High-Performance Pixel-Level Grasp Detection Based on Adaptive Grasping and Grasp-Aware Network

Dexin Wang, Chunsheng Liu, *Member, IEEE*, Faliang Chang, Nanjun Li, and Guangxin Li

***Abstract*—Machine vision-based planar grasping detection is challenging due to uncertainty about object shape, pose, size, etc. Previous methods mostly focus on predicting discrete gripper configurations, and may miss some ground-truth grasp postures. In this article, a pixel-level grasp detection method is proposed, which uses deep neural network to predict pixel-level gripper configurations on RGB images. First, a novel oriented arrow representation model (OAR-model) is introduced to represent the gripper configuration of parallel-jaw and three-fingered gripper, which can partly improve the applicability to different grippers. Then, the adaptive grasping attribute model is proposed to adaptively represent the grasping attribute of objects, for resolving angle conflicts in training and simplifying pixel-level labeling. Lastly, the adaptive feature fusion and grasp-aware network (AFFGA-Net) is proposed to predict pixel-level OAR-models on RGB images. AFFGA-Net improves the robustness in unstructured scenarios by using hybrid atrous spatial pyramid and adaptive decoder connected in sequence. On the public Cornell dataset and actual objects, our structure achieves 99.09% and 98.0% grasp detection accuracy, respectively. In over 2400 robotic grasp trials, our structure achieves an average success rate of 98.77% in single-object scenarios and 93.69% in cluttered scenarios. Moreover, AFFGA-Net completes a grasp detection pipeline within 15 ms.**

***Index Terms*—Adaptive grasping attribute model (AGA-model), convolutional neural network, grasp detection, oriented arrow representation (OAR).**

I. Introduction

With the advantage of high automation, robot grasping is widely used in industrial production [1], [2]. Reliable robotic grasping is challenging due to uncertainty about object shape, pose, size, etc. Recently, many algorithms have claimed to be effective for handling stacked scenarios as well as grasping novel objects [3], [4]. However, how to represent a gripper configuration and what is the output format of the learning algorithm are still open questions.

How to represent a gripper configuration? A complete gripper configuration includes 3-D location, 3-D orientation, and gripper grasp width, etc. [5]. Because predicting all variables is complicated, this task is often simplified to predict a "representation" of the gripper configuration, which is a projection of the gripper configuration on a plane. Jiang *et al.* [6] used an oriented rectangle to represent the gripper configuration of the parallel-jaw gripper; this method sets the gripper jaw size as an empirical value, but the jaw size is not potentially related to the object size, which makes the neural network confused. Mahler *et al.* [7] used a point and an angle to represent the gripper configuration; this method sets the grasp width as a constant, which limits the size of objects that can be grasped. Moreover, they are only applicative for the parallel-jaw gripper.

What is the output format of the learning algorithm? We argue that the output format of the learning algorithm is the grasping attribute of the object, i.e., the sum of all grasp configurations that an object can be grasped by a gripper. Most methods based on rectangle representation predict multiple discrete oriented rectangles on RGB images [8], [9]. Besides, Mahler *et al.* [7], [10], [11] first sampled some gripper configurations and, then, evaluated their confidence. However, the feasible gripper configurations on the object are continuous, and these methods may miss some ground-truth grasp postures.

To overcome these existing problems, we propose a pixel-level grasp detection method to generate pixel-level gripper configurations for parallel-jaw and three-fingered gripper, which consists of three parts, including oriented arrow representation model (OAR-model), adaptive grasping attribute model (AGA-model), and adaptive feature fusion and grasp-aware network (AFFGA-Net).

First, we design the OAR-model to improve the learnability and applicability of the grasp representation. By simplifying a three-fingered gripper into a parallel-jaw gripper with two jaws with different sizes, the OAR-model is applicative for the parallel-jaw gripper and the simplified three-fingered gripper. The OAR-model avoids the confusion of the neural network learning by constraining the size of the gripper jaw, and is

Manuscript received May 26, 2021; revised August 28, 2021 and September 22, 2021; accepted October 4, 2021. Date of publication October 26, 2021; date of current version June 6, 2022. This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1305300, in part by the National Natural Science Foundation of China under Grant 62176138 and Grant 62176136, and in part by Shandong Provincial Key Research and Development Program (Major Scientific and Technological Innovation Project) under Grant 2019JZZY010130 and Grant 2020CXGC010207. *(Corresponding authors: Faliang Chang; Chunsheng Liu.)*

The authors are with the School of Control Science, and Engineering, Shandong University, Jinan, Shandong 250061, China (e-mail: dexinwang@mail.sdu.edu.cn; liuchunsheng@sdu.edu.cn; flchang@sdu.edu.cn; 201613124@mail.sdu.edu.cn; liguangxin@mail.sdu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2021.3120474.

Digital Object Identifier 10.1109/TIE.2021.3120474

applicative for objects with different sizes by using variable grasp width.

Second, to optimize the learning process of network, we propose the AGA-model, which represents the grasping attribute of objects. One pixel may correspond to multiple OAR-models with different grasp angles, which may cause angle conflicts. By combining adjacent OAR-models, AGA-model resolves angle conflicts in training and avoids the extremely complicated pixel-level labeling process.

Third, the AFFGA-Net is proposed to generate an OAR-model and confidence at every pixel on RGB images. The pixel-level mapping avoids missing ground-truth grasp postures, and overcomes limitations of current deep-learning grasping techniques including, discrete sampling of grasp candidates and large computation consuming. Moreover, AFFGA-Net is robust for objects with various shapes and sizes, by using hybrid atrous spatial pyramid and adaptive decoder, which can adaptively extract and decode multiscale features.

On the Cornell Grasping dataset [12], our method achieves accuracy of 99.09% and 98.64% on image-wise and object-wise splitting, respectively, and outperforms the latest state-of-the-art approach by 1.35% and 2.03%, respectively. On the randomly selected household objects, our method achieves 98.0% grasp detection accuracy. In over 2400 robotic grasp trials, our structure achieves an average success rate of 98.77% in single-object scenarios and 93.69% in cluttered scenarios. Moreover, AFFGA-Net completes a grasp detection pipeline within 15 ms, which can be used for real-time applications.[1]

The contributions of our study are summarized as follows

1) The OAR-model is designed to represent the gripper configuration of parallel-jaw gripper and simplified three-fingered gripper, which avoids the confusion of the network learning and is applicative for objects with different sizes.
2) The AGA-model is proposed to resolve angle conflicts in training and avoids the extremely complicated pixel-level labeling process.
3) The AFFGA-Net is proposed to generate a pixel-level OAR-model on RGB images, which avoids missing ground-truth grasp postures and reduces computational times.
4) Our structure achieves the state-of-the-art performance on the ***Cornell Grasping dataset and is proved effective for novel objects in cluttered scenarios.

The rest of this article is organized as follows. Section II discusses related grasp detection methods. Section III introduces our method in detail. Section IV introduces our experiment setup. Section V demonstrates detailed experiments and results. Finally, Section VI concludes this article.

