Robotica (2023), **41**, pp. 3846–3885
doi:10.1017/S0263574723001285

CAMBRIDGE
UNIVERSITY PRESS

REVIEW ARTICLE

A review of robotic grasp detection technology

Minglun Dong and Jian Zhang

School of Mechanical Engineering, Tongji University, Shanghai, China
Corresponding author: Minglun Dong; Email: 364044341@qq.com

Received: 4 January 2023; **Revised:** 27 August 2023; **Accepted:** 31 August 2023; **First published online:** 22 September 2023

Keywords: robotic hands; grasping; control of robotic systems

Abstract

In order to complete many complex operations and attain more general-purpose utility, robotic grasp is a necessary skill to master. As the most common essential action of robots in factory and daily life environments, robotic autonomous grasping has a wide range of application prospects and has received much attention from researchers in the past decade. However, the accurate grasp of arbitrary objects in unstructured environments is still a research challenge that has not yet been completely overcome. A complete robotic grasp system usually involves three aspects: grasp detection, grasp planning, and control subsystem. As the first step, identifying the location of the object and generating the grasp pose is the premise of successful grasp, which is conducive to planning the subsequent grasp path and the realization of the entire grasp action. Therefore, this paper conducts a literature review focusing on grasp detection technology and concludes two significant aspects: the analytic and data-driven methods. According to the previous grasp experience of the target object, this paper divides the data-driven methods into the grasp of known and unknown objects. Then it describes in detail the typical grasp detection methods and related characteristics of each classification in the grasp of unknown objects. Finally, current research status and potential research directions in this field are discussed to provide some reference for related research.

1. Introduction

In recent years, with the continuous development of robotics and the increasing cost of labor, it has become a development trend to replace human beings with robots [1]. At present, intelligent robots have been widely used in various fields such as industry, agriculture, medical care, and life. Humans have always hoped that intelligent robots can perceive and interact with the environment in different application scenarios, which can make the work efficient, accurate, and safe [2, 3]. Therefore, the research and development of robot operation skills have important practical significance for transforming and upgrading the manufacturing industry and the improvement of the current situation of social labor shortage.

In robot control, the grasping skill of the robot arm is an important part, which is also the basis for the robot to move and transport objects [4]. As the most commonly used primary action of robots, robotic arm autonomous grasp has a wide range of application prospects. Compared with traditional manual operation, it can perceive the external environment in the process of grasping, and there is no need to set the pose of the target before each grasp, which dramatically improves the work efficiency. However, it is still an unsolved challenge to accurately grasp arbitrary objects when the robotic arm is working in unstructured environments or affected by other uncertain factors [5–7].

To solve these problems, many researchers are devoted to improving the interaction perception ability of robots with the external environment, and the emergence of machine vision makes up for the defects in the ability of robots to perceive the external environment to some extent. In recent years, with the successive appearance of Microsoft Kinect, Intel RealSense, and other visual sensing devices [8, 9], as well as the continuous development of relevant visual algorithms, the perception ability of robots in

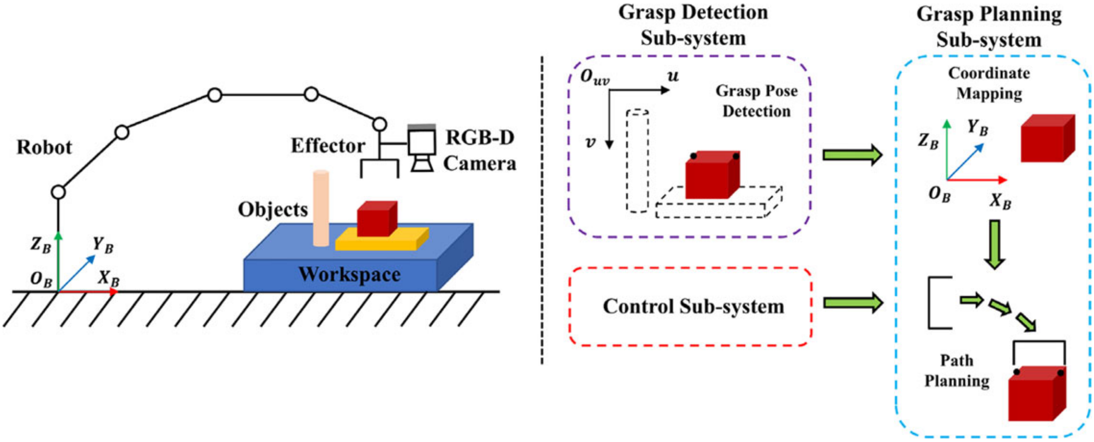

Figure 1. *The robotic grasping system. Left: The robot is equipped with an RGB-D camera and end effector for grasping target objects in the workspace. Right: The whole system mainly includes three parts: the grasp detection subsystem, the grasp planning subsystem, and the control subsystem.*

different scenes has been significantly improved, which makes the robot achieve breakthrough progress in the field of intelligent grasp.

In addition, the proposal of deep learning makes artificial intelligence technology more widely integrated into machine vision. Deep learning relies on the powerful computing ability of computers to autonomously learn relevant information from large datasets, which can enable robots to better adapt to unstructured environments and is widely used for general target detection with ideal results [10]. The emergence of deep learning has promoted the process of robot intelligence. When faced with different task scenarios and target poses, the robot arm can execute grasp operations autonomously, effectively improving the working efficiency of the system. Therefore, using deep learning is the main research direction of robot intelligent grasp, which has far-reaching significance for developing the robot control field. Generally speaking, a complete robotic grasping system mainly includes three parts [11], as shown in Fig. 1:

- **Grasp detection subsystem:** To detect the target object from images and obtain its position and pose information in the image coordinate system.
- **Grasp planning subsystem:** To map the detected image plane coordinates to the robot base coordinate system and generate a feasible path from the manipulator to the target object.
- **Control subsystem:** To determine the inverse kinematics solution of the previous subsystem and control the robot to execute the grasp according to the solution results.

As the starting point of the whole system, the primary purpose of grasp detection is to detect the target objects and generate the grasp poses to achieve a stable and effective grasp. The grasp planning and the control subsystem are more relevant to the motion and automation discipline, which are not the focus of this paper.

At present, there are many review papers about robotic grasping technology. However, most are based on introducing the entire grasping process, and there are few specific discussions on robotic grasp detection. For example, refs. [12–15] mainly introduced robotic grasping based on the mechanics of grasping and the finger–object contact interactions, which focus on the essence of grasping, but are not novel enough. Refs. [8, 10] mainly reviewed the current research progress of generalized robotic grasping from machine vision and learning perspectives. Refs. [16, 17] focused on the review of robotic grasp detection, but the classification of grasp detection is not detailed enough. Therefore, this paper firstly classifies the robotic grasp detection technology in detail. Then, many classical or novel grasp detection

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

techniques and related research progress are introduced. Finally, we analyze the future research direction and development trend of robotic grasp detection technology, which provides a certain reference for the research and practical application in this field.

2. Categorization of methods

Current robot grasp detection methods have various classification methods according to different criteria, which can generally be divided into two major categories [14, 18]. The first category is the traditional analytic method (sometimes called the geometric method) [10], whose basic principle is to determine the appropriate grasp pose by analyzing the geometry, motion state, and force of the target object. The second category is the data-driven method based on machine learning (sometimes called the empirical method) [10], the basic principle of this method is to let the robot imitate the human grasp strategy for grasp detection, which does not need to establish complex mathematical or physical models before grasping, but the calculation is relatively complex. However, with the increase in data availability, computer performance, and the improvement of related algorithms, more and more researchers chose to use the data-driven method. Therefore, this paper will focus on introducing the data-driven method. As for the analytic method, this paper will briefly introduce it.

3. Grasp detection technology based on analytic methods

The analytic methods usually require the kinematic and dynamic modeling of the grasp operation to find stable grasp points that can satisfy the constraints (such as grasp flexibility, balance, and stability). Generally, according to the multi-objective optimization methods, to find a stable grasp point needs to consider all the constraints. However, due to the high dimension of the grasp search space and the nonlinearity of the constraint conditions, only some of the limitations are considered and others are assumed as known or ignored. By reviewing relevant papers, the analytic methods can be divided into form-closure grasp, force-closure grasp, and task-oriented grasp [14], which will be introduced in the following sections.

