasp detection is the first and very important step for the robot to complete grasping. Accurately determining the appropriate grasping pose and locating the position of an object is essential for stable and robust robotic grasping. Grasp detection leverages camera images to infer the grasping pose of the robot manipulator. Previous works [16], [17] mainly relied on geometry-driven techniques that analyzed object contours to identify grasping points. These methods assumed that the object's geometric model was always available, but preparing CAD models for graspable objects is impractical for real-time implementation.

Recently, deep learning methods have been successfully applied in visual grasping tasks in real time [6], [18], [19], [20], [21], [22], [23], [24]. One of the earliest works that introduced deep neural networks to grasp detection is [12], which used a two-stage strategy to find possible grasping candidates and evaluate their quality. However, this method suffers from relatively slow speed due to the numerous grasping proposals. In recent years, many researchers have utilized convolutional neural networks to generate proposals for a bounding box and estimate the grasp pose of objects. For instance, Redmon and Angelova [6] used an Alexnet-like CNN architecture to regress grasping poses, while Kumra and Kanan [18] incorporated multimodal information including depth and RGB data using ResNet-50 as a backbone to improve grasp performance. Additionally, CNN-based grasp quality networks [25], [26] have been proposed to evaluate and predict the robustness of grasp candidates. Morrison et al. [9] developed GG-CNN, a fully convolutional neural network that provides a lightweight and real-time solution for visual grasping. Wang et al. [15] proposed a new architecture based on vision transform to address the loss of long-term dependency in feature extraction in current CNN-based grasp detection methods. Cortes et al. [27] proposed an imitation learning algorithm incorporating instance segmentation to support a simplified control scheme for soft gripper grasping objects with arbitrary poses. Zhang et al. [19] developed a grasping attentional convolutional neural network via real-time robot observation and force-torque feedback. Yang et al. [28] propose a novel lightweight generative convolutional neural network with grasp priority called GP-Net for multiobject grasping in densely stacked environments.

The successful execution of robotic grasping depends heavily on recent advances in computer vision, which have led to improvements in feature extraction methods and the development

of better models for popular datasets. However, detecting a viable grasp is just the first step in the grasp pipeline, and subsequent motion planning is required for successful grasping. Fortunately, modern grasp detection methods have dramatically increased in speed, allowing for multiple detections throughout the pipeline and adjusting motion planning based on information gained from different viewpoints. This enables robots to use active sensing techniques to plan the next optimal action and improve their ability to grasp objects in complex scenarios. Common strategies for active perception in robotics involve planning the next best action that efficiently minimizes measurement uncertainty or maximizes information gain, often using metrics such as Shannon entropy or KL divergence. These approaches enable the robot to intelligently select the most valuable information to gather and to leverage this information to improve the overall success of the grasping task.

Morrison et al. [9] introduced a visual grasp detection system that generates a pixelwise distribution of grasp pose estimates and then identifies the next best view location based on this information. On the other hand, Gualtieri and Platt [29] apply active perception directly to grasp detection by computing a distribution of viewpoints for object classes that are likely to improve the quality of detected grasps. However, when the target object is thin, it becomes challenging to find the next best view by maximizing the information gain through KL entropy because the depth difference is not obvious, and the robot may fall into singularities, causing it to stop abruptly or start a protection procedure.

III. FRAMEWORK

In this section, we present a comprehensive framework for robotic grasping, consisting of two pivotal components: the grasping detection algorithm, DHRNet, and the motion planning controller, MP-PC. DHRNet offers a more robust and accurate approach for grasp detection, while MP-PC is a motion planning controller that can adjust the motion planning strategy based on the detection information.

To address challenging scenarios, such as grasping thin or stacked objects, we have deeply integrated two components to enable the robot to cope with a variety of situations. In this section, we first introduce the representation method of the grasping problem in Section III-A. Next, we introduce the network architecture of our proposed deep hierarchical reinforcement learning network (DHRNet) in Section III-B. Finally, we present our motion planning controller, MP-PC, in Section III-C.

A. Grasp Representation

Autonomous visual grasping tasks typically involve collecting visual images of an object through sensory input and processing them to generate an effective grasp configuration that maximizes the probability of success. In the case of a parallel-plate gripper, the grasp can be represented as a four-tuple, as formulated by Jiang et al. [9], given by

$$\mathbf{g} = \{\mathbf{p}, \theta, w, q\} \quad (1)$$

where $\mathbf{p} = (x, y, z)$ denotes gripper's center position in Cartesian coordinates, θ denotes gripper's rotation around the z axis, and w denotes the required gripper width, respectively. In addition, a scalar quality measure q represents the chances of successful grasp.

Given a 2.5-D depth image $\mathbf{I} \in \mathbb{R}^{H \times W}$ with height H and width W taken from a camera with intrinsic matrix \mathbf{K} , the grasp in \mathbf{I} is described by

$$\mathbf{g}' = \{\mathbf{s}, \theta', w', q\} \quad (2)$$

where $\mathbf{s} = (u, v)$ is the center point in the image coordinates O_i , θ' is the rotation in the camera reference frame O_c , and w' is the grasp width in the image coordinates O_i . A grasp in the image space \mathbf{g}' can be converted to a grasp in world coordinates \mathbf{g} by applying a sequence of known transforms

$$\mathbf{g} = \mathbf{L}_{cw}(\mathbf{L}_{ic}(\mathbf{g}')) \quad (3)$$

where \mathbf{L}_{cw} transforms from the camera frame to the world (robot) frame and \mathbf{L}_{ic} transforms from 2-D image coordinates O_i to the 3-D camera frame O_c , based on the camera intrinsic matrix \mathbf{K} and known calibration between the robot and camera.

We refer to the set of grasps in the image space as the grasp map, which is denoted as

$$\mathbf{G} = (\mathbf{Q}, \Theta, \mathbf{W}) \in \mathbb{R}^{3 \times H \times W}. \quad (4)$$

The grasp map \mathbf{G} estimates the parameters of a set of grasps, executed at the Cartesian point \mathbf{p} , corresponding to each pixels. We represent the grasp map \mathbf{G} as a set of images.

- 1) \mathbf{Q} is an image describing the quality of a grasp executed at each point (u, v) . The value is a scalar within the range $[0, 1]$ where a value closer to 1 indicates higher grasp quality, i.e., higher chance of successful grasp.
- 2) Θ represents the grasp angle that should be executed at each point. Given the symmetry of the antipodal grasp around the $\pm\pi/2$ radians, the angles are provided in the range of $[-\pi/2, \pi/2]$.
- 3) \mathbf{W} indicates the width of the gripper to be employed at each point. To achieve depth invariance, we set the range of the variable w to $[0, 150]$ pixels, which can be converted to a physical measurement utilizing the depth camera parameters and the measured depth.

Instead of sampling the input image to create grasp candidates, the grasp point \mathbf{g}' is computed for each pixel in the depth image \mathbf{I} directly. Specifically, we define a function \mathbf{M} to retrieve the grasp point from the depth image and convert it to a function \mathbf{M} of the grasp map expressed in the image coordinates: $\mathbf{M}(\mathbf{I}) = \mathbf{G}$. The best grasp is then calculated in the image space $\mathbf{g}'^* = \max_{\mathbf{Q}} \mathbf{G}$, which yields the equivalent best grasp in world coordinates \mathbf{g}^* by (3).

