High-performance Pixel-level Grasp Detection based on Adaptive Grasping and Grasp-aware Network

Dexin Wang, Chunsheng Liu, *Member, IEEE*, Faliang Chang, and Nanjun 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 paper, a pixel-level grasp detection method is proposed, which uses deep neural network to predict pixel-level gripper configurations on RGB images. Firstly, 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 (AGA-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 2,400 robotic grasp trials, our structure achieves average success rate of 99.75% in single object scenarios and 95.71% in cluttered scenarios. Moreover, AFFGA-Net completes a grasp detection pipeline within 15ms.

Index Terms—grasp detection, oriented arrow representation, adaptive grasping attribute model, convolutional neural network

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 the 3D location, the 3D orientation, the gripper grasp width, etc [5]. Predicting all the variables is complicated, so the task is simplified to predicting a ‘representation’ of the gripper configuration, which is a

Dexin Wang, Chunsheng Liu, Faliang Chang, and Nanjun Li are with the School of Control Science and Engineering, Shandong University, Ji’nan, Shandong 250061, China (e-mail: fchang, liuchunsheng@sdu.edu.cn).

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

projection of the gripper configuration on the plane. Jiang *et al.* [6] use an oriented rectangle to represent the gripper configuration of the parallel-jaw gripper and Mahler *et al.* [7] use a point and an angle. Jiang *et al.* set the size of the gripper jaw as a variable, but the size of the gripper is not related to the object, which makes the neural network hard to learn. Mahler *et al.* set the grasp width as a constant which limits the size of objects that can be grasped. Moreover, they are only applicable for the parallel-jaw gripper.

What is the output format of the learning algorithm?

We argue that the output format of the algorithm is the grasping attribute of the object, i.e. the sum of all the grasp configurations that an object can be grasped by a gripper. Most methods based on the rectangle representation predict multiple discrete oriented rectangle from RGB images [8] [9]. Besides, Mahler *et al.* first sample some gripper configurations, and then evaluate their confidence [7] [10] [11]. However, the feasible gripper configurations on the object are continuous, 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, to improve the learnability and applicability of the grasp representation, we design the OAR-model. By simplifying the three-fingered gripper (Fig. 2(a)) into a parallel-

jaw gripper with two jaws of different sizes, OAR-model is applicable for the parallel-jaw gripper and the simplified three-fingered gripper. OAR-model avoids the confusion of the neural network learning by constraining the size of the gripper jaw, and is applicable for objects with different sizes by using variable grasp width.

Second, to optimize the learning process of network, we propose the AGA-model which represent the grasping attribute of objects. One pixel may correspond to multiple OAR-models with different grasp angle, 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 by avoiding discrete sampling of grasp candidates and long computation times. Moreover, AFFGA-Net is robust for objects of any shape and size by using *Hybrid Atrous Spatial Pyramid* (HASP) and *Adaptive Decoder* (AD) which adaptively extract and decode multi-scale features.

On the Cornell Grasp 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%. On the randomly selected household objects, our method achieves 98.0% grasp detection accuracy. In over 2,400 robotic grasp trials, our structure achieves an average success rate of 99.75% on single object scenarios and 95.71% on cluttered scenarios. Moreover, AFFGA-Net completes a grasp detection pipeline within 15ms, which can be used for real-time applications. Our method is proved effective for novel objects in multi-object stacked scenarios.

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 neural network learning and applicable for objects with different sizes.
- 2) The AGA-model is proposed to resolve angle conflicts in training and avoid the extremely complicated pixel-level labeling process.
- 3) The AFFGA-Net is proposed to generate pixel-level OAR-model and confidence 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 Grasp dataset and is proved effective for novel objects in cluttered scenarios.

This paper 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, and then Section VI presents the conclusion.