3.1. Form-closure grasp

Form-closure and force-closure are two major bases for judging the stability of a robot grasp [19]. Form-closure means the robot can completely restrain the object's motion in any direction without any positional change by configuring a suitable grasping position. As for how to judge whether a grasp is a form-closed, Salisbury and Roth [20] have demonstrated that a necessary and sufficient condition for form-closure is that the origin of the wrench space lies inside the convex hull of primitive contact wrenches. Liu [21] further demonstrated that the problem of querying whether the origin lies inside the convex hull is equivalent to a ray-shooting problem, which is dual to an LP problem based on the duality between convex hulls and convex polytopes. Ding et al. [22] studied higher-dimensional form-closure grasp and represented the n-finger form-closure grasp by two sets of inequalities involving the friction cone constraint and the form-closure constraint. The authors [23] simplified the above problems and defined a distance function to represent the distance between the manipulator and the target, as shown in Eq. (1):

$$d = \psi_i \left(u, q\right), 1 \leq i \leq n_c \rightarrow \begin{cases} d > 0, \text{no contact} \\ d = 0, \text{contacted} \\ d < 0, \text{penetrated} \end{cases} \tag{1}$$

where u and q represent the configurations of the target and manipulator for a given grasp, respectively, n_c indicates the number of contacts between the manipulator and the object. Based on the definition of

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

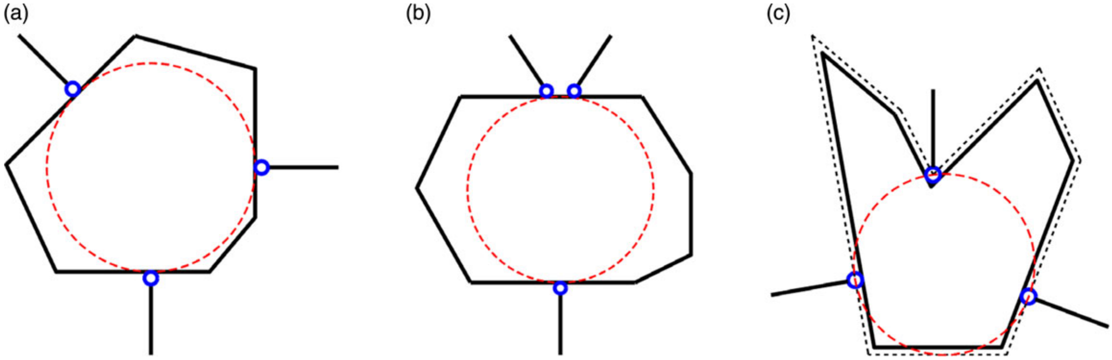

Figure 2. *The three-finger stable grasping strategy for convex polygons and nonconvex polygons proposed in ref. [24]. (a) For convex polygons, the maximal inscribed circle touches the polygon at three points. (b) For convex polygons, the maximal inscribed circle touches at two parallel edges. (c) For nonconvex polygons, the inscribed circle intersects a concave vertex or a linear edge of the expanded polygon.*

the distance function, they further proposed a judgment formula about the form-closure grasp. When the object's position produces a differential change Δu, it is necessary to satisfy that there is no penetration between the manipulator and the object ($d \geq 0$). Solving this inequality, if there exists only a solution for $\Delta u = 0$, it means that the current grasp action is a form-closure grasp. The whole process can be represented by Eq. (2):

$$\psi_i\left(u + \Delta u, q\right) \geq 0 \overset{\text{sove the inequality}}{\rightarrow} \Delta u = \begin{cases} 0 \\ \text{others} \end{cases} \tag{2}$$

Early studies on form-closure mainly focused on objects with simple geometry. Baker et al. [24] presented a method that achieves a stable grasp for 2D polygonal objects with a hand consisting of three spring-loaded fingers and with five degrees of freedom (DOFs). In this method, they proposed corresponding grasping strategies for convex polygons and nonconvex polygons, as shown in Fig. 2. On this basis, Markenscoff et al. [25] proved that any polygon object (except the circle) can always be form-closed with four frictionless contacts. They also indicated that a spatial object can be form-closed with only seven frictionless contacts in three dimensions.

As for complex geometry, other researchers also gave the calculation methods of form-closure grasp. Nguyen [26] proposed a simple test algorithm for two-finger form-closure grasps. Ponce and Faverjon [27] developed several sufficient conditions for three-finger form-closure grasps and computed all grasps satisfying those sufficient conditions. Cornellà and Suarez [28] performed 2D fixture planning of non-polygonal workpieces based on the form-closure and proposed a method for computing the independent form-closure region. They used the object presented in ref. [29] to validate the proposed method (as shown in Fig. 3), and four frictionless contacts were selected from the object boundary within an independent region to realize the form-closure grasp and improve the robustness of the grasp.

3.2. Force-closure grasp

Force-closure means that the appropriate contact force counteracts the external force on the object at the grasp points to constrain the object's movement completely. In past studies, there is a wide disparity in the descriptions of terms such as equilibrium, stability, form-closure, and force-closure in related literature [26, 30–32]. We adopt the terminology in [31] and summarize an equation to describe the force-closure. A grasped object with an external wrench is in equilibrium if and only if:

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

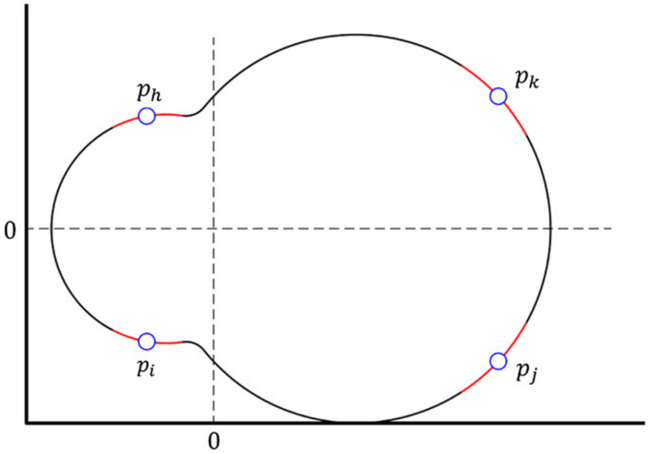

Figure 3. *Independent regions (red edges) and frictionless contacts (blue points) on the object boundary.*

$$\begin{cases} \forall i, c_n^i > 0, \left|c_t^i\right| < \mu_t^i c_n^i, \left|c_s^i\right| < \mu_s^i c_n^i \\ W\lambda + \hat{W} = 0, \lambda \neq 0 \end{cases} \tag{3}$$

In Eq. (3), c_t^i, c_n^i, c_s^i, respectively, represent the tangential force, normal force, and torque at the ith contact. μ_t^i and μ_s^i, respectively, represent the tangential and torsional friction coefficient. W indicates the spiral consisting of force and moment, λ is a coefficient, and $\hat{W}$ represents the external spiral.

There is a specific relation between force-closure and form-closure grasp. Form-closure property is usually a stronger condition than force-closure, and the analysis of form-closure is essentially geometric [14]. More precisely, a grasp achieves form-closure if and only if it achieves force-closure with frictionless point contacts. In this case, form-closure and force-closure are dual to each other [26, 33]. Hence, like form-closure, most of the early studies about force-closure focused on 2D objects due to the geometric simplicity and low calculation cost. Related works can be found in [26, 27, 34]. As for 3D objects, there are two main research aspects of force-closure grasp: (1) simplifying the contact model between the manipulator and the target; (2) finding optimal fingertips locations such that the grasp is force-closure.

Understanding the nature of contact is paramount to the analysis of grasping. Ciocarlie et al. [35] discussed some possible contact models, such as point contact with friction and soft finger contact. They also extended a simulation and analysis system with finite element modeling to evaluate these complex contact types. Bicchi et al. [15] analyzed the interrelationship between the contact model and the grasp contact forces in static grasping and found that not all contact internal forces need to be controlled, which means the DOFs of the end effector can be less than the contact forces. Rosales et al. [36] established a grasp contact model by introducing flexibility into the joint points and contact points of the robotic hand. Then, they analyzed the contact accessibility, object impedance, and manipulation force controllability as grasp constraints and finally achieved the force-closure grasp. Jia et al. [37] proposed a grasping algorithm based on the volume and flattening of a generalized force ellipsoid. They used the maximum volume of a generalized external force ellipsoid and the minimum volume of a generalized contact internal force ellipsoid as the objective function to establish an optimal grasp planning method to achieve the minimum internal force stable grasp of the three-finger dexterous hand, as shown in Fig. 4.