Loss Function: We propose DHRNet to approximate the complex function $\mathbf{M} : \mathbf{I} \rightarrow \mathbf{G}$. Let M_ϕ denote a neural network with ϕ representing the weights of the network. We show that $M_\phi(\mathbf{I}) = (\mathbf{Q}_\phi, \Phi_\phi, \mathbf{W}_\phi) \approx \mathbf{M}(\mathbf{I})$ can be learned with a training set of inputs \mathbf{I}_T and the corresponding outputs \mathbf{G}_T by applying the loss function \mathcal{L} as

$$\theta = \underset{\phi}{\operatorname{argmin}} \mathcal{L}(\mathbf{G}_T, M_\phi(\mathbf{I}_T)). \quad (5)$$

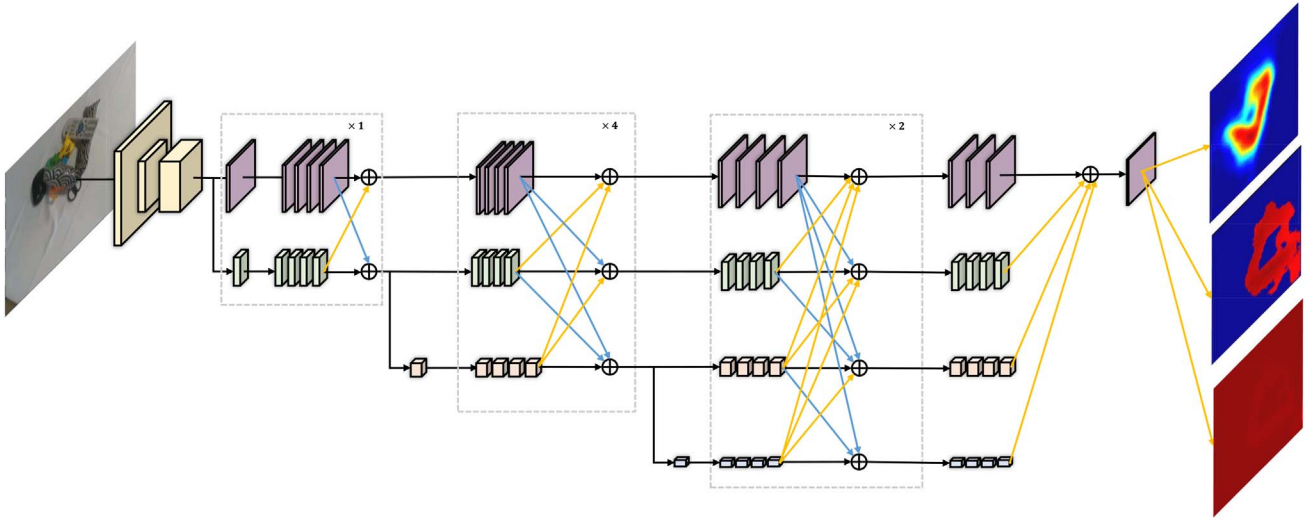


Fig. 2. Network architecture of DHRNet.

B. DHRNet Architecture

In this article, we propose a novel encoder–decoder architecture for robotic perception that distinguishes itself from previous work [9], [12], [18]. Our approach not only achieves better results in the dataset (Table II) but also enables faster extraction of the features that need attention for grasping (Fig. 8) and can effectively deal with thin objects and stacked objects.

As illustrated in Fig. 2, DHRNet stands apart from traditional CNNs by preserving a relatively high-resolution feature map throughout the network. DHRNet is structured into four stages, each containing parallel blocks with varying feature resolutions, and each block is a residual block. We achieve information interaction between different blocks by using a FuseLayer which transitions features between the high-resolution and low-resolution branches via up-sampling and down-sampling operations. Specifically, we employ a convolutional kernel 3×3 with stride 2 to decrease the resolution of the feature map and bilinear interpolation for up-sampling. The high-resolution representation is obtained by parallel mixing different resolution convolutional layers. By parallelly connecting high-to-low-resolution convolutions and iteratively performing fusion operations between parallel blocks, our approach preserves the high-resolution representation.

Our model has two main characteristics: 1) it utilizes parallel connections from high-resolution to low-resolution throughout all phases of the model, and 2) it facilitates the exchange of information across different resolutions to enrich semantic information. The network begins with a convolutional stem block and gradually stacks convolutional blocks with different resolutions, connecting them in parallel. As a result, the features learned by DHRNet are semantically and spatially strong. This is because the convolutional blocks of different resolutions are linked in parallel rather than serially, which is more beneficial for learning accurate spatial position information. Moreover, our model consistently maintains a high-resolution feature representation, instead of shrinking the feature maps as in

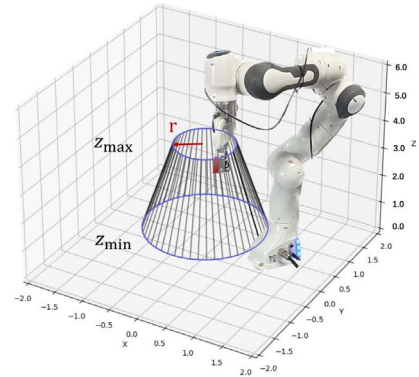


Fig. 3. Set of possible positions of the robot end-effectors in motion planning.

traditional encoders. Furthermore, the information is continuously fused among the different branches, providing a wealth of information at the semantic level.

C. MP-PC

Besides the DHRNet introduced in the previous section for grasping prediction, we also need a closed-loop motion planning algorithm for complex scenarios. Usually, the motion planning algorithm requires the end-effector to operate with a height within the range of z_{\min} to z_{\max} . Suppose the maximum horizontal and maximum vertical velocities of the end-effector are v_1 and v_2 , respectively, and the initial position is in a circle centered at $(0, 0, z_{\max})$ with a radius r , then the set of all possible positions of the end-effector constitutes a circle table as shown in Fig. 3. In existing algorithms, the grasp detection algorithm is usually integrated into the grasp pipeline. When the height of the end-effector is greater than z_{\min} , DHRNet calculates an optimal grasp prediction based on the image from the current point of view t and gives the distribution of information entropy

to calculate the horizontal velocity of the end-effector. This type of motion planning algorithm is often referred to as an active planning algorithm, and a well known method is multiview picking (MVP) [9], which is proven to be effective in solving the clutter and occlusion challenges. However, we found that if the grasping target is thin, the robotic arm can only explore along a random direction at the beginning stage because of the long distance of the camera to the target object and cannot yield correct grasping predictions or information entropy distribution. This can lead to the loss of the target field of view or even singularities. Motivated by this issue, we propose MP-PC to tackle the shortcomings of the previous method. The overall flow chart of MP-PC is shown in Algorithm 1. Before presenting the algorithm in Section III.C.2, we describe how the controller gets the information entropy distribution from the images and how the horizontal velocity v_{xy} of the end-effector is calculated in Section III.C.1.

1) *Information_Gain*: To combine observations along the viewpoint trajectory, we first represent the workspace of the robot as a two-dimensional raster grid map M . The grid map consists of $J \times K$ cells and each cell corresponds to a physical region of size $u \times u$. Within each cell (j, k) , the grasp quality observations q is discretized into N_q intervals and represented in a vector $\mathbf{q}_{j,k}$. Similarly, combined grasp quality and angle observations (q, θ) are discretized into a two-dimensional histogram $\mathbf{m}_{j,k}$, consisting of $N_q \times N_\theta$ intervals. These vectors capture the distribution of observations within each point and serve as the foundation for our information gain approach.