II. Related Work

The goal of grasp detection is to find a proper posture using the visual information in different scenarios, and the gripper can stably grasp the target when closing the jaws in this posture. The methods can be roughly divided into the following two categories: 1) analytic methods and 2) empirical methods [13]. Analytic methods use mathematical and physical models of geometry, kinematics, and dynamics to calculate stable grasping parameters [14], [15]; yet, they usually cannot transfer well to the real world due to the difficultly in modeling physical interactions between a manipulator and an object [16]. In contrast, empirical methods do not require object 3-D models; they train the grasp model using known objects and use this model to detect the grasping posture of unknown objects [17]–[19]. In ***recent years, some deep learning-based methods have been designed to first detect planar grasp representation and, then, map the representation to grasp posture in the world coordinate system; these methods usually perform better than traditional empirical methods based on shape primitives [20], [21], machine learning [22], etc.

Grasping representation. A planar grasp representation generally includes grasp point, grasp angle, and grasp width, as least. Saxena *et al.* [23] used supervised learning to predict a grasp point from the image and successfully extend it to new targets. Le *et al.* [24] suggested that a pair of points are used to represent grasping. Jiang *et al.* [6] reduced the dimensionality of the 7-D gripper configuration (the 3-D location, the 3-D orientation, and the distance between the two fingers) in the real environment to obtain a simplified 5-D rectangle grasping representation. These representations suffer from the problem that they are limited to parallel plate gripper.

Network. Previous grasp detection networks are often based on object detection network [25]. Zhou *et al.* [26] used ResNet-50 as the feature extractor and adopted anchor mechanism [27] to predict the 5-D rectangle grasp model, which greatly improves the prediction accuracy. Asif *et al.* [28] overcame limitations of individual models by combining convolutional neural network structures with feature fusion and producing grasping with confidence scores at different levels of the image hierarchy (i.e., global-, region-, and pixel-levels). However, these algorithms cannot generate dense rectangles, which makes it hard to accurately predict the grasping attribute of the object.

III. Our Method

To overcome the limitations of previous grasp detection methods that may miss some ground-truth grasp postures, we propose a pixel-level grasp detection method with three main parts, which is shown in Fig. 1. First, the oriented arrow representation model (OAR-model) is introduced to represent mapping of the gripper configuration on plane. Second, all OAR-models on the object are merged into multiple AGA-models, and the AGA-models are labeled as targets to train the AFFGA-Net. AFFGA-Net inputs an RGB image and outputs an OAR-model at every pixel. Then, the gripper configuration is calculated using the optimal OAR-model and point cloud. Lastly, the robot approaches the target and closes the jaws.

[1]Code is [Online]. Available: https://github.com/liuchunsense/AFFGA-Net.

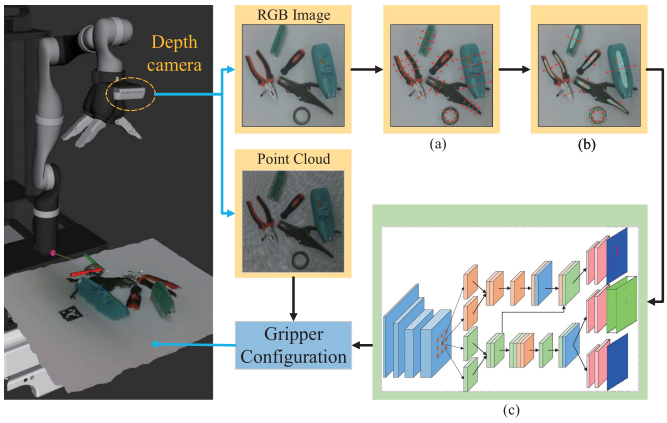


Fig. 1. Our pixel-level grasp detection method.

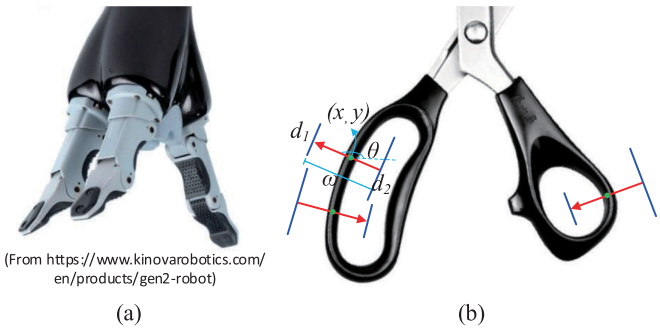


Fig. 2. (a) Three-fingered gripper. (b) Examples of OAR-models.

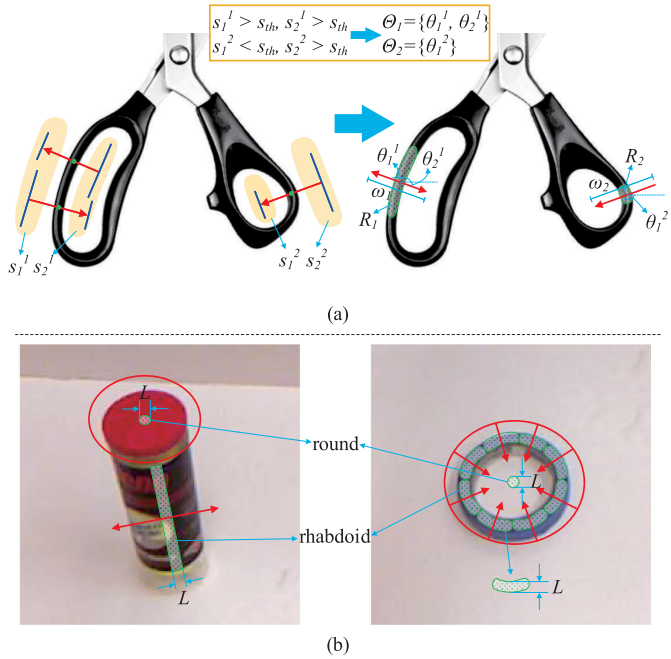


Fig. 3. (a) Illustration of AGA-models. An AGA-model is defined by the grasp region R (green pattern), grasp angle Θ (directions of the red arrow), and grasp width ω (length of the red arrow). The superscript of θ and s is used to distinguish different AGA-models. (b) Examples of rhabdoid and round graspable parts and the AGA-models in the graspable parts. The red circle indicates that $\Theta = \{\theta | 0 \leq \theta < 2\pi\}$. The diameter of the circle represents the grasp width of the AGA-model.

A. Oriented Arrow Representation Model

Unlike general multifingered grippers, each finger of the three-fingered gripper [see Fig. 2(a)] can only apply force toward the center of the gripper. In order to use only one grasp representation to represent both the parallel-jaw gripper and three-fingered gripper, we simplify the three-fingered gripper into a parallel-jaw gripper with two jaws of various sizes by keeping two adjacent fingers moving in sync. The OAR-model is shown in Fig. 2(b) and represented as

$$G_r = \{x, y, \omega, d_1, d_2, \theta\} \tag{1}$$

where (x, y) denotes the grasp point, ω is the grasp width, d_1 and d_2 represent the size of two jaws, respectively ($d_1 < d_2$), and θ is the grasp angle.