In general, multiple grasping methods exist for the same target to satisfy the force-closure. Mostly, optimal force-closure grasp synthesis concerns determining the contact point locations so that the grasp achieves the most desirable performance in resisting external wrench loads [14]. Many researchers have used this as a heuristic method. They optimize the objective function according to the predefined grasp

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

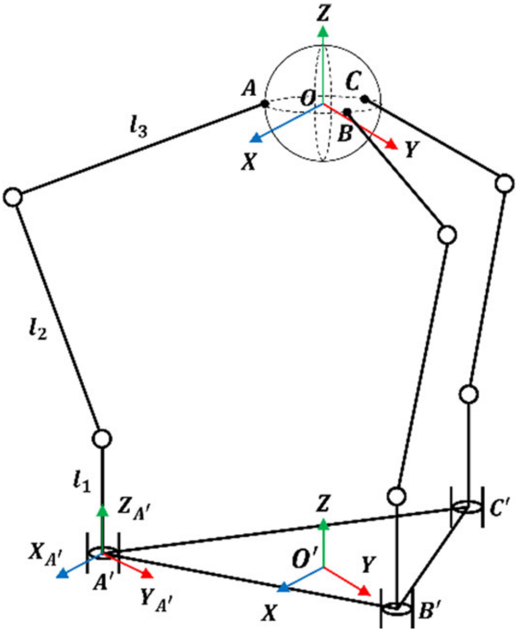

Figure 4. *Schematic figure of the grasping method proposed in ref. [37]: grasping a sphere with a three-finger dexterous hand.*

quality criteria to compute the optimal force-closure grasp. For example, Lin et al. [38] used the elastic deformation energy equivalent principle to calculate the optimal force-closed grasp and proposed a quality metric theory based on the grasping stiffness matrix. Ferrari et al. [39] solved the problem of optimal force-closure grasp by computing the maximum sphere in force screw convex space, which is easy to calculate but has limited applications. Mo et al. [40] also took the maximum force screw as a performance index to optimize the grasping position. Under the constraint of force-closure, an optimization model between the grasping position and maximum force screw was established. This method offsets the limitation that the generalized force ellipsoid is dimensionless to express the grasping effect clearly.

All of these methods designed various stability criteria to find the optimal grasps. After studying a variety of human grasps, the authors in [41] conclude that the choice of a grasp is determined by the tasks to be performed with the object. As a result, many researchers studied and addressed the task-oriented grasp, which will be introduced in the next section.

3.3. Task-oriented grasp

A good grasp plan is usually task-oriented, but there are few studies on task-oriented grasp for two main reasons: (1) it is complicated for modeling tasks, and (2) a single criterion lacks generalization ability, and different grasping criteria need to be designed for different tasks. Therefore, Li and Sastry [42] modeled the task by setting a 6D ellipsoid in the object wrench space (OWS) and designed three grasp criteria: the smallest singular value of the grasping matrix, the volume in wrench space, and the task-oriented grasp quality, which achieved ideal evaluation results. The problem with this approach is how to model the task ellipsoid for a given task, which the authors state to be quite complicated.

Pollard et al. [43] considered that a task is characterized as the wrench spaces that must be applied to the object by the robot to complete the task objective. If nothing is known about the grasping task and each wrench direction is assumed to occur with equal probability as a disturbance, the task wrench space (TWS) can be modeled as a unit sphere. Nevertheless, this approach lacks a physical interpretation since

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

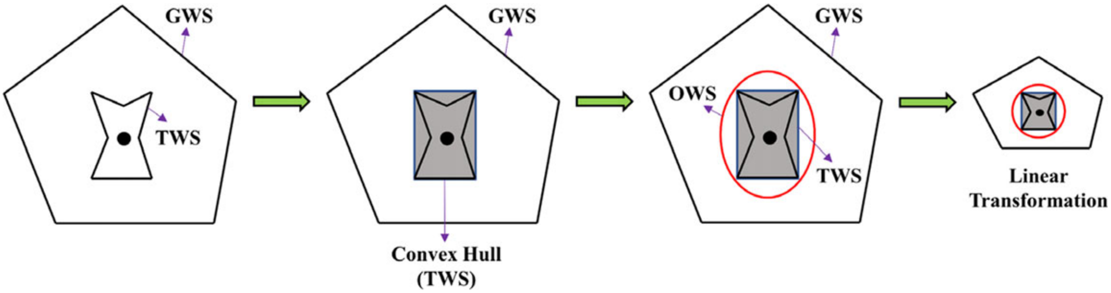

Figure 5. *An ellipsoid is used to approximate the object wrench space.*

wrenches occurring at an object boundary are not uniform, and the computed grasps are unlikely to be perfect for a given task or object. Therefore, they modeled the TWS as an OWS, incorporated the geometry of the object into the grasp evaluation, and then considered the effect of all possible disturbances on the object to the task and evaluated the grasping quality by scaling the TWS and OWS. Since OWS contains all spirals generated by disturbing forces which could act anywhere on the surface of an object, it is possible to generalize any task and model the TWS with OWS if the grasping task is unknown. Borst et al. [44] approximated the OWS as an ellipsoid and fitted it to a linearly transformed TWS to obtain another representation of the TWS. For a given TWS, the maximum scale factor is searched in order to place it into the grasping wrench space (GWS) (as shown in Fig. 5), and the grasp quality is then obtained by comparing the TWS with the GWS.

Considering the complexity of the TWS modeling process, some researchers have used novel devices or technologies that can more easily complete the given grasping task. EI-Khoury et al. [45] proposed a task-oriented approach based on manual demonstration and sensor devices. Firstly, the operator demonstrated the given task and obtained the force or moment through the sensor. Then, they modeled the task and calculated the grasp quality according to the task compatibility criterion. The experimental results show that the proposed method can be adapted to different hand kinematics models. Deng et al. [46] studied the reach-to-grasp (RTG) task and proposed an optimal robot grasp learning framework by combining semantic grasp and trajectory generation. Through experimental verification, this learning framework can enable a robot to complete the RTG task in the unstructured environment.

3.4. Summary

The application of analytic methods can accurately detect the robot grasp configuration with superior mechanical properties or satisfy task requirements from the image, which is widely used in early research. However, the quality of the detection results largely depends on the exact geometric model of the object and robotic hand, and there are certain limitations in the practical application:

1. First, it is not easy to obtain accurate geometric models of objects and manipulators, and there are always subtle differences between the actual objects and the geometric models.
2. With the transformation of the robot operating environment from a structured environment to an unstructured environment, there will be various errors in the environment, such as model errors, control errors, and noise. Therefore, the grasp detection results based on analytic methods have poor adaptability in an unstructured environment.
3. For complex geometric models, it is very time-consuming to calculate stable grasp poses by the analytic methods, which significantly reduces the robot’s work efficiency, and it is difficult to satisfy the real-time requirements of the actual grasp process of the robot, and it is impossible to grasp objects with unknown models.