II. RELATED WORK

The goal of grasp detection is to find a proper posture using the visual information of the scenarios so that the gripper can stably grasp the target when closing the jaws in this posture. The proposed methods can be roughly divided into two categories: analytic methods and empirical methods [13]. Analytic methods use mathematical and physical models of geometry, kinematics and dynamics to calculate grasping that are stable [14] [15], but tend to not transfer well to the real world due to the difficulty in modelling physical interactions between a manipulator and an object [16]. In contrast, empirical methods do not require the 3D model of the object, they train the grasp model from the known object and use this model to detect the grasping posture of the unknown objects [17] [18] [19]. Now, some deep learning based methods have 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] use supervised learning to predict a grasp point from the image and successfully extended it to new targets. Le *et al.* [24] suggest that a pair of points are used to represent the grasping. Jiang *et al.* [6] reduce the dimensionality of the 7-dimensional gripper configuration (the 3D location, the 3D orientation and the distance between the two fingers) in the real environment to obtain a simplified five-dimensional 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] use ResNet-50 as the feature extractor and adopt anchor mechanism [27] to predict the five-dimensional rectangle grasp model, which greatly improves the prediction accuracy. Asif *et al.* [28] overcome limitations of the individual models by combining CNN structures with layer-wise feature fusion and producing grasping and their 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 impossible 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. Firstly, the *Oriented Arrow Representation model* (OAR-model) is introduced to represent the mapping of the gripper configuration on the plane. Secondly, all OAR-models on the object are merged into multiple *Adaptive Grasping Attribute models* (AGA-models), and the AGA-models are labeled as target to train the *Adaptive Feature Fusion and Grasp Aware Network* (AFFGA-Net). AFFGA-Net inputs a RGB image and outputs an OAR-model at every pixel. Then, the gripper configuration is calculated

Fig. 2. (a): The three-fingered gripper. (b): Examples of OAR-models.

Fig. 3. (a): Illustration of AGA-models. 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.

using the optimal OAR-model and point cloud. Lastly, the robot approaches the target and close the jaws.

A. Oriented Arrow Representation Model

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

$$G_r = \{x, y, \omega, d_1, d_2, \theta\} \quad (1)$$

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 an OAR-model and the point cloud of object, the 3D coordinate of the grasp point is the projection point of (x, y) in the point cloud. For three-fingered gripper, the d_1 position is

where the single finger places, and the d_2 position is where the other two fingers place. For 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 towards the grasp point to grasp object.

Rectangle representation sets the size of the gripper jaw as a variable [6], but the size of the gripper is not related to the object, which makes the neural network hard to learn. 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 the neural network learning. Compared with Dex-Net 2.0 [7], the grasp width of OAR-model is variable and can be learned to fit objects with different sizes.

B. Adaptive Grasping Attribute Model

There are two difficulties with learning pixel-level mapping: (1) One pixel may correspond to multiple OAR-models with different grasp angle, which may cause angle conflicts. (2) It is extremely time-consuming to label pixel-level OAR-models. In order to solve the above problems, we propose AGA-model to model the grasping attribute of the object, which is composed of multiple adjacent OAR-models. AGA-model transforms single-angle learning tasks into multi-angle 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:

$$G_m = \{R, \Theta, \omega\} \quad (2)$$

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 often 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 θ and grasp width ω . The shape of the grasp region is similar to that of the graspable part of 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 object) of the grasp region to L (1cm 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 (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 parallel-jaw gripper, the three-fingered gripper need additionally consider the different space around the objects, such as tape with a small inner ring, scissors handle, etc. 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:

Fig. 4. Structure of the AFFGA-Net. (a): Shared Encoder. (b): Hybrid Atrous Spatial Pyramid. (c): Adaptive Decoder. (d): Grasping Heads.

- 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:

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

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:

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

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 :

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

We set $s_{th} = d_2$ 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 [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 . 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 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 traditional 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 four solutions to build our AFFGA-Net (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 follow.

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 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 three sub-problems: grasp region segmentation, grasp angle classification and grasp width regression.