Given the OAR-model and the point cloud of object, the 3-D coordinate of the grasp point is the projection point of (x, y) in the point cloud. For a three-fingered gripper, the d_1 position is where the single finger is placed, and the d_2 position is where the other two fingers are placed. For a parallel-jaw gripper, two fingers are placed in the d_1 or d_2 position, respectively. The gripper is perpendicular to the table, and all fingers apply forces toward the grasping point to grasp the object.

Rectangular representation sets the size of the gripper jaw as an empirical value, but the jaw size is not potentially related to the object size, which makes the neural network confused. We set d_1 and d_2 as the mapping of the real size of the three-fingered gripper jaw in the image coordinate system. This constraint avoids the confusion of training a network. Thus, the network can focus on learning the mapping from the object image to other values of the OAR-model. Compared with Dex-Net 2.0 [7], the grasp width of the OAR-model is variable and can be learned to fit objects with different sizes.

B. Adaptive Grasping Attribute Model

There are the following two difficulties with learning pixel-level mapping: (1) one pixel may correspond to multiple OAR-models with different grasp angles, which may cause angle conflicts; and (2) it is extremely time-consuming to label pixel-level OAR-models. In order to solve the abovementioned problems, we propose the AGA-model to model the grasping attribute of the object, which is composed of multiple adjacent OAR-models. The AGA-model transforms single-angle learning tasks into multiangle learning tasks, and transforms pixel-level labeling tasks into region-level labeling tasks.

The AGA-model is shown in Fig. 3(a) and represented as

$$G_m = \{R, \Theta, \omega\} \tag{2}$$

where R represents grasp region, Θ represents grasp angle, and ω represents grasp width.

1) Grasp Region: The points on the object that can be grasped are clustered into multiple regions. The defined grasp region is composed of multiple adjacent grasp points (x, y) on an object, in which these grasp points have the same grasp angle

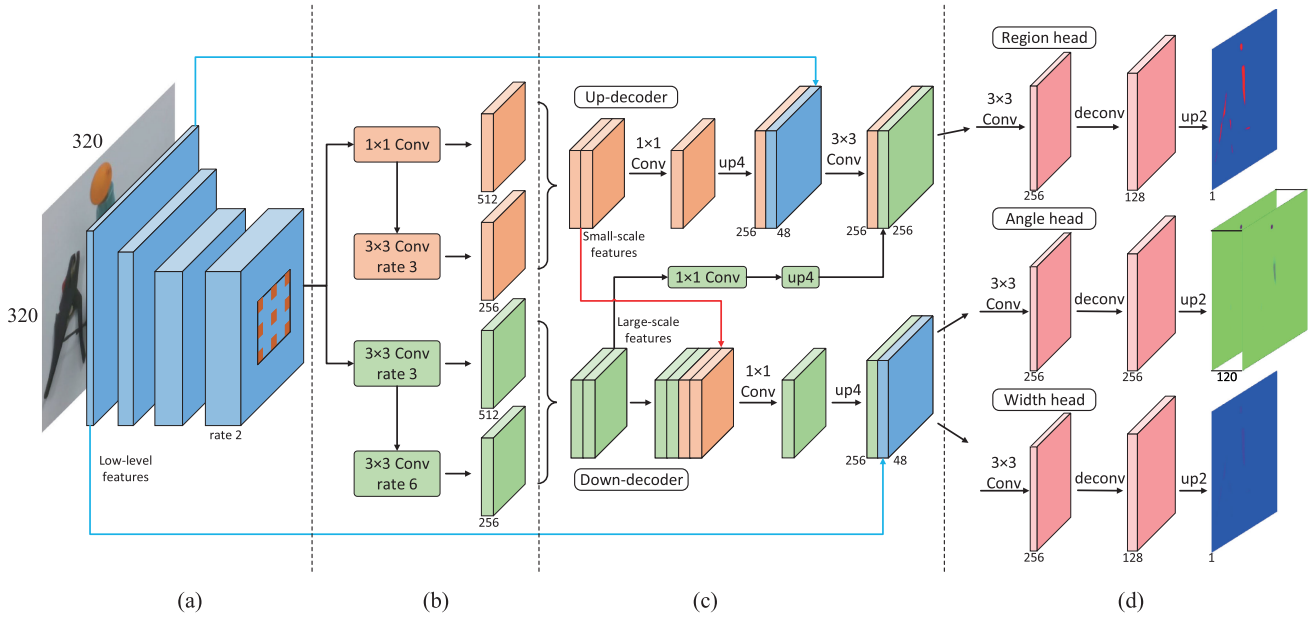


Fig. 4. Structure of the AFFGA-Net. (a) Shared encoder. (b) Hybrid atrous spatial pyramid. (c) Adaptive decoder. (d) Grasping heads.

θ and grasp width ω. The shape of the grasp region is similar to that of the graspable part of the object. To avoid moving the object while grasping, the grasp point is located near the central axis of the object. We limit the maximum width (perpendicular to the central axis of the object) of the grasp region to L (1 cm in this study). For the finer graspable part with a width less than L, the edge of the grasp region is aligned with the graspable part. For the thicker graspable part with a width greater than L, the grasp region is located inside the graspable part [see Fig. 3(b)].

2) Grasp Angle: The parallel-jaw gripper may grasp objects symmetrically when the space around the object can accommodate the gripper jaw. Comparing with a parallel-jaw gripper, the three-fingered gripper needs to additionally consider the different space around objects, such as tape with a small inner ring and scissors' handle. Formally, let S represent the shape of the graspable part of the object. If S is rhabdoid, let s_1 and s_2 denote the space on two sides of the graspable part, respectively. We set the elements of Θ according to the following three situations.

1) When S is rhabdoid, if $\max(s_1, s_2)$ is greater than s_{th} and $\min(s_1, s_2)$ is less than s_{th}, Θ contains only one grasp angle, which is

$$\Theta = \{\theta_1\}, \text{ if } \max(s_1, s_2) \geq s_{\text{th}}, \ \min(s_1, s_2) < s_{\text{th}} \quad (3)$$

where θ_1 points to the small side of the graspable part.

2) When S is rhabdoid, if $\min(s_1, s_2)$ is greater than s_{th}, Θ contains two retrorse grasp angles, which are

$$\Theta = \{\theta_1, \theta_2\} = \{\theta_1, \theta_1 + \pi\}, \text{ if } \min(s_1, s_2) \geq s_{\text{th}} \quad (4)$$

where θ_1 and θ_2 point to the two sides of the graspable part, respectively.

3) When S is round, Θ contains all values between 0 and 2π as

$$\Theta = \{\theta | 0 \leq \theta < 2\pi\}, \text{ if } S = \text{round}. \quad (5)$$

We set $s_{\text{th}} = d_2$ to avoid collision between the gripper and the object. Examples of rhabdoid and round graspable part, and the AGA-model in the graspable part are shown in Fig. 3(b).

3) Grasp Width: The grasp width is an integer value whose definition and labeling method are the same as those in [6].