A grasp in the cell (j, k) is parameterized by the mean of the observations within the cell

$$\mathbf{g}_{j,k} = (\mathbf{c}_{j,k}, \bar{\theta}_{j,k}, \bar{w}_{j,k}, \bar{q}_{j,k}) \quad (6)$$

where $\mathbf{c}_{j,k}$ is the physical position at the center of the cell and the mean observations for a cell are calculated as

$$\bar{q} = \frac{1}{\sum \mathbf{q}} \sum_{n_q=1}^{N_q} \frac{n_q}{N_q} \mathbf{q}_{n_q}. \quad (7)$$

The mean angle $\bar{\theta}$ and mean grasp width \bar{w} can be calculated by weighting the corresponding grasp quality observations as

$$\bar{\theta} = \arctan \frac{\sum_{n_\theta=1}^{N_\theta} \sum_{n_q=1}^{N_q} \sin\left(\frac{n_\theta}{N_\theta} \pi\right) \frac{n_q}{N_q} \mathbf{m}_{n_\theta, n_q}}{\sum_{n_\theta=1}^{N_\theta} \sum_{n_q=1}^{N_q} \cos\left(\frac{n_\theta}{N_\theta} \pi\right) \frac{n_q}{N_q} \mathbf{m}_{n_\theta, n_q}}, \quad (8)$$

$$\bar{w} = \frac{1}{n} \sum w. \quad (9)$$

Our controller is based on an information gain approach, where viewpoints are selected to minimize the uncertainty in the grasp pose observations. More specifically, we aim to decrease the entropy of grasp quality observations in M that correspond to high-quality grasps. The entropy of grasp quality observations in a single grid cell can be computed as

$$H(\mathbf{q}) = - \sum_{n_q=1}^{N_q} \frac{\mathbf{q}_{n_q}}{\sum \mathbf{q}} \log \left(\frac{\mathbf{q}_{n_q}}{\sum \mathbf{q}} \right). \quad (10)$$

Entropy in the distribution of grasp quality measurements emerges as a promising indicator for predicting informative

viewpoints. To simplify the calculation of the expected information gain, we adopt the widely accepted assumption that the total entropy of the observed region will provide a good approximation to calculate the expected information gain [30]. In essence, viewpoints that provide high-entropy observations are anticipated to be more informative, as they are capable of reducing entropy more effectively than those that provide low-entropy observations. Consequently, we estimate the expected information gain for an observation at a given viewpoint by computing the weighted sum of entropy of the grid cells that are visible from that viewpoint. In (11), O_t refers to the set of coordinates of the visible grid cells from viewpoint \mathbf{t} , while $P(j, k)$ assigns weights to the points based on a Gaussian function, computed from their distance to the geometric center of O . This approach incentivizes the controller to prioritize front-on views of high-entropy areas over peripheral views. To predict the optimal viewpoint for the next observation, we calculate the utility of relocating the viewpoint directly above each cell in M . This utility approximates the desirability of each potential viewpoint and is computed using (12)

$$E[I(M, t)] \approx \sum_{j,k \in O_t} P(j, k) \cdot H(\mathbf{q}_{j,k}) \quad (11)$$

$$U(M, t) = E[I(M, t)] - \gamma r(t). \quad (12)$$

The effectiveness of selecting the viewpoint is determined by the potential benefits and costs associated with the motion to that viewpoint, as indicated by the function

$$r(t) = \|t_{xy} - \mathbf{c}_{xy}\| \cdot \left(1 - \frac{t_z - z_{\min}}{z_{\max} - z_{\min}} \right). \quad (13)$$

The value of this function is determined by the exploration cost parameter γ , which can be adjusted to balance the tradeoff between exploration and exploitation. To incentivize the controller to prioritize areas near the best detected grasp, rather than exploring irrelevant and distant points, the cost is calculated as the horizontal distance from the current viewpoint to the grid cell with the highest average grasp quality, centered at position \mathbf{c} .

The second term in the cost function is determined by the vertical position of the camera in the trajectory, denoted as t_z . At the start of the trajectory, the cost is set to zero to encourage exploration of the workspace. As the end-effector descends, the cost increases linearly, motivating the controller to converge to the optimal grasp. To move in the direction of maximum utility, the horizontal velocity generated by the controller is $v_{xy} = U(M, \mathbf{t})$.

2) *Algorithm*: The MP-PC process is outlined in Algorithm 1. In the beginning, the height z_{\max} of the robot's end-effector is fixed and the horizontal position is roughly restricted near the center (x, y) of the workspace. MP-PC is used to plan the trajectory of the robot arm when the height of the end-effector is greater than z_{\min} . During the motion of the manipulator, if the position of the end-effector is higher than z_{\min} , it will continuously call DHRNet. If there is no peak in the grasping quality map \mathbf{Q} , a conservative method will be adopted for horizontal detection. Here, we use the 3×3 matrix as a filter and do convolution directly with the grasp detection quality

Algorithm 1: MP-PC

```

Input:  $z_{\min}, z_{\max}, x, y$ 
Output: goal
1 The initial position of robot end-effector:  $\mathbf{t}_0 = (x, y, z_{\max})$ ;
2  $\mathbf{t} = \mathbf{t}_0$ ;
   //  $\mathbf{t}$  is the current position of the robot
   end-effector.
3 while  $t[2] \geq t_{\min}$  do
4    $v_z = \text{const}$ ;
5    $\mathbf{Q}, \mathbf{W}, \Theta = \text{DHRNet}(\mathbf{I})$ ;
   //  $\mathbf{I}$  is current depth image.
6   if exist_peak() then
7      $v_{xy} = U(M, \mathbf{t})$ 
8   else
9      $v_{xy} = \alpha * U(M, \mathbf{t})$ 
10  end
11   $\mathbf{s} = (x^*, y^*) = \arg \max_{(x,y) \in \mathbf{Q}} \mathbf{Q}[x][y]$ ;
12   $w' = \mathbf{W}[x^*][y^*], \theta' = \Theta[x^*][y^*], \mathbf{q} = \mathbf{Q}[x^*][y^*]$ ;
13   $\mathbf{g}' = \{\mathbf{s}, \theta', w', \mathbf{q}\}$ ;
14   $\mathbf{g} = \mathbf{L}_{cw}(\mathbf{L}_{ic}(\mathbf{g}'))$ ;
15  Store  $\mathbf{g}$  in the array  $\mathbf{B}$ ;
16 end
17 find  $g^* = \operatorname{argmax}_{g \in \mathbf{B}} q, \mathbf{B} = \emptyset$ 
18 return  $g^*$ ;
    
```

map \mathbf{Q} to determine whether there is a peak. If there are points greater than the threshold in the new feature map, the detection result is considered reliable, that is, the function *exist_peak()* in Algorithm 1 returns True (lines 6–7 in Algorithm 1). Otherwise, it will aggressively move horizontally in the direction of high entropy (lines 8–9 in Algorithm 1). The robot calculates the position and attitude with the highest possibility of grasping and the width of the gripper according to the quality score map \mathbf{Q} , and transfers it to the grasping representation \mathbf{g} in the world coordinate system O_w through coordinate transformation. When the height is not lower than z_{\min} , the robot arm will save all stored \mathbf{g} into the grabbing experience pool \mathbf{B} , and finally select \mathbf{g} with the highest grasping score q and clear the experience pool \mathbf{B} (lines 11–18 in Algorithm 1).

IV. EXPERIMENT

In this section, we present a series of comprehensive experiments to evaluate our framework by answering the following three questions.

- 1) Is the grasp detection algorithm DHRNet superior to other state-of-the-art grasping detection algorithms?
- 2) What are the benefits of maintaining high resolution feature maps and fusing feature maps of different scales in DHRNet?
- 3) Can the DHRNet and MP-PC based approach successfully grasp unseen, thin or stacked objects?