2) *Hybrid Atrous Spatial Pyramid*: Chen *et al.* [31] verify that Atrous Spatial Pyramid Pooling (ASPP) is effective for improving the segmentation accuracy of multi-scale objects. Nonetheless, the large hole rate ($rate = \{6, 12, 18\}$) leads to low information utilization and loss of local features. We design Hybrid Atrous Spatial Pyramid (HASP) to solve the above problems.

HASP includes two parallel two-layer feature pyramids, which are used to extract features of 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 multi-scale 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 Adaptive Decoder (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, then concatenated with large-scale features. In the down-decoder, the large-scale features are first concatenated with the small-scale features, then concatenated with the low-level features (Fig. 4(c)). The difference in feature fusion order makes the output of the upper decoder not contain the edge information of large-scale objects, while the output of the lower decoder contains all the information of objects of different scales. The features output by the upper decoder are input into the region head to predict the grasp region. The features output by the lower decoder are used to predict the grasp angle and grasp width, because the grasp angle and grasp width on objects of 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 mask of the object, we use a 3×3 deconvolution followed by a bilinear upsampling to accurately restore the grasp region.

5) *Sigmoid*: Predicting grasp angle is a multi-label single-classification task. We use the sigmoid function to normalize the output of angle head to avoid competition between each category.

D. Loss Function

We calculate the loss separately for the output of each head and use the sum of the 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:

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

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

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

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

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, y_l^n is the corresponding label. We show in Sec.V-A that using sigmoid instead of softmax increases the accuracy.

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

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

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

4) *multi-task loss*: In order to balance the loss of each branch, the final multi-task loss is defined as:

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

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

IV. EXPERIMENTS SET-UP

The performance of the proposed method is evaluated on the Cornell Grasp dataset [12] and a test set captured in actual scenarios. The grasp detection accuracy and robotic grasp success rate is 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 two datasets:

- 1) **Cornell dataset**: There are 878 images in the Cornell dataset which contains 240 graspable objects [12]. We relabel the images with AGA-model.
- 2) **Clutter dataset**: Because there is no public available dataset for cluttered scenarios on RGB images, a Clutter dataset is built by our lab and publicly released¹. There are 505 images in the dataset which contains 80 graspable objects shown in Fig. 5. Each image contains 1 to 10 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 labelling method of the grasp angle Θ and grasp width ω is described in Section III-B.

In Cornell ataset, 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:

¹<https://github.com/liuchunsense/Clutter-Grasp-Dataset>

- 1) **Image-wise splitting** divides the images into train set and test set at random. This aims to test the generalization ability of the network to a new position and orientation of an object it has seen before.
- 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 a new object.

B. Training Details

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

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

Grasp Angle. Each grasp angle θ in set Θ is labelled 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 centre crop of 320×320 pixels with random translation up to 30 pixels in both horizontally and vertically. This image patch is then randomly rotated up to 30 degrees in both clockwise and anti-clockwise 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 learning-rate decays stepwise at a rate of 0.5 times in the range $[100, 200, 300, 400]$ of epochs.

C. Grasp Detection Evaluation Metric

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

- 1) The difference between the predicted grasp angle θ and the labelled 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 labelled grasp G^* is defined as:

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

In order to compare with previous methods based on rect-angle representation, we set both d_1 and d_2 of 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 of our approach (Fig. 5).

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

Fig. 5. Left: Set-up 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 Clutter dataset with various shapes, sizes, and material properties. Bottom-Right: A set of 40 test objects in Test-set. Half of the objects appear in Clutter dataset (known objects), and the other half are novel objects (unknown objects).

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

* **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 Set-up

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 set-up is shown in Fig. 5. The AFFGA-Net is performed on a PC running Ubuntu 16.04 with a 3.5GHz Intel Core i9-9900 CPU and NVIDIA TITAN-XP graphics card.