Taking any point in the grasp region R as the grasp point (x, y), and selecting any element in Θ as the grasp angle θ, an OAR-model is built with the grasp width ω. The AGA-model merges the grasp angles on the OAR-models located at the same location into a set to avoid angle conflicts. Besides, the neural network can be trained by labeling the AGA-model on the dataset, which greatly simplifies the labeling process.

C. Adaptive Feature Fusion and Grasp-Aware Network

Based on the OAR-model and AGA-model, we take the planar grasp detection problem as a pixel-level segmentation problem, and propose a novel AFFGA-Net to fast generate an optimal gripper configuration to guide robot grasping.

Based on deeplabv3+ [29], we propose the following four solutions to build our AFFFGA-Net (see Fig. 4):

1) hybrid atrous spatial pyramid (HASP),
2) adaptive decoder (AD),
3) mixed upsampling,
4) sigmoid.

We introduce the baseline and our solutions separately as follows.

1) Baseline: We retain all the details of the encoder–decoder structure of deeplabv3+ [29], and only modify the final task head to output pixel-level OAR-models. ResNet-101 [30]

is utilized as the backbone of the shared encoder. Three semantic grasping heads are designed and parallelly attached to the decoder, including region head, angle head, and width head, whose output channels are $\{1, K, 1\}$ ($K = 120$ in this study). Region head outputs the confidences that each pixel point is located in the grasp region R. Angle head outputs the category k of the grasp angle corresponding to each point, from which we calculate the grasp angle by $\theta = \frac{2\pi}{K}k$. Width head outputs the grasp width corresponding to each point. The point with the maximal confidence is chosen as the grasp point (x, y), and the grasp angle θ and grasp width ω are the predicted results at the (x, y) position. The optimal OAR-model is built with the grasp point (x, y), grasp angle θ, and grasp width ω. AFFGA-Net decomposes the grasp detection problem into the following three subproblems: 1) grasp region segmentation; 2) grasp angle classification; and 3) grasp width regression.

2) Hybrid Atrous Spatial Pyramid: Chen *et al.* [31] verified that atrous spatial pyramid pooling is effective for improving the segmentation accuracy of multiscale objects. Nonetheless, the large hole rate (rate $= \{6, 12, 18\}$) leads to low information utilization and loss of local features. We design HASP to solve the abovementioned problems.

HASP includes two parallel two-layer feature pyramids, which are used to extract features with different scales, and each pyramid consists of two atrous convolutions connected in series. The cascade structure improves the information utilization rate without reducing the receptive field, and the parallel structure avoids the redundancy between multiscale features. The details of HASP are shown in Fig. 4(b).

3) Adaptive Decoder: In the AGA-model, the grasp region of the thicker graspable part is located inside, and the grasp region of the finer graspable part is aligned with the edge. Therefore, we design AD to avoid the predicted grasp region of thicker objects close to the edge, which is shown in Fig. 4(c).

AD includes two parallel decoder networks, which merge different levels of features in different orders. In the up-decoder, the small-scale features are first concatenated with the low-level features from the backbone and, then, concatenated with large-scale features. In the down-decoder, the large-scale features are first concatenated with the small-scale features and, then, concatenated with the low-level features. The difference in feature-fusion order makes the output of the up-decoder not contain the edge information of large-scale objects, while the output of the down-decoder contains all the information of objects with different scales. The features output by the up-decoder are input into the region head to predict the grasp region. The features output by the down-decoder are used to predict the grasp angle and grasp width, because the grasp angle and grasp width on objects with different scales are all related to accurate edge information.

4) Mixed Upsampling: The decoder features are computed with output stride $= 4$. Since the grasp region is smaller than the object mask, we use a 3×3 deconvolution followed by a bilinear upsampling to accurately restore the grasp region.

5) Sigmoid: Predicting grasp angle is a multilabel single-classification task. We use a sigmoid function to normalize the output of the angle head to avoid competition among categories.

D. Loss Function

We calculate the loss separately for the output of each head and use the sum of losses to optimize AFFGA-Net.

1) Grasp Region: Predicting grasp region is a binary classification problem. We first use the sigmoid function to normalize the prediction results followed by binary cross-entropy function (BCE) to calculate the loss, which is defined as follows:

$$L_{\text{reg}} = -\frac{1}{N}\sum_{n=0}^{N}[y_q^n \cdot \log(p_q^n) + (1 - y_q^n) \cdot \log(1 - p_g^n)] \quad (6)$$

where N is the size of output feature maps, p_q^n is the predicted probability at the nth position, and y_q^n is the corresponding label.

2) Grasp Angle: After using the sigmoid function to normalize the output of the angle head, the BCE function is utilized to calculate the grasp angle loss, which is defined as

$$L_{\text{ang}} = -\frac{1}{N \times L}\sum_{n=0}^{N}\sum_{l=0}^{L}[y_l^n \cdot \log(p_l^n) + (1 - y_l^n) \cdot \log(1 - p_l^n)] \quad (7)$$

where p_l^n represents the probability that the predicted grasp angle is within $[\frac{l}{L} \times 2\pi, \frac{l+1}{L} \times 2\pi]$ at the n position and y_l^n is the corresponding label. We show in Section V-A that using sigmoid instead of softmax increases accuracy.

3) Grasp Width: Predicting the grasp width is a regression problem. We use the BCE function to calculate the loss of the grasp width branch as follows:

$$L_{\text{wid}} = -\frac{1}{N}\sum_{n=0}^{N}[y_w^n \cdot \log(p_w^n) + (1 - y_w^n) \cdot \log(1 - p_w^n)] \quad (8)$$

where p_w^n is the predicted grasp width at the n position and y_w^n is the corresponding label.

4) Multitask Loss: In order to balance the loss of each branch, the final multitask loss is defined as

$$L_{\text{total}} = \gamma_1 \times L_{\text{reg}} + \gamma_2 \times L_{\text{ang}} + \gamma_3 \times L_{\text{wid}} \quad (9)$$

where γ_1, γ_2, and γ_3 are weight coefficients of the loss. In our study, we experimentally set $\gamma_1 = 1, \gamma_2 = 10$ and $\gamma_3 = 5$.

IV. Experimental Setup

The performance of the proposed method is evaluated on the Cornell Grasping dataset [12] and a test set captured in actual scenarios. The grasp detection accuracy and robotic grasp success rate are selected as the main performance metric. We also propose a simplified method to label AGA-model for training AFFGA-Net.

A. Training Dataset

To train and test AFFGA-Net, we create the two following datasets.