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

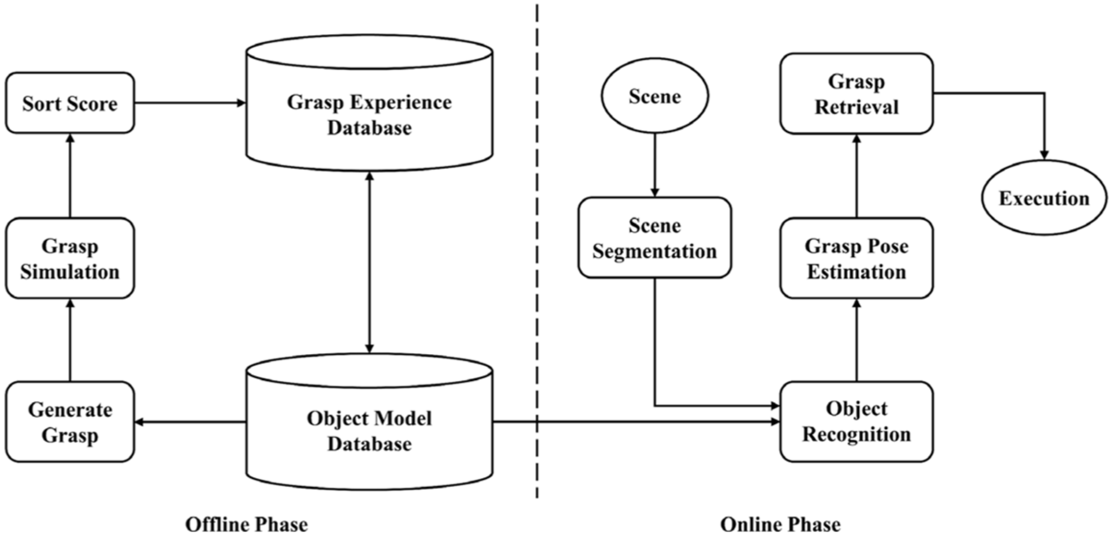

Figure 6. *Typical process for grasping known objects.*

4. Grasp detection technology based on data-driven methods

The data-driven methods rely mainly on the previously known successful grasp experience and can be classified in various ways. Firstly, it can be classified according to the applied algorithm, that is, whether the system uses heuristic or learning methods for grasp detection [47]. Secondly, it can be classified according to the perceived information as model-based and model-free grasp detection [48]. It can also be divided into single-object scene and multi-object scene grasp detection according to the number of target objects [10]. Furthermore, the grasp detection of known and unknown objects can be classified according to whether the system has previous grasp experience with the targets [18]. The last classification methods better reflect the characteristics of data-driven methods, which will be introduced specifically in the following sections.

4.1. Grasp detection of known objects

A known object usually has a complete 3D geometric model and grasp poses set. The robot can access this set and choose a good grasp pose that already exist in the set before performing the grasp operation. This set is generally constructed offline and called the grasp experience database. Figure 6 shows the basic process of grasping a known object.

In the offline stage, the models in the object model database are first analyzed to generate a number of grasp poses. Then, each grasp candidate was simulated and scored according to the simulation results. Finally, each grasp position is sorted according to the score, and the mapping relationship between the grasp pose and the grasp experience database is established for grasp retrieval in the online stage.

In the online stage, the target object is first segmented from the scene. Then, the recognition of the object and the estimation of the grasp pose are performed. After that, an existing grasp pose is retrieved from the grasp experience database according to the pose estimation results. Finally, the robot performs the grasp operation based on the retrieved pose results.

For the grasping of known objects, its related research mainly focuses on two points. The first is how to establish an offline grasp experience database. The second is how to execute object recognition and pose estimation. According to the different ways of establishing the offline database, the grasp detection methods of known objects can be divided into three categories: direct analysis methods based on 3D models, demonstration methods, and trial-and-error methods, which will be introduced in the following sections.

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

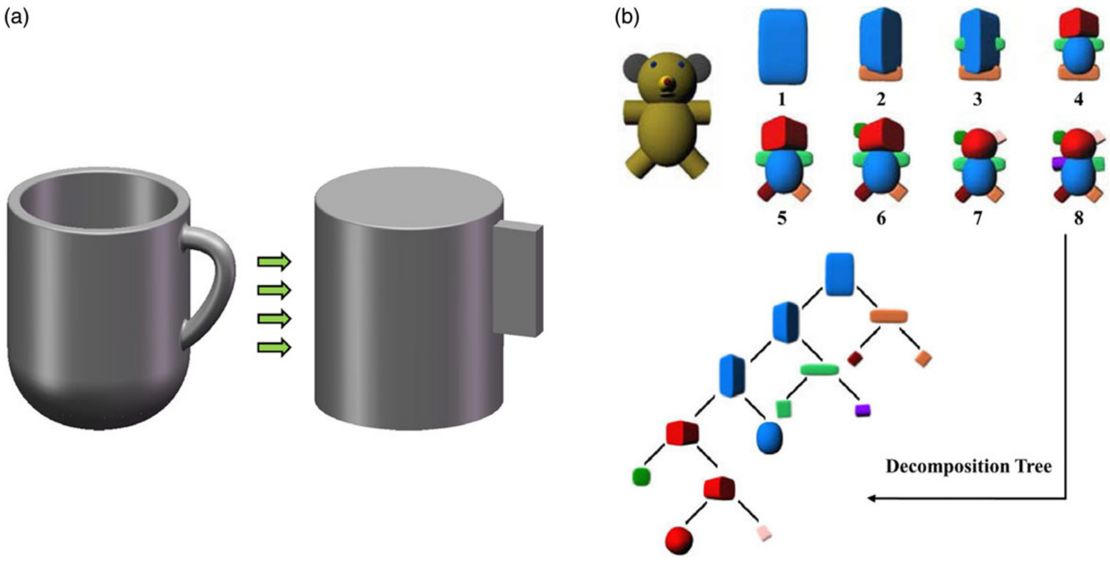

Figure 7. *The generation of grasp poses based on the shape primitive decomposition method. (a) Decompose the target object (mug) into two basic geometric models: a cylinder and a cuboid. (b) For models with complex geometric shapes, the superquadratic decomposition tree is established (the figure is from ref. [51]).*

4.1.1. Direct analysis methods based on 3D models

This kind of method needs to assume that the 3D model of the object is known, and the difficulties include how to generate good grasp poses automatically, how to set the evaluation criteria of the grasp poses, and how to sample the grasp poses on the object's surface.

Early representative research methods include the pose generation method based on shape primitive decomposition proposed by Miller et al. [49]. In this method, the target object is firstly decomposed into simple basic geometric models (sphere, cylinder, cone, etc., as shown in Fig. 7(a)), and a series of robot pre-grasp poses are generated by combining the grasp of these basic geometric models. Then use the "Gasp It!" [50] grasp simulator to test the feasibility of the grasp poses and evaluates the grasp quality, and finally the construction of the grasp experience database is completed. Goldfeder et al. [51] further studied this method and proposed the concept of the superquadratic decomposition tree. For models with complex geometric shapes, the whole grasp space of the model can be divided into multiple small grasp subspaces and arranged in the form of a tree, as shown in Fig. 7(b).

Considering the shape primitive decomposition method has low computational efficiency and accuracy, Pelossof et al. [52] used support vector machine (SVM) algorithm [53] to establish a regression mapping among object shape, grasp parameters, and grasp quality. After training, this regression mapping can effectively estimate the grasp parameters with the highest grasp quality for the new shape parameters. However, this method is simulated and verified on a grasping simulator, which has certain limitations for grasping real objects. Hereto, a pre-grasp pose generation method based on abstract image matching for grasping simple geometric models from unstructured scenes was proposed in ref. [54]. Firstly, the 3D point cloud of the target object and scene is obtained and converted into an annotated graph. Each node in the graph represents the detected simple shape or scene, and each edge stores the relative pose of the primitives. Next, objects in the scene are located by matching parts of the target graph in the scene graph, and a rigid transformation is calculated to verify the pose of the original model in the scene (the whole object recognition process is shown in Fig. 8). Finally, the estimation effect of the grasp poses is evaluated by comparing whether the point clouds between the rigid transformation results and the initial scanning results have enough overlap, and the evaluation results are sorted and input to the grasp simulator to complete the construction of the grasp experience database.

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

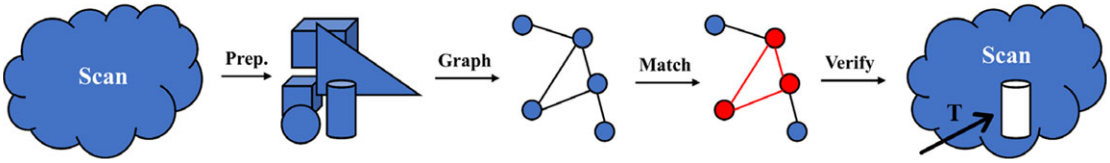

Figure 8. *The object recognition process: input scan → fast preprocessing, primitive detection → abstract graph generation → match, transformation estimation, and verification.*

4.1.2. Demonstration methods

The basic principle of this method is to let the robot learn how to grasp objects by observing and imitating the operator's grasping actions. During this process, two perceptual actions are carried out simultaneously: one is to recognize the object and the other is to record the grasp pose. Finally, the object model and the corresponding grasp pose are saved as a grasp example.