V. EXPERIMENTS

A. Ablation Experiment

In this subsection, 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 Cornell dataset. The baseline gets accuracy of 95.0% and 95.5% in image-wise and object-wise splitting, respectively. Mixed Upsampling increases accuracy by 1.8% and 0.9% respectively, because

TABLE II
GRASP DETECTION ACCURACIES ON CORNELL GRASP DATASET FOR DIFFERENT JACCARD INDEXES AND ANGLE THRESHOLD.

Method	Splitting	Jaccard index					Angle threshold				
		0.20	0.25	0.30	0.35	0.40	30°	25°	20°	15°	10°
Song <i>et al.</i> [32]	Img (%)	-	95.6	94.9	91.2	87.6	-	-	-	-	-
Chu <i>et al.</i> [33]		-	96.0	94.9	92.1	84.7	-	-	-	-	-
Zhou <i>et al.</i> [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 <i>et al.</i> [32]	Obj (%)	-	97.1	97.1	96.4	93.4	-	-	-	-	-
Chu <i>et al.</i> [33]		-	96.1	92.7	87.6	82.6	-	-	-	-	-
Zhou <i>et al.</i> [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
PERFORMANCE OF DIFFERENT ALGORITHMS ON CORNELL GRASP DATASET.

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

deconvolution reduces the loss of detail during the upsampling process. Sigmoid function increases accuracy by 1.4% and 0.9% respectively due to avoiding competition between each category of grasp angle. HASP increases accuracy by 0.9% in object-wise splitting, and AD increases accuracy by 0.9% and 0.4% respectively by extracting and decoding multi-scale features.

B. Grasp Detection in Cornell Dataset

AFFGA-Net is trained and tested in image-wise and object-wise splitting respectively. The accuracy is evaluated by the metric in Sec. IV-C and the results are shown in Table.III. AFFGA-Net achieves accuracy of 99.1% and 98.6% in image-wise and object-wise splitting respectively. Moreover, AFFGA-Net completes a grasp detection pipeline within 15ms from reading a RGB image to output an OAR-model at every pixel, which can be used for real-time applications.

In Table.III, we compare AFFGA-Net with representative planar grasp detection algorithms that based on grasping rectangle. The results show that, we obtain the maximal accuracy using less scenarios information (i.e. RGB). Methods in Table.III 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 prediction targets 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] use a group of upsampling layers to predict grasping rectangles on each pixel. However, pure upsampling layers cannot adapt to objects of different scales. Instead, we use HASP to improve the adaptability of AFFGA-Net to objects of different scales, and use AD to optimize the shape of the grasp region, which improve the grasp detection accuracy.

To evaluate the robustness, our method is compared with the methods of [32] [33] and [26] in Table. II, under varying Jaccard index and angle threshold. As the threshold increases, the accuracies of [32] [33] [26] 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. Firstly, the proposed OAR-model avoids the confusion of the neural network learning; Secondly, the AGA-model makes full use of contextual information; Lastly, the proposed AFFGA-Net adaptively generates features for objects of 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. 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 multi-fingered gripper can grasp the object stably. The frequent problem in other methods, that the grasp point with the maximal confidence tends to appear in the centre of all labelled grasp [36], 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. We sample one object from the Test-set dataset in order and put the object on the center of the workspace in random. On each timestep the AFFGA-Net receives a RGB image as input and outputs the OAR-model with the highest confidence. The accuracy is evaluated by the metric in Sec. IV-C. We test 10 random orientations of each object. AFFGA-Net is trained on Cornell dataset and Clutter dataset.

Our method achieves grasp detection accuracy of 98.50% with known objects and 98.0% with unknown objects. The grasp detection accuracy with unknown objects is shown in

Fig. 6. Grasp detection results on 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 the local maxima.

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

Table.V. Fig. 6 visualizes 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 7 unsuccessful detections in total, which is shown in Fig. 7. The white tape is recognized as the background due to their similarity, and the pattern inside objects also affects the performance.

D. Robotic Grasping

To study performance on a physical robot, we design a robotic grasping experiment benchmark as illustrated 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 the grasping policy inputs a RGB image, point cloud, and camera intrinsics, and outputs multiple OAR-model candidates with grasp confidences Q over 0.5

TABLE IV
ROBOTIC GRASP SUCCESS RATE OF OUR METHOD (%).