In the subsequent subsections, we first provide a brief introduction to the dataset and experimental details in Section IV-A. To address the first question, we compare our method with the current state-of-the-art methods [7], [8], [15], [20] on the Cornell and Jacquard datasets in Section IV-B. We demonstrate that our method is effective in improving performance on

TABLE I
PARAMETERS USED DURING EXPERIMENTS

Parameter		Value
J, K	Grid Map Size	68 cm
u	Grid Cell Size	5 mm
N_q	Quality Bins	10
N_ϕ	Angle Bins	18
γ	Exploration Cost	Varied
$ v_z $	End-effector Vertical Velocity (During Reach)	0.05 m/s
α	Attenuation factor	0.2
z_{\max}	initial Height	50 cm
z_{\min}	Final Height	17.5 cm

these datasets. To answer the second question, we conduct two ablation experiments in Section IV-C to evaluate the efficacy of the high-resolution and parallel fusion design. Finally, we evaluate the performance of our framework against other existing frameworks in various complex scenarios under real-world cases (Section IV-D).

A. Dataset and Experiment Setup

The Cornell grasping dataset [12] consists of 885 images with a resolution of 640×480 pixels, each featuring multiple objects that can be grasped. Given the relatively small size of the dataset, various data-augmentation techniques have been utilized to prevent overfitting, such as rotation, zooming, and random cropping. To further evaluate the performance of DHRNet, we have tested it on the Jacquard dataset [13], which was generated using CAD models in a simulator. The Jacquard dataset is significantly larger, containing over 50 000 images spanning 11 000 object categories, with more than 1 million annotated grasp labels available for analysis.

Implementation Details: DHRNet is implemented by PyTorch, and the entire grasp detection system is running on the Ubuntu 18.04 desktop with an Intel Core i9 CPU and a Geforce NVIDIA RTX3090 GPU. For our high-resolution grasp model, we utilize AdamW optimizer [31] with an initial learning rate of $1e^{-4}$ and weight decay of $5e^{-2}$. The input resolution is cropped to 224×224 and the input batch is $I(N, C, H, W)$. The momentum of the BN layer is set to 0.1. The batch size is set as 32 and the epoch is set as 50.

To conduct physical world experiments, we utilized 7 degrees of freedom (DoF) Franka robot equipped with parallel grippers for grasping. To ensure accurate prediction of grasping position and pose, a RealSense D435 RGB-D camera is mounted on the robot's end-effector. For real-time control of the robot, an Ubuntu real-time kernel is used. Additionally, we used a separate computer with a 3.6GHz Intel i7-9700K CPU and GeForce RTX 2080 Super GPU for network deployment and computation. The system is built on top of the Robot Operating System (ROS). The parameters of the variables in the experiment are shown in Table I.

B. Experimental and Analysis in Datasets

The performance of DHRNet is evaluated on the Cornell and Jacquard datasets. To mitigate the risk of overfitting

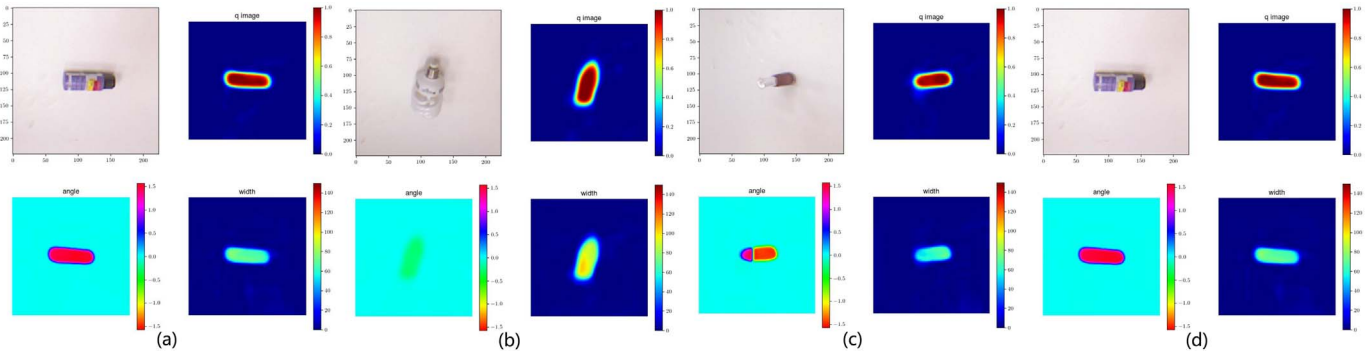


Fig. 4. (a)-(d) are heatmap demonstrations of the samples in the Cornell dataset under the DHRNet prediction. The upper left corner is the original image, the upper right corner is the image of grasping quality Q predicted by DHRNet, the lower left corner is the image of grasping width W , and the lower right corner is the image of grasping angle Θ .

TABLE II
ACCURACY ON CORNELL GRASPING DATASET

Method	Channel	Accuracy (%)	
		IW	OW
Fast Search [5]	RGB-D	60.5	58.3
GG-CNN [9]	D	73.0	69.0
SAE [12]	RGB-D	73.9	75.6
ResNet-50x2 [18]	RGB-D	89.2	88.9
AlexNet, MultiGrasp [6]	RGB-D	88.0	87.1
STEM-CaRFs [32]	RGB-D	88.2	87.5
GRPN [33]	RGB	88.7	-
Two-stage closed-loop [34]	RGB-D	85.3	-
GraspNet [20]	RGB-D	90.2	90.6
ZF-net [7]	RGB-D	93.2	89.1
E2E-net [10]	RGB	98.2	-
GR-ConvNet [8]	D	93.2	94.3
	RGB	96.6	95.5
	RGB-D	97.7	96.6
TF-Grasp [15]	D	95.2	94.9
	RGB	96.78	95.0
	RGB-D	97.99	96.7
DHRNet	D	98.86 ± 0.43	96.5 ± 0.47
	RGB	98.05 ± 0.29	96.5 ± 0.43
	RGB-D	99.17 ± 0.37	97.3 ± 0.82

due to the small size of the dataset, we performed a fivefold cross-validation using the evaluation criteria from the image perspective (IW) and the object perspective (OW), as in the works of [6], [12], [18]. The input patterns and run times are compared and presented in Table II. Our proposed DHRNet achieved the best performance on the Cornell dataset for three different channels, RGB, D, and RGB-D, and two different evaluation criteria (imagewise and objectwise). This indicates that our method has better feature extraction capability and robustness. We also visualized part of the validation images from the Cornell dataset to demonstrate the performance of DHRNet. Fig. 4 presents the experiment results, displaying both the original image and the corresponding heatmap, which highlights the recommended grasp positions and their respective widths and angles. To evaluate the real-time performance of DHRNet in physical experiments, we performed a run-time analysis and compared it with existing methods. Fig. 5 presents the results of this analysis. Although our architecture retains high-resolution feature maps and concatenates branches, no significant

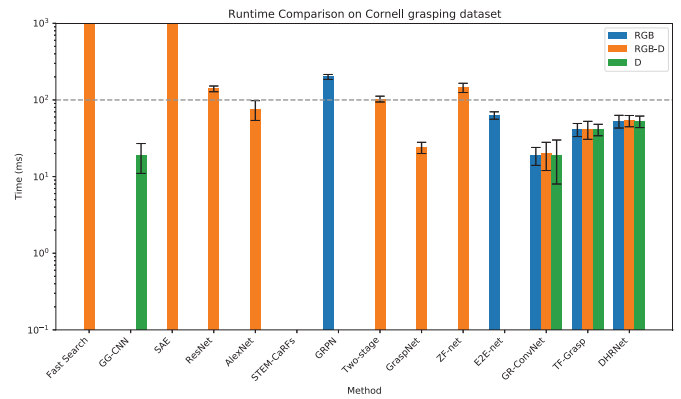


Fig. 5. Run times of the different methods. Run times less than 100 ms (corresponding to the yellow dashed line in the figure) do not affect motion planning.