In daily life, the human hand can easily grasp objects of various shapes and sizes. However, the complexity and versatility of the human hand make the classification of grasp challenging. In general, the human hand has 24 DOFs, and each DOF is not independent. Cutkosky [41, 55] conducted a detailed classification study on the manual grasp of the target object and divided it into 16 grasp types. Subsequently, Kjellström et al. [56] proposed a vision-based grasp classification method based on this classification method, established a mapping relationship between manual grasp and robot grasp, and stored the mapping relationship into a locality-sensitive hashing (LSH). It is shown that good grasp results can be achieved by using LSH to retrieve the robot pre-grasp pose, which corresponds to a certain manual grasp type. Feix et al. [57] referred to the classification method in ref. [55] and classified each grasp into three categories according to the precision or power of the manipulator when grasping objects: power grasp, intermediate grasp, and precision grasp. Then, they further extended the taxonomy based on the number of fingers in contact with the object and the position of the thumb. At last, 33 unique prehensile grasp types were extracted. Subsequently, Cini et al. [58] optimized the taxonomies in refs. [55] and [57] according to the shape of the object and the hand joints used in grasping, finally classified the manual grasp into three major categories and 15 subcategories with a total of 28 grasp types, as shown in Fig. 9.

The classification of grasp types has been used in human demonstration [57], where the human action is to be imitated by a robot, as well as an intermediate functional layer mapping human hand grasp kinematics to artificial hands [59]. Balasubramanian et al. [60] guided the robot to produce different grasp poses through the physical interaction between the human hand and the robotic arm and recorded these poses. By comparing the grasp poses generated by manual guidance and the grasp poses independently generated by the robot, it is found that they are similar in grasp effect, but the former has better stability. Ekvall et al. [61] proposed a grasp pose generation method based on shape primitives and manual demonstration, as shown in Fig. 10, where the demonstrator wears a data glove, which is used to collect the motion data of the human hand when grasping the target object. After that, the whole grasp process is mapped to the 3D space. By collecting a large amount of data, the robot can learn the grasp habits of human beings and finally complete the grasp operation. The experimental results [62, 63] show that when the number of objects in the grasp space is 5, the grasping success rate of this method is about 100%. When the number is 10, the grasping success rate is about 96%.

In order to better learn the experience of human hands, it is necessary to understand the deep meaning and mechanism of human operation. Lin et al. [64] considered that the position of the thumb and the grasp type of the human hand are two key characteristics of human hand grasping. Based on these two critical features, they proposed a grasping strategy based on human demonstration learning. By extracting these features from human grasp demonstrations and integrating them into the grasp planning process, a feasible grasp for target objects was generated. This strategy is applied to the simulation and real robot system to grasp many common objects in life, and the effectiveness of the algorithm is verified. Deng et al. [65] built a visual analysis framework based on an attention mechanism for robot

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

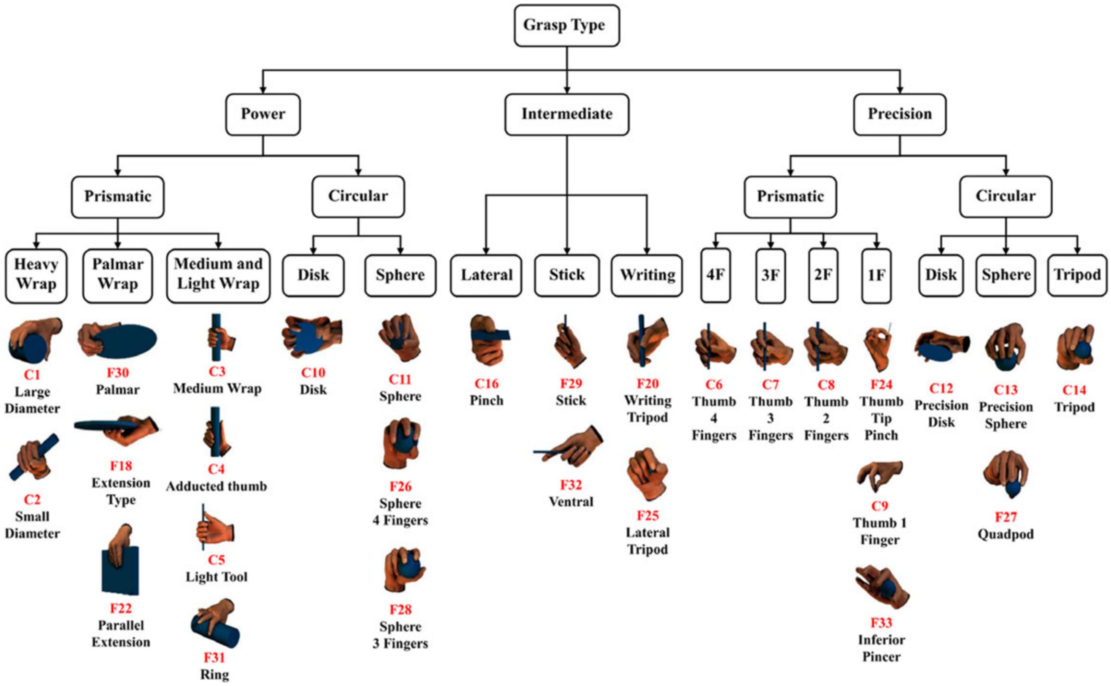

Figure 9. *Classification method of grasp in ref. [58]. There are three top-level categories: power, intermediate, and precision grasps. Power and precision grasps are both subdivided into prismatic and circular types. According to further classification at a higher level of detail, 15 categories and a total of 28 grasp types are finally obtained (in Fig. 9, the images numbered C are from ref. [55], and the images numbered F are from ref. [57]).*

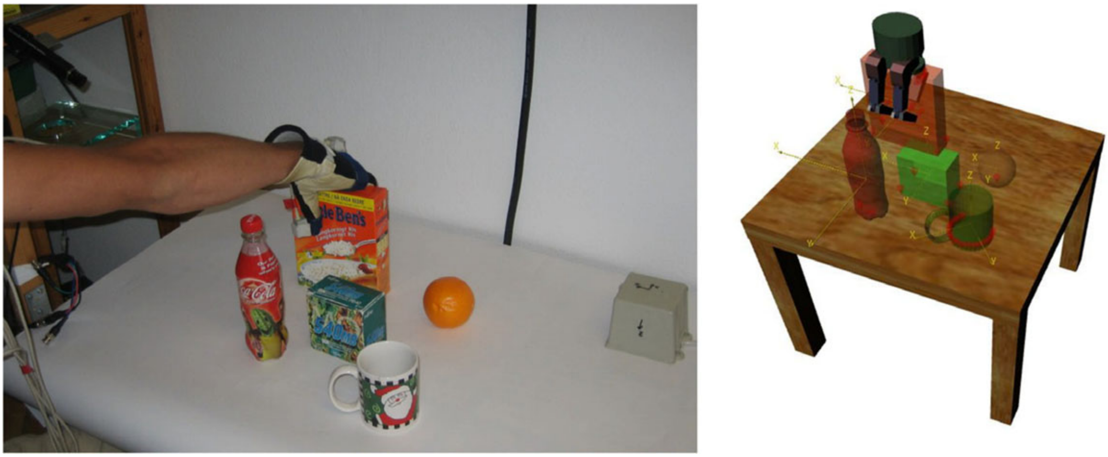

Figure 10. *The robot is guided to grasp by manual demonstration. Left: The human is moving a box. The system recognizes which object has been moved and chooses an appropriate grasp. Right: The robot grasps the same object using the mapped version of the recognized grasp (the images in Fig. 10 are from ref. [61]).*