Gripper Objects (M)	Parallel-jaw			Three-fingered		
	1	5	10	1	5	10
Known	99.50	95.69	93.46	99.01	94.79	93.02
Unknown	99.50	97.09	95.24	100	96.62	93.90

* **Known**: Objects in Clutter dataset. **Unknown**: Novel objects.

(Fig. 8). We select the optimal OAR-model Gr^* by:

$$Gr^* = \arg \max_{Gr} Q$$

$$s.t. \begin{cases} D_{xy} - D_1 \geq D_{th} \\ D_{xy} - D_2 \geq D_{th} \\ Q > 0.5 \end{cases} \quad (11)$$

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 meter 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 transport the target object to the receptacle.

The three-fingered gripper and a parallel-jaw (freeze one finger of the three-fingered gripper) are tested in our experiments. If the robot has 5 consecutive failed grasps, the objects on the workspace are randomly placed again. Each object is successfully grasped 10 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 Cornell and Clutter dataset.

Table.IV shows the performance with $M = \{1, 5, 10\}$ test objects. In order to successfully grasp each object 10 times in each set of experiments, we conduct a total of 2,489 grasping trials. The total number of failed grasps is 89. There are three main types of failed grasping: (1) The gripper is blocked by other objects when approaching the object (Fig. 9); (2) The object is touched by the gripper and deviates from the original position; (3) The object falls while being lifted. The experimental procedure is shown in the supplemental video².

In Table.V, we compare the performance of different algorithms on robotic grasping with unknown objects. The physical set-up used by these researchers are not exactly the same, such as the robot and test objects. Our method achieves grasp success rate of 99.75% on single object scenarios and 95.71% on cluttered scenarios. Compared with other method based on RGB images [28], our method has a higher grasp success rate in cluttered scenarios. Compared with methods based on point cloud [37] [3] [18], our method avoids collecting large amounts of point clouds to train the network, and avoids learning the gap between simulation and reality [7] [10].

VI. CONCLUSION

In this paper, we propose a fast pixel-level grasp detection method. Firstly, a novel *Oriented Arrow Representation model*

²<https://www.youtube.com/watch?v=KUa3XlVwDsU>

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

Method	Detection accuracy (%)	Grasp success rate (%)		Input data format	Gripper
		Single	Clutter		
Mahler <i>et al.</i> [7]	-	80	-	Point cloud	Parallel-jaw
Douglas <i>et al.</i> [37]	-	92	87	Point cloud	Parallel-jaw
Asif <i>et al.</i> [28]	97.2	-	90	RGB	Parallel-jaw
Mahler <i>et al.</i> [10]	-	-	91.5	Point cloud	Parallel-jaw
Pas <i>et al.</i> [3]	-	-	93	Point cloud	Parallel-jaw
Wu <i>et al.</i> [18]	-	95.5	93.1	Point cloud	Parallel-jaw & Three-fingered
Ours	98.0	99.75	95.71	RGB	Parallel-jaw & Three-fingered

* **Single**: The grasp success rate in the scenarios of 1 object. **Clutter**: The grasp success rate is equal to the average of the success rate in the scenarios of 5 objects and 10 objects.

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

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

(OAR-model) is introduced to represent the full gripper configuration, which is universal to parallel-jaw and three-fingered gripper. In addition, we propose a novel *Adaptive Grasping Attribute model* (AGA-model) to model the grasping attribute of the object, which resolves angle conflicts in training and avoids the extremely complicated pixel-level labeling process. At last, a new network called *Adaptive Feature Fusion and Grasp Aware Network* (AFFGA-Net) is proposed to predict

the pixel-level OAR-models on RGB images. The pixel-level mapping avoids missing ground-truth grasp postures, and overcomes limitations of current deep-learning grasping techniques by avoiding discrete sampling of grasp candidates and long computation times. Experimental results show that the proposed method achieves the state-of-the-art grasp detection accuracies on Cornell Grasp Dataset and performs well for unknown objects in multi-object stacked scenarios.