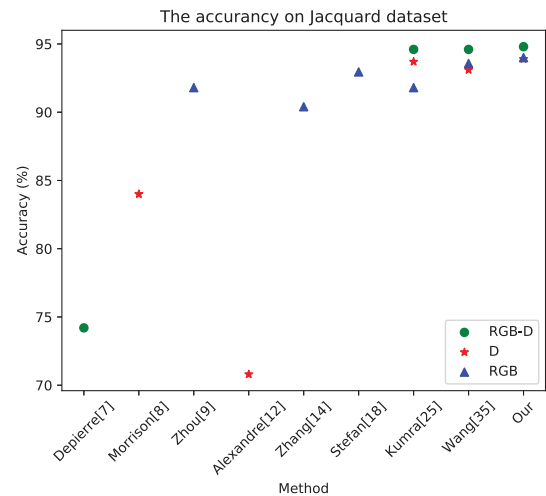


Fig. 6. Accuracy of different methods on the Jacquard dataset.

run-time advantage is observed over existing methods. However, we have improved the accuracy while maintaining real-time performance. We established a threshold value of 100 ms to guarantee real-time performance, and our method successfully meets this requirement, as shown in the analysis.

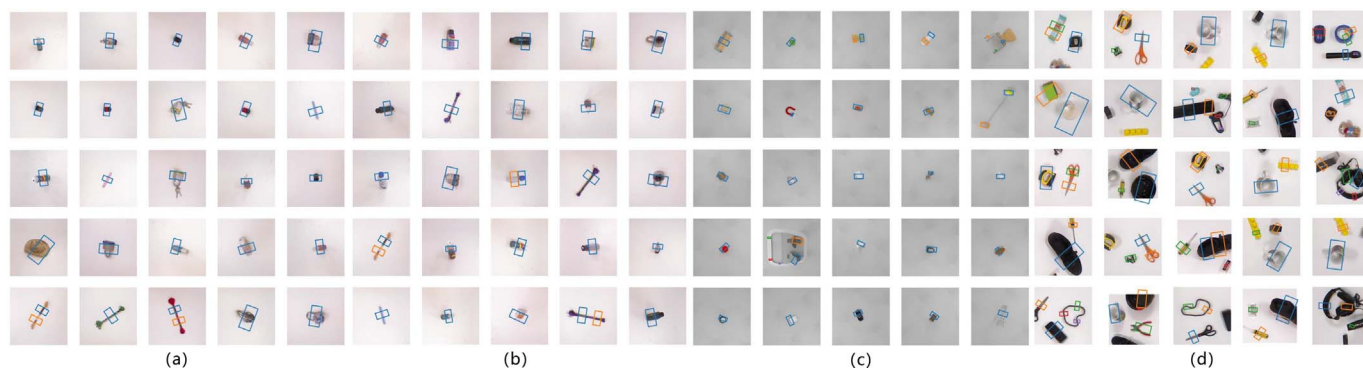


Fig. 7. Visualization of DHRNet prediction results on multiple datasets, where (a) and (b) is the Cornell dataset, (c) is the Jacquard dataset, and (d) is the Multi-object dataset. The top five rectangles, highlighted in blue, orange, green, red, and purple, correspond to the locations with the highest probability in the grasping prediction heatmap \mathbf{Q} . If the rectangles are too close, only the ones with the higher probability of grasping are displayed using nonmaximum value suppression.

The Jacquard dataset is a synthetic dataset comprising 54 000 distinct scenes with 11 000 objects and 1.1 million captured annotations. We partitioned this dataset into a 90% training set and a 10% validation set. As shown in Fig. 6, our method demonstrated high performance on all three channels of RGB, Depth, and RGB-D. Moreover, the depth images and RGB images exhibit a close correlation on this dataset, as the objects and background are relatively easy to distinguish and align closely with the contour boundaries of the objects.

To show the effectiveness of DHRNet, we visualize the predicted grasping rectangles across multiple datasets in Fig. 7. We use the network parameters trained on the Cornell dataset, and the figures show the results for the Cornell dataset [Fig. 7(a) and 7(b)], the Jacquard dataset [Fig. 7(c)], and a multiobject dataset [Fig. 7(d)]. These results clearly demonstrate that our method outperforms existing methods on these datasets.

C. Ablation Studies

Our DHRNet is conceptually different from the traditional design in the robotics community. In contrast to the conventional models that build serial convolutions from high-resolution feature maps to low-resolution maps, our grasping model always maintains high-resolution representation through the parallel branch. To evaluate the role of each component, ablation experiments are carried out on the DHRNet. Fig. 8 shows the precision achieved by the network with different architectures after 10 ten epochs of training, where the first is the fully convolutional neural network (FCN) [35]. Clearly, it achieves an accuracy of about 60% after 10 ten epochs. By adding a residual block [36], the precision is significantly improved because it can enhance the propagation of the gradient and effectively avoid network degradation. It achieves an accuracy of 73% after ten epochs. In contrast to adding residual blocks to the original CNN, we utilize the U-Net as the backbone in our approach and achieve a similar performance to that of the FCN with residual blocks after ten epochs. The original CNN architecture is susceptible to information loss due to the pooling of layers and fixed receptive fields of the convolutional kernel. The introduction of residual connections in our DHRNet model addresses this issue and improves CNN's performance. The

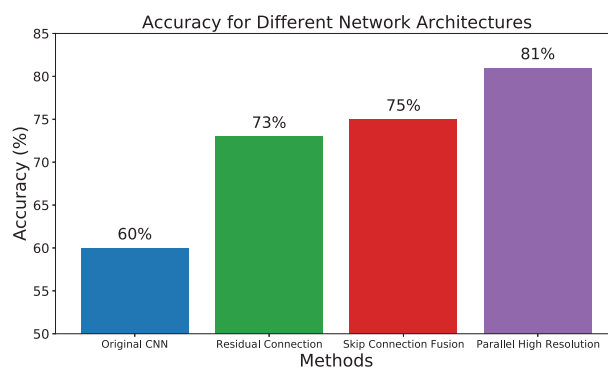


Fig. 8. Accuracy for different network architectures after ten epochs in Cornell datasets.

U-Net architecture employs skip connections to merge detailed information from lower layers to higher layers, facilitating feature fusion and achieving the best results in a nonparallel architecture. However, all the aforementioned methods undergo downsampling and upsampling processes, inevitably resulting in spatial information loss when the objects are small and may occupy only a few pixels. Such downsampling may lead to incorrect grasp predictions. To overcome this limitation, DHRNet maintains several feature maps of different sizes simultaneously and allows information interaction between feature maps of different sizes. Our approach achieves an accuracy of 81% with only ten epochs of training, which is significantly better than other architectures. This is because our approach leverages rich semantic information from the low-resolution feature maps while preserving the accurate spatial information embedded in the high-resolution feature maps.

In order to investigate the impact of fused information from the last feature, we performed an additional ablation experiment using the architecture depicted in Fig. 9. We evaluated the performance in the Cornell and Jacquard datasets using the structure shown in Fig. 10 and observed that layer fusion significantly enhances the prediction capacity of the network across all three channels of RGB, RGB-D, and depth. Our experiments demonstrate that incorporating high-resolution representation and parallel branches in our network not only accelerates the network training but also enhances its performance.