1) *Cornell dataset:* There are 878 images in the Cornell dataset, which contains 240 graspable objects [12]. We relabel the images with the AGA-model.
2) *Clutter dataset:* Because there is no publicly available dataset for cluttered scenarios on RGB images, a Clutter

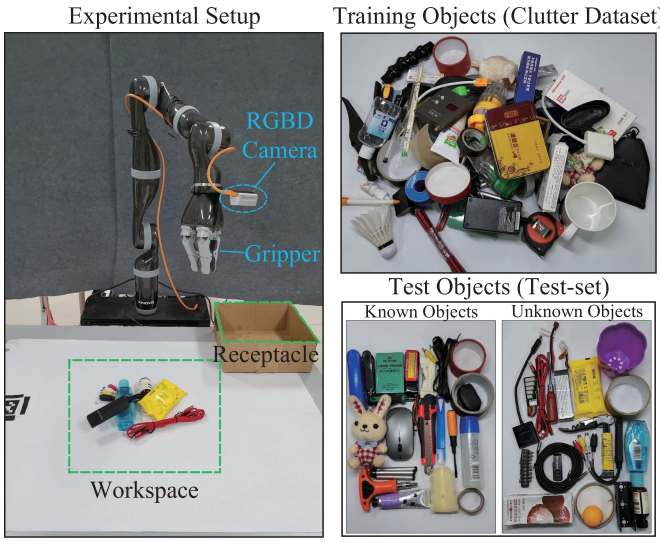


Fig. 5. Left: Setup for robotic grasping experiments. For each experiment, M objects from the test set are randomly dropped on the center of the workspace, at which point the robot iteratively plans grasps from RGB images and attempts to lift and transport the objects to a receptacle. Top-right: A set of 80 training objects in the Clutter dataset with various shapes, sizes, and material properties. Bottom-right: A set of 40 test objects in the test set. Half of the objects appear in the Clutter dataset (known objects), and the other half are novel objects (unknown objects).

dataset is built by our lab and publicly released.[2] There are 505 images in the dataset, which contains 80 graspable objects shown in Fig. 5. Each image contains one to ten stacked objects.

The labeling method is described as follows. The shape of the graspable part of the object is approximately round or rhabdoid. We simplify the grasp region into a standard figure based on the shape of the graspable part.

1) When the shape of the graspable part is round, the grasp region R is approximately circular in the middle of the graspable part. We label it as a circle whose center coincides with the center of the graspable part, and the grasp angle Θ contains all values from 0 to 2π.
2) When the shape of the graspable part is rhabdoid, the grasp region R is approximately rectangular in the middle of the graspable part. We label it as a rotatable rectangle, which is symmetrical along the central axis of the graspable part. The grasp angle Θ has one or two elements according to the size of space on both sides of the graspable part.

The labeling method of the grasp angle Θ and grasp width ω is described in Section III-B.

In the Cornell dataset, we randomly select 75% as the training set and the remaining 25% as the test set. The evaluation methods are divided into the following two principles.

1) *Image-wise splitting* divides the images into the train set and the test set at random. This aims to test the generalization ability of the network on new positions and orientations of a trained object.
2) *Object-wise splitting* divides the dataset at the object instance level. All images of an instance are put into the same set. This aims to test the generalization ability of the network to new objects.

[2][Online]. Available: https://github.com/liuchunsense/Clutter-Grasp-Dataset

B. Training Details

In order to facilitate the training of AFFGA-Net, we normalize the labeled data as follows.

Grasp confidence. We treat grasp confidence Q of each pixel point as a binary label, and set the already labeled grasp points in grasp region a value of 1. All other points are 0.

Grasp angle. Each grasp angle θ in set Θ is labeled in the range $[0, 2\pi)$. We discretize θ to $k = \lfloor \frac{120}{2\pi}\theta \rfloor$, $k \in [0, 119]$.

Grasp width. We scale the values of ω by $\frac{1}{W}$ to put it in the range [0,1]. We set $W = 400$ to prevent the label of the grasp width from exceeding 1 during data enhancement.

We perform data enhancement in multiple ways. We take a center crop of 320×320 pixels with random translation up to 30 pixels horizontally and vertically. This image patch is, then, randomly rotated up to 30° in both clockwise direction and anticlockwise direction. Then, the image is randomly flipped horizontally. After that, we put the image into AFFGA-Net.

AFFGA-Net is implemented with PyTorch. We use Adam optimizer to optimize the networks. The initial learning rate is set to 0.001. The network is trained end-to-end for 500 epochs and the learning-rate decays at a rate of 0.5 times in the range $[100, 200, 300, 400]$ of epochs.

C. Evaluation Metrics for Grasp Detection

The predicted grasping is correct when the following two conditions are met.

1) The difference between the predicted grasp angle θ and the labeled grasp angle is less than 30°.
2) The Jaccard index of the predicted grasp and the label is higher than 25%. The Jaccard index for a predicted grasp G and a labeled grasp G^* is defined as

$$J(G^*, G) = \frac{|G^* \cap G|}{|G^* \cup G|}. \tag{10}$$

To compare with methods based on rectangle representation, we set both d_1 and d_2 of the OAR-model to 30 pixels during the testing phase, which is approximately the mean value of the labeled gripper jaw sizes in the Cornell dataset [12].

D. Test Objects

We build a set of objects on which we test the grasp detection accuracy and robotic grasp success rate (see Fig. 5).

Test set. The set consists of 40 household objects with varying sizes, shapes, and difficulties. Half of the objects appear in the Clutter dataset. The weight of all objects is less than 200 g to ensure that the robot can grasp the objects when the grasp point deviates from the object's center of gravity.

TABLE I
ABLATION EXPERIMENT

Baseline	Mix. up	Sigmoid	HASP	AD	Img (%)	Obj (%)
✓					94.98	95.45
✓	✓				96.80	96.36
✓	✓	✓			98.17	97.27
✓	✓	✓	✓		98.17	98.18
✓	✓	✓	✓	✓	99.09	98.64

Notes: Mix. up: Mixed upsampling. Sigmoid: Sigmoid normalization for angle head. HASP: Hybrid atrous spatial pyramid. AD: Adaptive decoder. Img: Accuracy in image-wise splitting. Obj: Accuracy in object-wise splitting.

E. Physical Setup

To perform robotic grasping experiments, we use a Kinova Gen2 7DOF robot fitted with a Kinova KG-3 gripper. Our camera is an Intel RealSense D435i RGB-D camera and is mounted to the wrist of the robot. This setup is shown in Fig. 5. The AFFGA-Net is performed on a PC running Ubuntu 16.04 with a 3.5 GHz Intel Core i9-9900 CPU and NVIDIA TITAN-XP graphics card.

V. EXPERIMENTS

A. Ablation Experiment

In this section, we perform an ablation study to evaluate the impact of each component of the proposed AFFGA-Net on performance. Table I summarizes the experimental results.

All networks are trained and tested in the Cornell dataset. The baseline gets an accuracy of 94.98% and 95.45% in image- and object-wise splitting, respectively. Mixed upsampling increases accuracy by 1.8% and 0.9%, because deconvolution reduces the loss of detail during the upsampling process. The sigmoid function increases the accuracy by 1.4% and 0.9% due to avoiding competition between each category of the grasp angle. HASP increases the accuracy by 0.9% in object-wise splitting, and AD increases the accuracy by 0.9% and 0.4% by extracting and decoding multiscale features.

B. Grasp Detection in Cornell Dataset

AFFGA-Net is trained and tested in image- and object-wise splitting, respectively. The accuracy is evaluated by the metric described in Section IV-C and the results are shown in Table II. AFFGA-Net achieves the accuracy of 99.09% and 98.64% in image- and object-wise splitting, respectively. Moreover, AFFGA-Net completes a grasp detection pipeline within 15 ms from reading an RGB image to output an OAR-model at every pixel, which can be used for real-time applications.