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

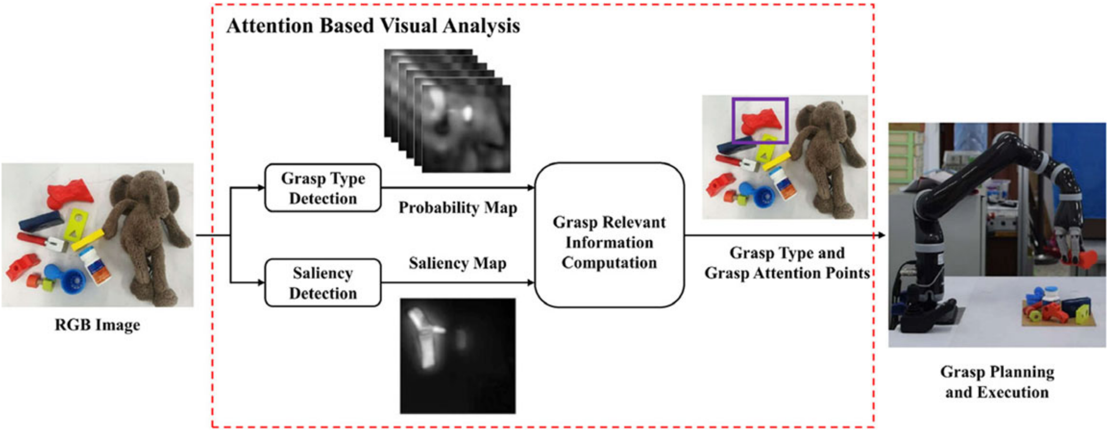
Attention Based Visual Analysis
Grasp Type Detection
Probability Map
Saliency Detection
Saliency Map
Grasp Relevant Information Computation
Grasp Type and Grasp Attention Points
RGB Image
Grasp Planning and Execution

Figure 11. *The attention-based visual analysis framework proposed in ref. [65]. Using RGB images as input, the ROI was selected using the saliency map generated by a saliency detection model. Inside the ROI, the grasp type and grasp attention points were calculated according to six probability maps produced by the grasp-type detection network. The robot is guided to grasp according to the obtained grasp type and grasp attention points.*

grasp operations, as shown in Fig. 11. The framework takes the RGB image containing the target object and scene as input, then use the computational visual attention model to select the regions of interest (ROI) in the RGB image and use the deep convolutional neural network (CNN) to detect the grasp types and key points of the target object which are contained in the ROI, as the basis for the execution of the grasp operation.

4.1.3. Trial-and-error methods

This kind of method considers that the grasp pose of the target object is not constant but needs to be improved by continuous debugging and repeated trials. Specifically, according to the type or shape of the object, a new grasp pose is generated or selected in the grasp experience database, and the robot is controlled to complete the grasp operation and evaluate the grasp performance. Finally, the database is updated according to the evaluation results.

In essence, the trial-and-error method is a process of continuous learning of the existing grasp poses of the known objects. According to this, Detry et al. [66] proposed a probabilistic method for learning and representing the grasping ability of objects. This method uses the grasp density to build the grasp affordances model of objects and connects the grasp pose of the target object with the probability of successful grasping. By controlling the robot to repeatedly complete the grasp operation for an object, it can constantly learn and update the obtained grasp pose. When a relative optimal solution is obtained, the kernel density estimation (KDE) [67] is used to convert it into a grasp density. The experimental results show that the robot can effectively select the grasp method with the highest probability of successful grasp in most cases, even when the external environment is complex or the target objects are placed irregularly. Kroemer et al. [68] proposed a hierarchical control architecture for the problems of “how to determine the location of the grasping object” and “how to perform the grasping operation,” as shown in Fig. 12. The controller consists of an upper level based on reinforcement learning and a lower level based on reactive control, where the upper level decides the location of the grasped object and the lower level decides how to execute the grasp operation. The generated grasp operation will be fed back to the upper level in real time for the reward function calculation, and an ideal experimental effect is obtained.

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

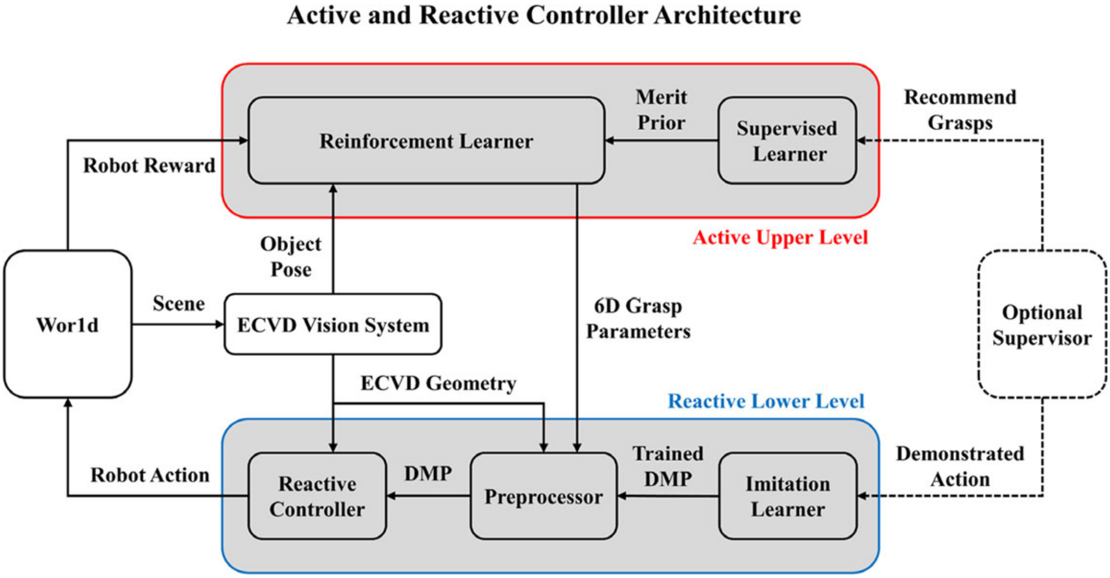

Figure 12. *The controller architecture proposed in ref. [68]. The controller consists of an upper level based on reinforcement learning and a lower level based on reactive control. Both levels are supported by supervised or imitation learning. The world and supervisor are external elements of the system.*

However, applying model-free direct reinforcement learning to practical grasp operations remains extremely challenging. The papers [69–72] illustrate several reasons for this.

1. Usually, the grasp operation involves physical contact, and the transition from noncontact to contact leads to discontinuities in the cost function. Furthermore, using reinforcement learning to compute the discontinuous cost function can cause large errors and low learning speed [69, 70].
2. In the actual grasping, the end point of the movement should adapt to the pose and shape of the goal. However, direct reinforcement learning has only been applied to learning the path of the movement, not the end point [71].
3. If the robot can complete the grasp operation at the expected position, then using reinforcement learning for learning can achieve good results. However, the the object's actual position may deviate from the expected position for some reasons (as shown in Fig. 13). Then it requires considering all possible positions of the target object and finding a grasp pose in these positions that could maximize the expectation of successfully grasping the object [72]. Therefore, it is necessary to use reinforcement learning based on the shape and goal to learn to maximize this expectation in order to generate motion primitives that are robust to object position uncertainty.

Stulp et al. [71] proposed a simple, efficient, and object model-independent reinforcement learning algorithm named Policy Improvement with Path Integrals (PI^2). This algorithm can learn both the shape and goal of the motion primitive. When the object's pose is uncertain, the learning of shape and goal can be used to obtain motion primitives with higher robustness. Experimental studies show that after learning with the PI^2, the robot can successfully grasp all the perceived objects in the 40 cm × 30 cm area of the table (the position of the objects is uncertain).

The above is an introduction to the grasping detection technology of the known 3D model. Although these methods have high accuracy, they still have certain limitations on how to build accurate 3D models of objects and how to adapt to grasp in different environments. On the one hand, it is often difficult to obtain an accurate 3D model of the object in practice, and the process requires a lot of time to sample

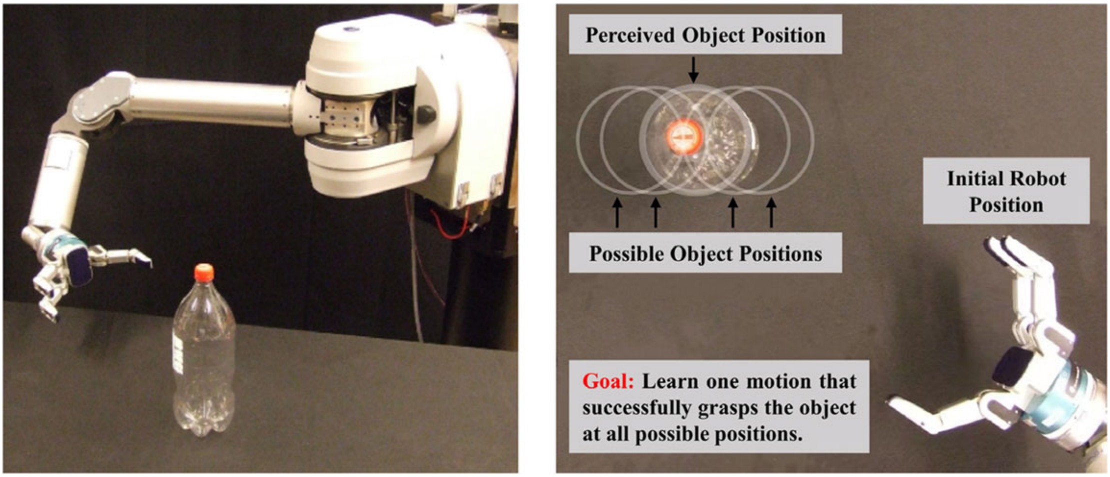
Perceived Object Position
Initial Robot Position
Possible Object Positions
Goal: Learn one motion that successfully grasps the object at all possible positions.