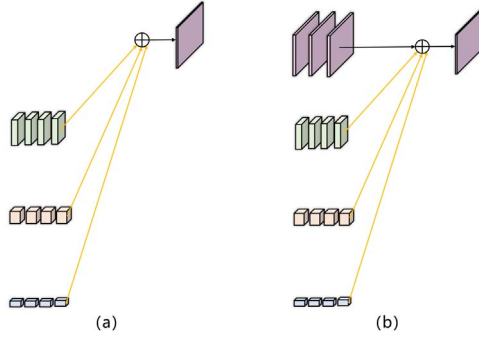


Fig. 9. (a) Features of all branches except the highest resolution branch are combined to predict the corresponding grasp configuration. (b) Features of all different resolution branches are combined to predict the corresponding grasp.

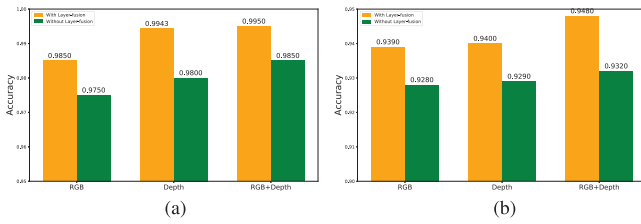


Fig. 10. Performance on the Cornell and Jacquard datasets w/o layer fusion. Ablation study in (a) Cornell and (b) Jacquard.

TABLE III
RESULTS FOR PHYSICAL SETUP

Method	Physical Grasp	Success Rate (%)
Morrison et al. [9]	88/100	88%
Lenz [12]	76/100	76%
Pinto [37]	64/100	64%
Wang [15]	87/100	87%
DHRNet (Ours)	91/100	91%

D. Experiments in Real Physical Environments

In this section, we present the experimental evaluation of our method in comparison to other state-of-the-art methods in physical environments. The results are summarized in Table III. Specifically, to test the robustness of our method, we selected GG-CNN [9] and TF-Grasp [15] methods as control groups. In addition, we conducted experiments in three challenging scenarios in the dataset, including grasping unseen objects, thin objects, and stacked objects. The objective of these experiments was to demonstrate the effectiveness of our proposed approach in challenging real-world scenarios where object recognition and motion planning play a crucial role in achieving successful grasping.

1) *Unseen Object*: To evaluate the generalization capability of our proposed method to unseen objects, we conducted experiments using objects that were not included in the training dataset, as shown in Fig. 11. Specifically, we compared the performance of our method with two baseline methods on these unseen objects, and the results are summarized in Table IV. The experimental results demonstrate that our DHRNet outperforms GG-CNN [9] and TF-Grasp [15] methods in terms of grasp



Fig. 11. Unseen toys not in the datasets.

success rate. Furthermore, we observed that applying the grasp detection algorithm in combination with MP-PC can lead to a slight improvement in the grasp success rate for previously unseen objects compared to the MVP method [9].

2) *Thin Object*: In the experiments of grasping thin objects, we chose a U-shaped disk with the same background color as the target object, making it difficult to distinguish it from the background with RGB or depth information, especially when the robot arm was placed far away from the target object. In the initial stage, the camera is too far away from the object, resulting in no information for all grasp detection methods, which leads to the end-effector randomly choosing a direction to move all the time. If the MVP controller is used, the horizontal speed of the end-effector is very fast and it is very easy to lose the view of the target object before the detection algorithm can detect the object, thus the end-effector continues to move in the original direction until the singularity occurs. This is why robots using MVP controllers are prone to stuck-shield pauses, which reduce grasping efficiency.

To overcome this problem, we use the MP-PC controller, which is able to adopt a conservative strategy to significantly reduce horizontal displacement when there is no peak in the heat map and quickly adjust the strategy to change the direction of horizontal movement after the object is subsequently detected. As shown in Fig. 12, we intercepted the graphs of the grasp quality of different grasp detection methods that intergrade with MP-PC after 2 s of the initial phase. In Fig. 12(a), the GG-CNN still fails to detect the USB flash drive and lacks useful information to guide the motion planning approach. As a result, the robot arm continues to explore in the original direction, and eventually the robot enters a singular point and triggers the protection mechanism, thus failing to grasp it successfully. In contrast, as shown in Fig. 12(b), TF-Grasp is able to detect a more suitable grasping point in the initial stage, but the image is significantly disturbed due to the unevenness of the tablecloth and the small size of the U disk, resulting in the grasping prediction being not necessarily accurate and may find pseudograsping points. Furthermore, as shown in Fig. 12(c), our method has been able to produce relatively accurate predictions at 2 s, enabling it to accurately determine the next viewpoint based on the information obtained from the heat map during execution. DHRNet produced more accurate predictions as the robot approached the target object and was significantly more resistant to interference than the other two methods. Finally,

TABLE IV
SUCCESS RESULT IN UNSEEN OBJECTS

Objects/Methods	GG-CNN+MVP	GG-CNN+MP-PC	TF-Grasp+MVP	TF-Grasp+MP-PC	DHRNet+MVP	DHRNet+MP-PC
Toy1	6/10	6/10	8/10	8/10	8/10	9/10
Toy2	6/10	6/10	7/10	8/10	8/10	10/10
Toy3	5/10	6/10	5/10	7/10	7/10	7/10
Toy4	7/10	8/10	8/10	8/10	9/10	10/10

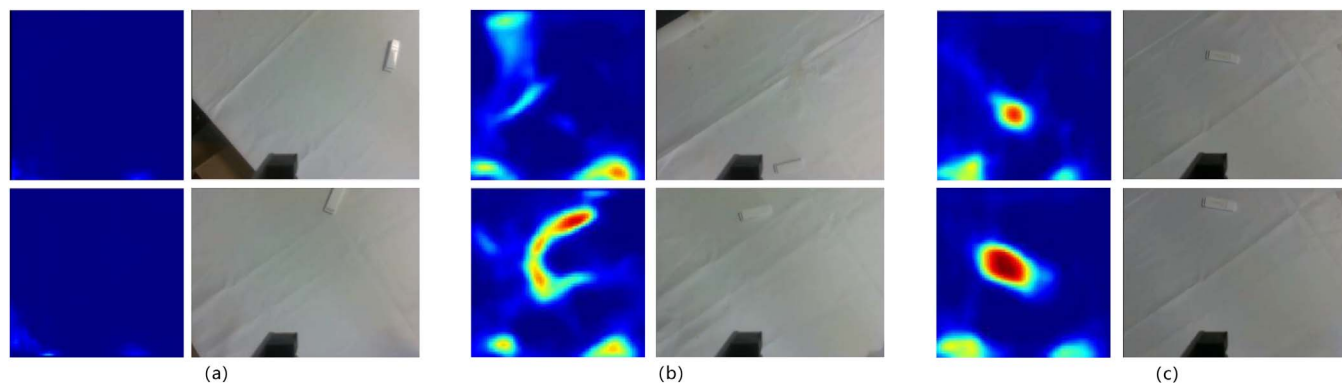


Fig. 12. This study presents the visualization results of different grasp detection methods in the motion planning process. Specifically, we evaluated three methods: (a) GG-CNN; (b) TF-Grasp; and (c) DHRNet. The first row of Fig. 12 displays the heat map and RGB images during the initial phase of motion planning, while the second row displays the heat map and RGB images during the intermediate phase of motion planning.