In Table II, we compare AFFGA-Net with representative planar grasp detection methods in the public Cornell dataset and with the same experimental conditions, i.e., evaluation methods described in Section IV-A and evaluation metrics described in Section IV-C. Based on the Cornell dataset, different input data formats are used in different methods: RGB images are used in [26], [28], [32], [34], and our method, and the point cloud is used in [35] and [33]. The results show that we obtain the maximal accuracy using less scenario information. Methods

Fig. 6. Grasp detection results on the Cornell dataset and actual scenarios. The first row visualizes all the grasp points with grasp confidences over 0.5, and the grasp points in green have higher grasp confidence. The second row visualizes the OAR-models whose grasp confidences are greater than their eight-connected neighbors.

in Table II use a grasping rectangle to represent gripper configuration and use the most advanced network in the field of target detection to detect rectangles. However, the ground truth of their network are a set of discrete rectangular boxes, which is inconsistent with the actual grasping attribute of the object. Instead, our AGA-model is pixel-level to cover the grasping attribute of the object and has fewer variables to make the network easy to train.

Asif *et al.* [28] used a group of upsampling layers to predict grasping rectangles on each pixel. However, pure upsampling layers cannot adapt to objects with different scales. Instead, we use HASP to improve the adaptability of AFFGA-Net to objects with different scales, and use AD to optimize the shape of the grasp region, which improves the accuracy.

To evaluate robustness, our method is compared with the methods of [26], [32], and [33] in Table III, under varying Jaccard index and angle threshold. As the threshold increases, the accuracies of [26], [32], and [33] decrease rapidly, while our method still has a high accuracy rate. Moreover, our method achieves the best results in all experiments. There are three reasons that contribute to this achievement. First, the proposed OAR-model avoids the confusion of the neural network learning; second, the AGA-model makes full use of contextual information; lastly, the proposed AFFGA-Net adaptively generates features for objects with different scales and shapes.

In Fig. 6, we visualize the detection results of some objects. The predicted grasp region covers almost all the graspable positions of the object. The grasp point with maximal confidence tends to appear in the middle of the graspable part, which makes the grasping stable. For objects with small space, such as scissors' handles and tape, the predicted grasp angle is toward the side with smaller space, or the OAR-model spans the entire ring; hence, the multifingered gripper can grasp the object stably. The

TABLE II
PERFORMANCE OF DIFFERENT ALGORITHMS ON CORNELL GRASPING DATASET

Method	Splitting	Jaccard index					Angle threshold				
		0.20	0.25	0.30	0.35	0.40	30°	25°	20°	15°	10°
Song *et al.* [32]	Img (%)	-	95.6	94.9	91.2	87.6	-	-	-	-	-
Chu *et al.* [33]		-	96.0	94.9	92.1	84.7	-	-	-	-	-
Zhou *et al.* [26]		98.31	97.74	96.61	95.48	-	97.74	97.74	97.18	94.35	86.44
Ours		**99.54**	**99.09**	**99.09**	**99.09**	**98.17**	**99.09**	**99.09**	**97.26**	**95.89**	**93.15**
Song *et al.* [32]	Obj (%)	-	97.1	97.1	96.4	93.4	-	-	-	-	-
Chu *et al.* [33]		-	96.1	92.7	87.6	82.6	-	-	-	-	-
Zhou *et al.* [26]		97.74	96.61	93.78	91.53	-	96.61	96.04	95.48	93.22	85.31
Ours		**98.64**	**98.64**	**97.73**	**96.82**	**95.91**	**98.64**	**98.64**	**98.18**	**97.73**	**94.55**

TABLE III
GRASP DETECTION ACCURACIES ON CORNELL GRASPING DATASET FOR DIFFERENT JACCARD INDEXES AND ANGLE THRESHOLD

Method	Img (%)	Obj (%)	speed (fps)	Input data format
Asif *et al.* [34]	88.2	87.5	-	RGB
Kumra *et al.* [35]	89.21	88.96	16.03	Point cloud
Song *et al.* [32]	95.6	97.1	-	RGB
Chu *et al.* [33]	96.0	96.1	8.33	Point cloud
Asif *et al.* [28]	97.5	-	9	RGB
Zhou *et al.* [26]	97.74	96.61	8.51	RGB
Ours	**99.09**	**98.64**	**66.7**	RGB

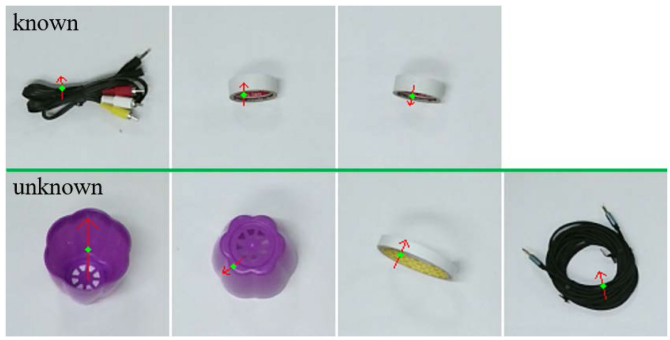


Fig. 7. Unsuccessful detection results of known objects and unknown objects in the actual scenarios. Known: objects in the Clutter dataset. Unknown: Novel objects.

frequent problem in other methods that the grasp point with the maximal confidence tends to appear in the center of all labeled grasp [37] does not appear in our method.

C. Grasp Detection in Actual Scenarios

To evaluate the grasp detection performance in real scenarios, we design a grasp detection experiment, and the experimental conditions are set to be the same as in [7]. We sample one object from the test set dataset in order and put the object on the center of the workspace in random. Specifically, a human operator samples a random pose of the object by shaking the object in a box and placing it upside down in the workspace. On each timestep, the AFFGA-Net receives an RGB image taken by a single overhead depth camera as input and outputs the OAR-model with the highest confidence. The accuracy is evaluated by the metric described in Section IV-C. We test ten random orientations of each object.

Our method achieves a grasp detection accuracy of 98.50% with known objects and 98.0% with unknown objects. The accuracy with unknown objects is shown in Table IV. Fig. 6 shows the detection results of some objects in the actual scenarios. The results show that our method is efficient for objects that have not been trained, even though the category of the object is unknown. We have seven unsuccessful detections in total, which are shown in Fig. 7. The white tape is recognized as the background due to its similarity, and the pattern inside objects also affect the performance.

Fig. 8. Experiments for real-world robotic grasping.