Figure 13. *Left: The manipulation platform used in ref. [71]. Right: The PI^{2}algorithm is used to learn the goal and shape of a motion primitive, to obtain a motion that can grasp the object in all possible positions.*

and model the object's data. On the other hand, the grasp detection for a known 3D model has great limitations in the practical application, and it is unsuitable for use in the unstructured environment with a wide variety of objects.

4.2. Grasp detection of unknown objects

An unknown object usually has an uncertain physical model and no prior grasping experience. Different from grasping a known object, when the robot grasps an unknown object, it needs to compare the unknown object with the previously grasped object and estimate the grasp pose of the object through relevant methods. In this paper, the grasp detection technology of unknown objects is divided into two categories: perception-based and learning-based methods, which will be introduced in the following sections.

4.2.1. Perception-based methods

In the actual grasp process of unknown objects, the robot can only perceive part of the information from the outside world, such as RGB and depth information. Therefore, the robot needs to use incomplete information to generate a good grasp pose. Perception-based approaches focus on identifying structures or features in the data to generate and evaluate grasp candidates. By referring to the relevant paper, there are mainly two ways to generate grasp poses through perception.

The first is to extract the 3D or 2D features from the segmented point cloud or image data and then perform the grasp detection heuristically based on these features. Dunes et al. [73] sampled the contour features of objects from multiple angles and generated a 2D curve according to the sample points. Then the contour of objects was estimated through the quadratic curve. Finally, the robot's grasp direction and configuration are inferred by the long axis and center of mass of the curve. Detry et al. [74] proposed a grasp strategy transfer method, which generated candidate objects by extracting fragments of the target point cloud, then clustered the generated candidates through nonlinear reduction and unsupervised learning algorithm, and finally selected the center of clustering as the newly generated grasp prototype to grasp new objects. The whole process is shown in Fig. 14.

In the process of heuristic grasping based on the extracted features, some researchers also pay attention to the grasping robustness and efficiency. Hsiao et al. [75] proposed a simple but robust reactive

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

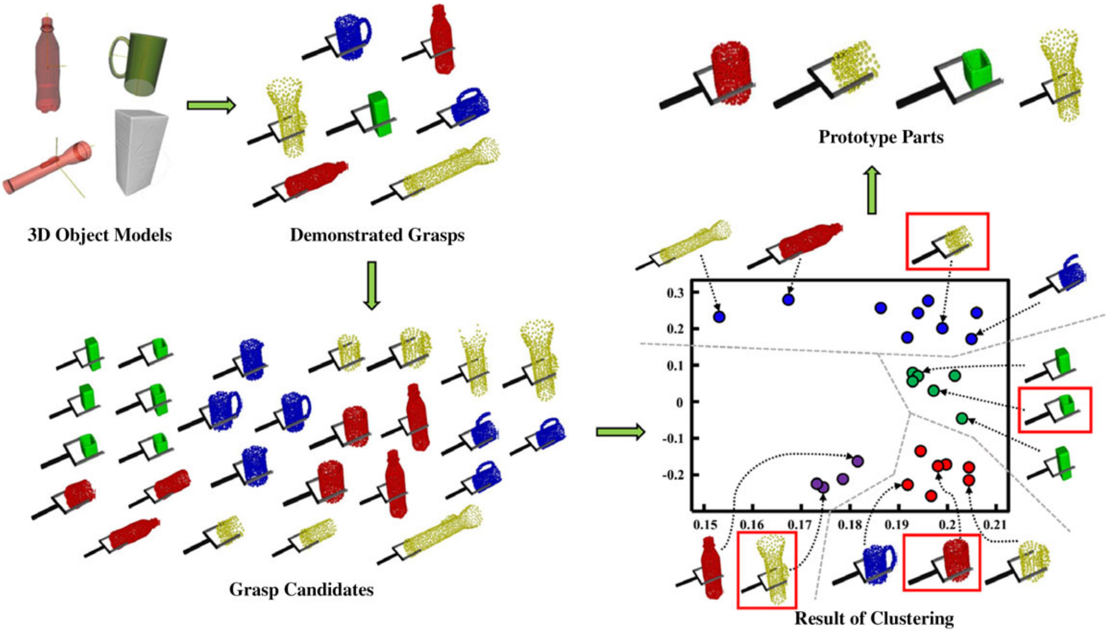

Figure 14. *The generalizing grasp strategies proposed in ref. [74]. In the experimental data, three of the objects are cylinders of different sizes, and one is a cuboid. According to the four kinds of objects, seven grasps are demonstrated, and 27 grasp candidates are computed. By clustering the grasp candidates, the central elements of the clusters are selected as the prototype parts for grasping new objects (the images in Fig. 14 are from ref. [74]).*

adjustment approach for grasping unknown objects, which acquires the target point cloud in the scene by a 3D sensor and calculates the bounding box of the point cloud using principal component analysis (PCA) algorithm, then generates the grasp candidates using a heuristic method based on the overall shape and local features of the target, and finds the optimal grasp result by using a feature weight table (such as the number of point clouds in the bounding box, whether to grasp at the boundary, the distance between the fingertips and the object center along the approach direction, etc.). Finally, the tactile sensor on the robot’s end effector is used for real-time monitoring of the grasp. When the shape or position of the object is uncertain, it can be corrected in time to obtain a robust grasp. Liu et al. [76] proposed a method to quickly grasp unknown objects. By installing a 2D depth sensor on the robot, partial shape information of unknown objects was obtained, and then features were extracted from the partial shape information to determine the grasp candidate points of unknown objects (as shown in Fig. 15). At last, the feasibility of the grasp candidate points is judged by checking whether the robot can grasp and lift the object successfully. This method does not need to acquire and process all the target information and can reduce the grasp time. Relevant experimental results also verify the feasibility of this algorithm.

The second is to directly fit or estimate the basic geometry of the object based on the existing segmented results and then to plan the grasp based on the geometric shape. Morales et al. [77] used visual feedback information to guide the robot to grasp and proposed an intelligent algorithm for two-finger and three-finger grasping. The algorithm considers the force-closure and contact stability conditions during grasping, and the grasp candidates of the planar objects can be selected directly according to the geometry information. They also used this algorithm to control a Barrett hand to grasp a lot of nonmodeled planar extruded objects and obtained good grasping results. Richtsfeld and Vincze [78] developed a novel vision-based grasp system for unknown objects based on range images and applied it to the grasp of table objects. The system first uses a laser scanner to acquire the point cloud of the scene and preprocess the raw data and then uses the RANSAC algorithm [79] and the 3D Delaunay triangulation