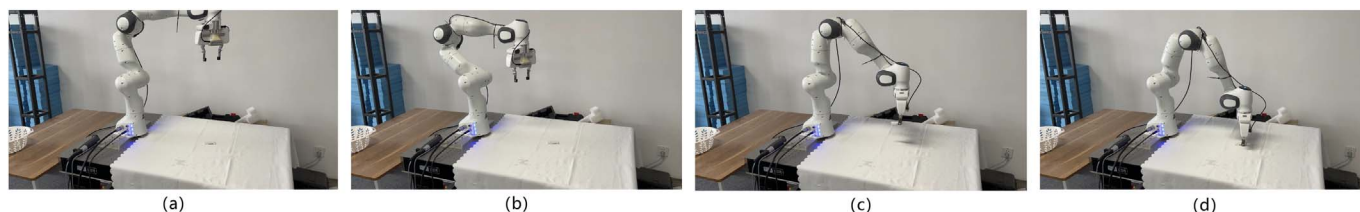


Fig. 13. (a)-(d) Shows the whole process of robot grasping a thin object, for more details refer to the video <https://youtu.be/cfLAdKW04u8>.

TABLE V
RESULT IN THIN OBJECTS

Method	GG-CNN+MVP	GG-CNN+MP-PC	TF-Grasp+MVP	TF-Grasp+MP-PC	DHRNet+MVP	DHRNet+MP-PC
Rate	3/10	6/10	4/10	6/10	7/10	9/10

Fig. 13 depicts the grasping process in the experiment, showing that DHRNet achieves smooth trajectories when dealing with even thin objects. Overall, our method DHRNet outperforms the other two methods in detecting and grasping thin objects.

In the experiments, we identified two significant factors strongly associated with the success rate of our approach. The flatness of the tablecloth plays a pivotal role. Given the relatively low height of the USB flash drive, small deviation from a flat tablecloth surface can lead to misinterpretation by our DHRNet. That is, the network may erroneously treat elevated portions of the tablecloth as the target object. This sensitivity to depth information is another key characteristic of our network. During the initial design phase, there is a conflict between false detection and nondetection when dealing with objects at varying depths. In this context, we consciously opted for prioritizing false detection over nondetection, so that the network’s sensitivity to object depth would not result in missed detections. Additionally, the cleanliness of the table surface has a notable

impact on the input to the RGB camera and, subsequently, the accuracy of the prediction results. To mitigate this issue, we flatten the tablecloth and incorporate a separate depth camera as input source. The specific outcomes of these adjustments are detailed in Table V.

3) *Stacked Object*: For the task of grasping stacked objects and placing them in the basket, different methods are compared based on the time elapsed in completing the task. In order to minimize the impact of the initial conditions, we followed the same stacking pattern each time. Table VI shows that, although all methods were able to complete the task, DHRNet can complete the task faster than the other two grasp detection algorithms with the same motion planning controller. We believe this is due to the fact that DHRNet can maintain high resolution feature maps and fuse feature maps of different sizes, so that it can provide more accurate grasp predictions and avoid collisions during grasping. Comparing MP-PC and MVP, we observe that MP-PC slightly outperforms MVP, although the

TABLE VI
RESULT IN STACKED OBJECTS

Time/Methods	GG-CNN+MVP	GG-CNN+MP-PC	TF-Grasp+MVP	TF-Grasp+MP-PC	DHRNet+MVP	DHRNet+MP-PC
Second	721	555	421	255	168	128

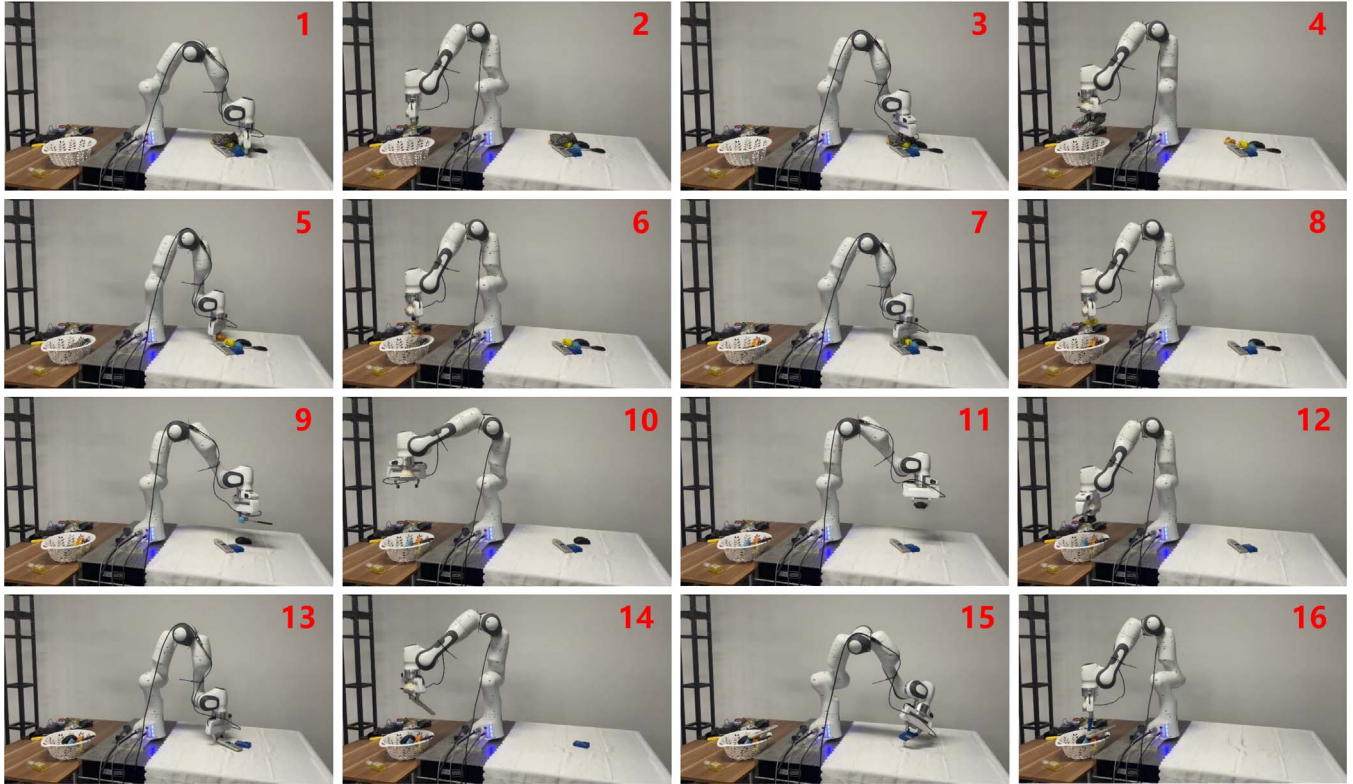


Fig. 14. (1)-(16) Shows the whole process of the robot grasping the stacked objects, for more details refer to the video <https://youtu.be/biuGoTSoupU>.

difference is subtle. This is because all three grasp detection algorithms produce relatively good grasp quality maps, resulting in a low probability of entering singularities. Therefore, the main difference is caused by the grasp detection algorithm, rather than motion planning methods.

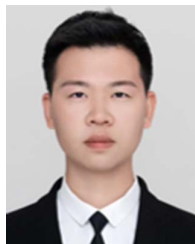
V. CONCLUSION

This article presents a novel framework for robotic visual grasping that leverages high-resolution representations and parallel branches to improve the performance of perception tasks. Our approach diverges from traditional stacked convolutional layers by utilizing high-to-low-resolution convolution streams and fusion between different resolutions. Our physical experiments, which include comparisons with mainstream baselines, demonstrate that our method significantly outperforms existing techniques on various datasets. Our study also underscores the importance of accurate spatial information and motion planning for rich semantic information, particularly for objects located far from the initial camera. Future work will focus on exploring challenging scenarios, such as grasping objects with ambiguous depth information.