Our method has two limitations: (1) The movement of the finger in the three-fingered gripper is constrained; (2) The method is only applicative for RGB images and cannot handle point cloud. In future, we plan to use a simulation system to automatically generate pixel-level labels from point cloud. All code and data in this study will be publicly released.

ACKNOWLEDGMENT

This work is supported in part by National Key R&D Program of China (NO.2018YFB1305300), Shandong Provincial Key Research and Development Program (Major Scientific and Technological Innovation Project) (NO.2019JZZY010130, 2020CXGC010207), and China Natural Science Foundation Committee(61673244, 61703240).

REFERENCES

- [1] L. Birglen and T. Schlicht, "A statistical review of industrial robotic grippers," *Robotics and Computer-Integrated Manufacturing*, vol. 49, pp. 88–97, 2018.
- [2] 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, 2019.
- [3] K. S. Andreas ten Pas, Marcus Gualtieri and R. P. Jr., "Grasp pose detection in point clouds," *Int. J. Robotics 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 *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6427–6434, 2019.
- [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, 2020.
- [6] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgb-d images: Learning using a new rectangle representation," in *2011 IEEE International conference on robotics and automation*, pp. 3304–3311, 2011.
- [7] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Robotics: Science and Systems*, 2017.

- [8] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, 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 *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1609–1614, 2017.
- [10] M. Jeffrey and G. Ken, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Conference on robot learning*, pp. 515–524, 2017.
- [11] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, 2019.
- [12] R. L. Lab, "Cornell grasping dataset," [EB/OL], 2009, http://pr.cs.cornell.edu/grasping/rect_data/data.php Accessed September 1, 2017.
- [13] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2013.
- [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, 2019.
- [15] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *2000 IEEE International Conference on Robotics and Automation*, pp. 348–353, 2000.
- [16] C. Rubert, D. Kappler, A. Morales, S. Schaal, and J. Bohg, "On the relevance of grasp metrics for predicting grasp success," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 265–272, 2017.
- [17] Y. Inagaki, R. Araki, T. Yamashita, and H. Fujiyoshi, "Detecting layered structures of partially occluded objects for bin picking," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5786–5791, 2019.
- [18] B. Wu, I. Akinola, A. Gupta, F. Xu, J. Varley, D. Watkins-Valls, and P. K. Allen, "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 *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 1824–1829. IEEE, 2003.
- [21] J. Mahler, S. Patil, B. Kehoe, J. Van Den Berg, M. Ciocarlie, P. Abbeel, and K. Goldberg, "Gp-gpis-opt: Grasp planning with shape uncertainty using gaussian process implicit surfaces and sequential convex programming," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4919–4926. IEEE, 2015.
- [30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [22] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4679–4684. IEEE, 2007.
- [23] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, 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 *2010 IEEE International Conference on Robotics and Automation*, pp. 5062–5069, 2010.
- [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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7223–7230, 2018.
- [27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.
- [28] U. Asif, J. Tang, and S. Harter, "Densely supervised grasp detector (dsgd)," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8085–8093, 2019.
- [29] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 801–818, 2018.
- [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 transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
- [32] Y. Song, L. Gao, X. Li, and W. Shen, "A novel robotic grasp detection method based on region proposal networks," *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101963, 2020.
- [33] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multi-object, multi-grasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.
- [34] U. Asif, M. Bennamoun, and F. A. Sohel, "Rgb-d object recognition and grasp detection using hierarchical cascaded forests," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 547–564, 2017.
- [35] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 769–776, 2017.
- [36] L. Chen, P. Huang, and Z. Meng, "Convolutional multi-grasp detection using grasp path for rgbd images," *Robotics and Autonomous Systems*, vol. 113, pp. 94–103, 2019.
- [37] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 183–201, 2020.