D. Robotic Grasping

To study performance on a physical robot, we design a robotic grasping experiment with reference to [7] and [10]. The setup is shown in Fig. 5. First, we sample M objects from the test set at random. Then, each of the M objects is randomly placed on the center of the workspace. On each timestep, with our grasping policy, we input an RGB image, point cloud, and camera intrinsics, and output multiple OAR-model candidates with grasp confidences Q over 0.5 (see Fig. 8). We select the

TABLE IV
PERFORMANCE OF DIFFERENT ALGORITHMS ON GRASP DETECTION AND ROBOTIC GRASPING WITH UNKNOWN OBJECTS

Method	Detection accuracy (%)	Grasp success rate (%) Single	Clutter	Input data format	Gripper
Mahler *et al.* [7]	-	80	-	Point cloud	Parallel-jaw
Douglas *et al.* [36]	-	92	87	Point cloud	Parallel-jaw
Asif *et al.* [28]	97.2	-	90	RGB	Parallel-jaw
Mahler *et al.* [10]	-	-	91.5	Point cloud	Parallel-jaw
Pas *et al.* [3]	-	-	93	Point cloud	Parallel-jaw
Wu *et al.* [18]	-	95.5	93.1	Point cloud	Parallel-jaw & Three-fingered
Ours	**98.0**	**98.77**	**93.69**	RGB	Parallel-jaw & Three-fingered

Notes: Single: The grasp success rate in the scenarios of one object. Clutter: The grasp success rate is equal to the average of the success rate in the scenarios of five objects and ten objects.

TABLE V
ROBOTIC GRASP SUCCESS RATE OF OUR METHOD (%)

Gripper	Parallel-jaw			Three-fingered		
Objects (M)	1	5	10	1	5	10
Known	99.50	95.69	93.46	99.01	94.79	93.02
Unknown	98.52	95.24	93.46	99.01	93.46	92.60

Notes: Known: Objects in the Clutter dataset. Unknown: Novel objects.

optimal OAR-model Gr^* by

$$Gr^* = \arg\max_{\text{Gr}} Q$$
$$s.t. \begin{cases} D_{\text{xy}} - D_1 \geq D_{\text{th}} \\ D_{\text{xy}} - D_2 \geq D_{\text{th}} \\ Q > 0.5 \end{cases} \tag{11}$$

where D_{xy} refers to the height of the grasp point relative to the platform, and D_1 and D_2 refer to the height of the center of d_1 and d_2 in the OAR-model relative to the platform, respectively. D_{th} is set to 0.005 m in our experiments. Based on the optimal OAR-model Gr^*, the robot, then, approaches the target and closes the jaws. Grasp success is defined by whether or not the grasp transports the target object to the receptacle.

The three-fingered gripper and a parallel-jaw gripper (freeze one finger of the three-fingered gripper) are tested in our experiments. If the robot has five consecutive failed grasps, the objects on the workspace are randomly placed again. Each object is successfully grasped ten times in each set of experiments. If multiple objects are grasped at the same time, one of the objects is randomly selected and put into the receptacle, and the other objects are randomly placed on the workspace again. AFFGA-Net is trained on the Cornell and Clutter datasets.

Table V shows the performance with $M = \{1, 5, 10\}$ test objects. In order to successfully grasp each object ten times in each set of experiments, we conduct a total of 2511 grasping trials. The total number of failed grasps is 111. There are the following three main types of failed grasping.

1) The gripper is blocked by other objects when approaching the object (see Fig. 9).
2) The object is touched by the gripper and deviates from the original position.
3) The object falls while being lifted.

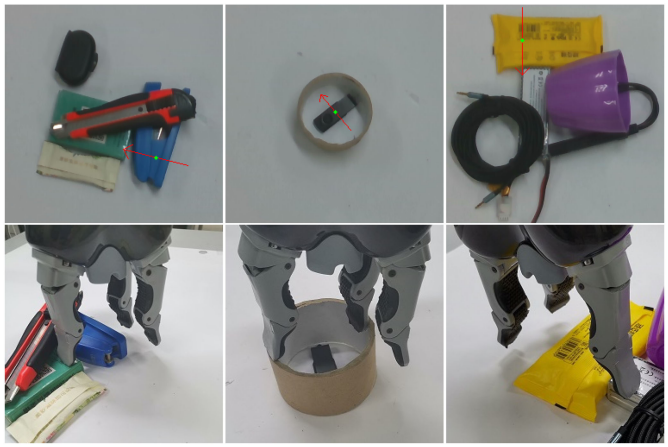


Fig. 9. Three examples of failed grasps. The most common failure modes is that the gripper is blocked by other objects.

The three-fingered gripper performs better in single-object scenarios, and the parallel-jaw gripper performs better in cluttered scenarios. In single-object scenarios, the larger grasp range of the three-fingered gripper makes the grasp more stable. In cluttered scenarios, the jaw size of the three-fingered gripper is larger, and it is more likely to be blocked by other objects, which causes grasp failure. The experimental procedure is shown in the supplemental video.[3]

In Table IV, we compare the performance of different algorithms on robotic grasping with unknown objects. The physical setup used by these researchers are not exactly the same, such as the robot and test objects. Our method achieves grasp success rate of 98.77% in single-object scenarios and 93.69% in cluttered scenarios. Compared with other methods based on RGB images [28], our method has a higher grasp success rate in cluttered scenarios. Compared with methods based on point cloud [3], [18], [36], our method avoids collecting large amounts of point clouds to train the network, and avoids learning the gap between simulation and reality [7], [10].

[3][Online]. Available: https://youtu.be/ccA1jkkbBJA

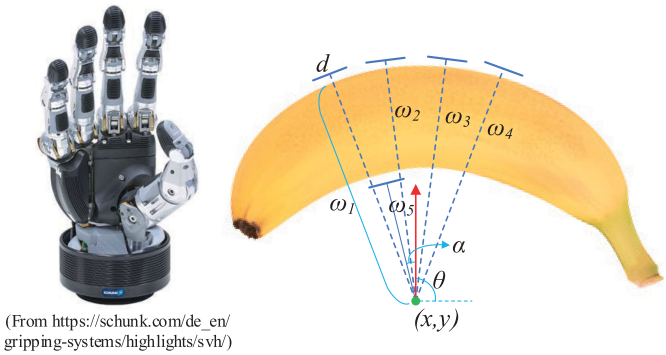


Fig. 10. (a) Multifinger fully actuated anthropomorphic hand. (b) Possible improvement of the OAR-model. It is a possible way to introduce five ω_i to present multifinger instead of one parameter ω in formula (1), and also introduce a new parameter α to represent the angle of the closed direction of the thumb relative to the axis of the palm. Similar to the original OAR-model, θ is defined to represent the palm orientation relative to the horizontal axis of the image, and d is the size of each finger. Based on this improved OAR-model for multifinger grasping, the original AGA-model can be extended to multifinger with more parameters, and the original AFFGA-Net can be modified by designing new grasping heads in Fig. 4, making the whole structure adapt to multifinger grasping.

VI. Conclusion

In this article, we proposed a fast pixel-level grasp detection method. First, a novel OAR-model was introduced to represent the full gripper configuration, which was universal to parallel-jaw and three-fingered grippers. In addition, we proposed a novel AGA-model to model the grasping attribute of the object, which resolved angle conflicts in training and avoided the extremely complicated pixel-level labeling process. At last, a new network called AFFGA-Net was proposed to predict the pixel-level OAR-models on RGB images. The pixel-level mapping avoided missing ground-truth grasp postures, and overcame limitations of current deep-learning grasping techniques by avoiding discrete sampling of grasp candidates and long computation times. Experimental results showed that the proposed method achieved the state-of-the-art grasp detection accuracies on the Cornell Grasping dataset and performed well for unknown objects in multiobject stacked scenarios.