REFERENCES

- [1] Q. Yu, W. Shang, Z. Zhao, S. Cong, and Z. Li, "Robotic grasping of unknown objects using novel multilevel convolutional neural networks: From parallel gripper to dexterous hand," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 1730–1741, Oct. 2021.
- [2] S. Yu, D.-H. Zhai, Y. Xia, H. Wu, and J. Liao, "SE-ResUNet: A novel robotic grasp detection method," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 5238–5245, Apr. 2022.
- [3] R. Newbury et al., "Deep learning approaches to grasp synthesis: A review," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3994–4015, Oct. 2023.
- [4] Z. Zhou et al., "Local observation based reactive temporal logic planning of human-robot systems," *IEEE Trans. Autom. Sci. Eng.*, early access, Aug. 25, 2023.
- [5] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: Learning using a new rectangle representation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 3304–3311.
- [6] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 1316–1322.
- [7] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi, "A hybrid deep architecture for robotic grasp detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 1609–1614.
- [8] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 9626–9633.
- [9] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *Int. J. Rob. Res.*, vol. 39, nos. 2–3, pp. 183–201, 2020.

- [10] S. Ainetter and F. Fraundorfer, "End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from RGB," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 13452–13458.
- [11] S. Wang, Z. Zhou, H. Wang, Z. Li, and Z. Kan, "Unsupervised representation learning for visual robotics grasping," in *Proc. Int. Conf. Adv. Robot. Mechatron.*, Piscataway, NJ, USA: IEEE Press, 2022, pp. 57–62.
- [12] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Rob. Res.*, vol. 34, nos. 4–5, pp. 705–724, 2015.
- [13] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 3511–3516.
- [14] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "GraspNet-1billion: A large-scale benchmark for general object grasping," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2020, pp. 11444–11453.
- [15] S. Wang, Z. Zhou, and Z. Kan, "When transformer meets robotic grasping: Exploits context for efficient grasp detection," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8170–8177, Jul. 2022.
- [16] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC Press, 2017.
- [17] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, USA, Apr. 2000, pp. 348–353.
- [18] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2017, pp. 769–776.
- [19] H. Zhang, J. Peeters, E. Demeester, and K. Kellens, "Deep learning reactive robotic grasping with a versatile vacuum gripper," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 1244–1259, Apr. 2023.
- [20] U. Asif, J. Tang, and S. Herrer, "GraspNet: An efficient convolutional neural network for real-time grasp detection for low-powered devices," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, vol. 7, pp. 4875–4882.
- [21] X. Zhu, L. Sun, Y. Fan, and M. Tomizuka, "6-DoF contrastive grasp proposal network," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 6371–6377.
- [22] T. Zou et al., "KAM-Net: Keypoint-aware and keypoint-matching network for vehicle detection from 2-D point cloud," *IEEE Trans. Artif. Intell.*, vol. 3, no. 2, pp. 207–217, Apr. 2022.
- [23] X. Li, X. Li, Z. Li, X. Xiong, M. O. Khyam, and C. Sun, "Robust vehicle detection in high-resolution aerial images with imbalanced data," *IEEE Trans. Artif. Intell.*, vol. 2, no. 3, pp. 238–250, Jun. 2021.
- [24] M. B. Abubaker and B. Babayigit, "Detection of cardiovascular diseases in ECG images using machine learning and deep learning methods," *IEEE Trans. Artif. Intell.*, vol. 4, no. 2, pp. 373–382, Apr. 2023.
- [25] J. Mahler et al., "Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proc. Robot.: Sci. Syst.*, Cambridge, MA, USA, Jul. 12–16, 2017.
- [26] A. Gariépy, J.-C. Ruel, B. Chaib-Draa, and P. Giguere, "GQ-STN: Optimizing one-shot grasp detection based on robustness classifier," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 3996–4003.
- [27] D. S. D. Cortes, G. Hwang, and K.-U. Kyung, "Imitation learning based soft robotic grasping control without precise estimation of target posture," in *Proc. IEEE 4th Int. Conf. Soft Robot. (RoboSoft)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 149–154.
- [28] Y. Yang et al., "GP-Net: A lightweight generative convolutional neural network with grasp priority," *APSIPA Trans. Signal Inf. Proc.*, vol. 12, no. 1, e26, 2023.
- [29] M. Gualtieri and R. Platt, "Viewpoint selection for grasp detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Piscataway, NJ, USA: IEEE Press, 2017, pp. 258–264.
- [30] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.
- [31] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, May 6–9, 2019. [Online]. Available: <https://openreview.net/forum?id=Bkg6RiCqY7>
- [32] U. Asif, M. Bennamoun, and F. A. Sohel, "RGB-D object recognition and grasp detection using hierarchical cascaded forests," *IEEE Trans. Robot.*, vol. 33, no. 3, pp. 547–564, Jun. 2017.
- [33] H. Karaoguz and P. Jensfelt, "Object detection approach for robot grasp detection," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019, pp. 4953–4959.
- [34] Z. Wang, Z. Li, B. Wang, and H. Liu, "Robot grasp detection using multimodal deep convolutional neural networks," *Adv. Mech. Eng.*, vol. 8, no. 9, 2016, Art. no. 1687814016668077.
- [35] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 379–387.
- [36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [37] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2016, pp. 3406–3413.



Zhangli Zhou received the B.E. degree from Wuhan University of Technology, Wuhan, China, in 2016. He is currently working toward the Ph.D. degree with the University of Science and Technology of China, Hefei, China.

His research interests include robotics and reactive task and motion planning.



Shaochen Wang (Graduate Student Member, IEEE) received the B.E. degree from Hefei University of Technology, Hefei, China, in 2016. He is currently working toward the Ph.D. degree in control science and engineering with the University of Science and Technology of China, Hefei, China.

His current research interests include reinforcement learning and robotics.



Ziyang Chen received the B.E. degree from Beihang University, Beijing, China, in 2020. He is currently working toward the Ph.D. degree in control science and engineering with the University of Science and Technology of China, Hefei, China.

His current research interests include robotics and multiagent systems.



Mingyu Cai received the Ph.D. degree in mechanical engineering from the University of Iowa, Iowa City, IA, USA, in 2021.

From 2021 to 2023, he was a Postdoctoral Associate with the Department of Mechanical Engineering at Lehigh University, Bethlehem, USA. He worked at Honda Research Institute, San Jose, CA, USA, as a Research Scientist. From 2024, he is an Assistant Professor with the University of California, Riverside, Riverside, CA, USA. His research interests encompass robotics, machine learning, control theory, and formal

methods, with a focus on applications in motion planning, decision-making, nonlinear control, and autonomous driving.



Zhen Kan (Senior Member, IEEE) received the Ph.D. degree from the Department of Mechanical and Aerospace Engineering at the University of Florida, Gainesville, FL, USA, in 2011.

He was a Postdoctoral Research Fellow with the Air Force Research Laboratory (AFRL), Shalimar, FL, USA, Eglin AFB, and the University of Florida REEF, Shalimar, FL, USA, from 2012 to 2016, and was an Assistant Professor with the Department of Mechanical Engineering at the University of Iowa, Iowa City, IA, USA, from 2016 to 2019. He is currently a Professor with the Department of Automation, University of Science and Technology of China, Hefei, China. His research interests include networked control systems, nonlinear control, formal methods, and robotics.

Dr. Kan is an Associate Editor of IEEE TRANSACTIONS ON AUTOMATIC CONTROL and IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS. He currently serves on program committees of several internationally recognized scientific and engineering conferences.