Our method has the following two limitations: 1) the method is only applicative for RGB images and cannot handle point cloud and 2) two adjacent fingers on the three-fingered gripper are constrained by the OAR-model to move synchronously. By separately modeling the closing direction of each finger, it is possible to extend the method in this article to anthropomorphic hands with more flexibility. Specifically, the modified OAR-model models the closing directions and positions of five fingers of the anthropomorphic hand, as shown in Fig. 10.

References

[1] Y. Hu, X. Wu, P. Geng, and Z. Li, "Evolution strategies learning with variable impedance control for grasping under uncertainty," *IEEE Trans. Ind. Electron.*, vol. 66, no. 10, pp. 7788–7799, Oct. 2019.

[2] G. Li, N. Li, F. Chang, and C. Liu, "Adaptive graph convolutional network with adversarial learning for skeleton-based action prediction," *IEEE Trans. Cogn. Develop. Syst.*, early access, Aug. 11, 2021, doi: 10.1109/TCDS.2021.3103960.

[3] K. S. Andreas ten Pas, M. Gualtieri, and R. Platt Jr., "Grasp pose detection in point clouds," *Int. J. Robot. Res.*, vol. 36, no. 13–14, pp. 1455–1473, 2017.

[4] C. Yang, X. Lan, H. Zhang, and N. Zheng, "Task-oriented grasping in object stacking scenes with CRF-based semantic model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 6427–6434.

[5] A. H. Memar and E. T. Esfahani, "A robot gripper with variable stiffness actuation for enhancing collision safety," *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 6607–6616, Aug. 2020.

[6] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: Learning using a new rectangle representation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 3304–3311.

[7] 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.*, 2017. [Online]. Available: http://www.roboticsproceedings.org/rss13/p58.html

[8] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.*, vol. 34, no. 4–5, pp. 705–724, 2015.

[9] 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. Automat.*, 2017, pp. 1609–1614.

[10] M. Jeffrey and G. Ken, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Proc. Conf. Robot Learn.*, 2017, pp. 515–524.

[11] J. Mahler *et al.*, "Learning ambidextrous robot grasping policies," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaau4984, doi: 10.1126/scirobotics.aau4984.

[12] R. L. Lab, "Cornell Grasping dataset," 2009. Accessed: Sep. 1, 2017. [Online]. Available: http://pr.cs.cornell.edu/grasping/rect_data/data.php

[13] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—A survey," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 289–309, Apr. 2014.

[14] Z. He, C. Wu, S. Zhang, and X. Zhao, "Moment-based 2.5-D visual servoing for textureless planar part grasping," *IEEE Trans. Ind. Electron.*, vol. 66, no. 10, pp. 7821–7830, Oct. 2019.

[15] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 348–353.

[16] C. Rubert, D. Kappler, A. Morales, S. Schaal, and J. Bohg, "On the relevance of grasp metrics for predicting grasp success," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 265–272.

[17] Y. Inagaki, R. Araki, T. Yamashita, and H. Fujiyoshi, "Detecting layered structures of partially occluded objects for bin picking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 5786–5791.

[18] B. Wu *et al.*, "Generative attention learning: A 'general' framework for high-performance multi-fingered grasping in Clutter," *Auton. Robots*, vol. 44, no. 6, pp. 971–990, 2020.

[19] S. R. Lakani, A. J. Rodríguez-Sánchez, and J. H. Piater, "Towards affordance detection for robot manipulation using affordance for parts and parts for affordance," *Auton. Robots*, vol. 43, no. 5, pp. 1155–1172, 2019.

[20] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2003, pp. 1824–1829.

[21] J. Mahler *et al.*, "GP-GPIS-OPT: Grasp planning with shape uncertainty using Gaussian process implicit surfaces and sequential convex programming," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 4919–4926.

[22] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 4679–4684.

[23] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *Int. J. Robot. Res.*, vol. 27, no. 2, pp. 157–173, 2008.

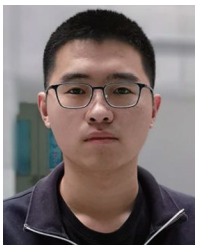
[24] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, "Learning to grasp objects with multiple contact points," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 5062–5069.

[25] D. Guo, F. Sun, B. Fang, C. Yang, and N. Xi, "Robotic grasping using visual and tactile sensing," *Inf. Sci.*, vol. 417, pp. 274–286, 2017.

[26] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng, "Fully convolutional grasp detection network with oriented anchor box," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 7223–7230.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[28] U. Asif, J. Tang, and S. Harrer, "Densely supervised grasp detector (DSGD)," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8085–8093.
[29] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 801–818.
[30] 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.
[31] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
[32] Y. Song, L. Gao, X. Li, and W. Shen, "A novel robotic grasp detection method based on region proposal networks," *Robot. Comput.- Integr. Manuf.*, vol. 65, 2020, Art. no. 101963.
[33] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multi-object, multi-grasp detection," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3355–3362, Oct. 2018.
[34] 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.
[35] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 769–776.
[36] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *Int. J. Robot. Res.*, vol. 39, no. 2–3, pp. 183–201, 2020.
[37] L. Chen, P. Huang, and Z. Meng, "Convolutional multi-grasp detection using grasp path for RGBD images," *Robot. Auton. Syst.*, vol. 113, pp. 94–103, 2019.



Dexin Wang received the B.S. degree in automation in 2019 from Shandong University, Jinan, China, where he is currently working toward the M.S. degree in pattern recognition.

His research interests are computer vision and robotics grasp detection.

Chunsheng Liu (Member, IEEE) received the M.S. and Ph.D. degrees in pattern recognition and machine intelligence from Shandong University, Jinan, China, in 2012 and 2016, respectively.

He was a Postdoctoral and Visiting Researcher with the University of Washington from 2018 to 2019. He is currently an Associate Professor with the School of Control Science and Engineering, Shandong University. His research interests include pattern recognition, machine learning, intelligent transportation system, and bionic intelligence.

Faliang Chang received the B.S. and M.S. degrees in control theory and engineering from Shandong Polytechnic University, Jinan, China, in 1986 and 1989, respectively, and the Ph.D. degree in pattern recognition and intelligence systems, Shandong University, Jinan, in 2003.

Since 2003, he has been a Professor in Pattern Recognition and Machine Intelligence with the School of Control Science and Engineering, Shandong University. His research interests include computer vision, image processing, intelligent transportation system, and multicamera tracking methodology.

Nanjun Li received the B.S. degree in automation from the North University of China, Taiyuan, China, in 2016. He is currently working toward the Ph.D. degree in pattern recognition from Shandong University, Jinan, China.

His research interests include computer vision and abnormal event detection.

Guangxin Li received the B.S. degree in automation in 2018 from Shandong University, Jinan, China, where he is currently working toward the M.S. degree in pattern recognition and intelligence systems with the School of Control Science and Engineering.

His research interests include computer vision and action analysis.