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

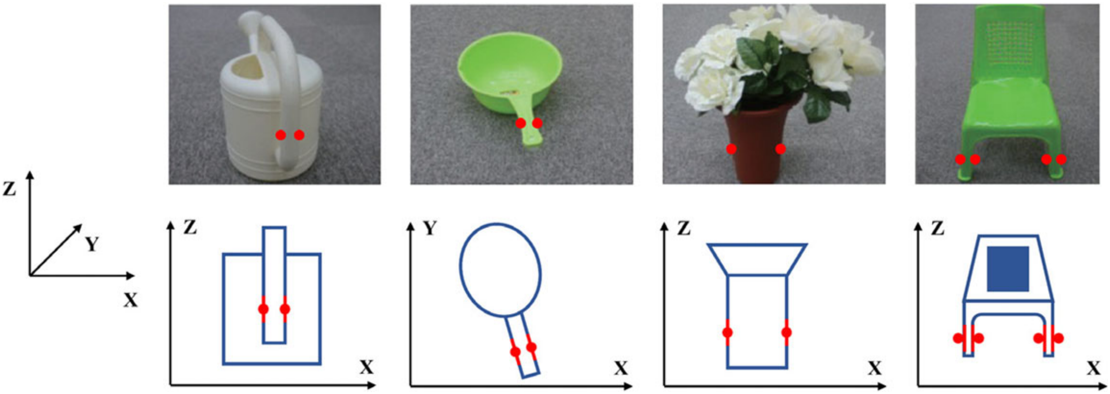

Figure 15. *Objects for robotic grasp used in ref. [76]. Top row: Every object has parallel surfaces or parallel tangent planes, and the red points are the grasping points of the objects. Bottom row: The 2D shapes of objects are obtained by projecting the 3D models into the XY or XZ plane.*

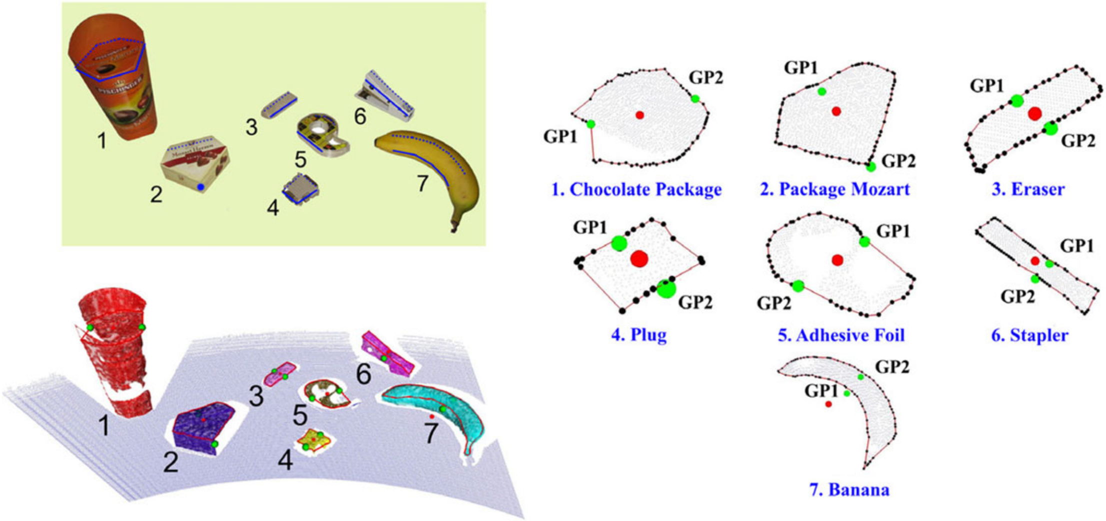

Figure 16. *Table scene with seven different objects in ref. [78]. Left: The actual models and 3D point clouds of the objects. The green points represent the grasp points, and the red points are the calculated centroids of different top surfaces. Right: Top surfaces of the seven objects. The red and green points represent the same meaning as in the left figure. GP1 is the first grasp point with the shortest distance to the centroid, and GP2 is the second grasp point.*

algorithm [80] to segment the table and object point clouds, respectively. Finally, 2D Delaunay features of the top surface of the object were obtained, the feature edge points and surface centroid of each object were detected, and the location of the grasp point was determined according to the principle of minimum distance between edge points and surface centroid, as shown in Fig. 16. Bohg et al. [81] proposed a method to estimate the complete object model from a local view by assuming that the target object satisfies the symmetry condition. In this way, the whole 3D model of the target object is estimated by complementing the original 3D point cloud of the object so that the grasping ability can be estimated by using the grasp method of the known object.

In summary, Table I organizes the methods presented above, mainly including the classification of perception-based grasp detection methods, and the detection results of related methods.

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

Table I. *Summary of perception-based grasp detection methods.*

Classification		**Method**	**Result**	**Reference**
Perception-based methods	Extracting image features and using heuristic methods	2D curve estimation	The fitting error of the object shape after 27 iterations was 0.081 mm, and no robot grasp experiment was conducted.	[73]
		Grasp strategy transfer	It has certain grasping generalization ability for cylinder and cube shape objects of different sizes.	[74]
		Reactive grasping	Achieved 66 successful grasps and 60 open-loop grasps out of 68 attempts.	[75]
		Quickly grasp	Six kinds of daily necessities were selected as experimental objects, and the grasping success rate was 91.6%. Compared with the method in ref. [134], the execution time of grasping is reduced by 45%.	[76]
	Direct estimation or fitting	Visual feedback	The two-finger hand and three-finger hand are used to grasp six kinds of objects, and the candidate grasp points of the target object in a single picture can be successfully calculated within 0.04 s and 0.24 s.	[77]
		Depth image-based grasping	The grasp experiments were conducted for seven kinds of objects, the grasping success rate was 85.71%, and the grasp points detection took about 30 s.	[78]
		Estimate the whole from the parts	The average error of the overall point cloud estimated from the local point cloud was 7 mm in all directions for 12 household objects and toy models, but no robot grasp experiment was conducted.	[81]

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

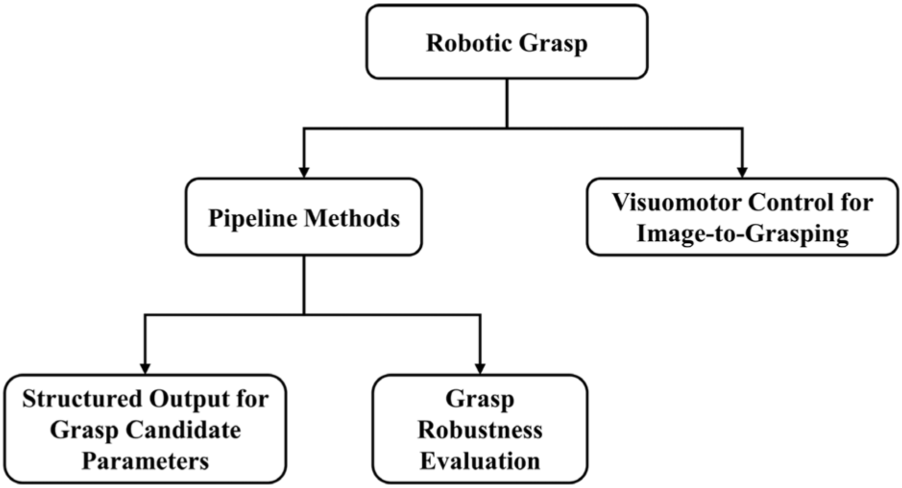
Robotic Grasp
Pipeline Methods
Visuomotor Control for Image-to-Grasping
Structured Output for Grasp Candidate Parameters
Grasp Robustness Evaluation

Figure 17. *Classification of learning-based robotic grasp detection methods.*

4.2.2. Learning-based methods

At present, machine learning methods have been shown to be applicable to most perception problems [82–86], which allow perceptual systems to learn mappings from datasets to various visual properties [87]. Machine learning-based methods for robot grasp detection are also one of the current research hotspots, which allow robots to better grasp known objects in occluded or stacked environments [88, 89], objects with known systems but uncertain poses [66], and objects with completely unknown systems [90]. In recent years, with the booming development of deep learning in image processing [84, 91, 92], more and more researchers have applied deep learning to robotic grasp detection [17, 93], allowing computers to automatically learn high-quality grasp features from large amounts of image data, which has greatly contributed to the development of unknown objects' grasp detection technology.

For unknown objects, learning-based grasping detection can be divided into two main categories [17], as shown in Fig. 17. One is the pipeline methods, through the relevant learning algorithm, to generate the grasp pose and then use a separate path planning system to execute the grasp. The other is an end-to-end grasp method based on a visual motion control strategy to map from image data to grasp actions. The first method can be further divided into two categories according to the learning content: one is to learn the structured output of the grasp parameters (such as grasp points and grasp rectangles); the other is to learn the grasp robustness evaluation. The following will focus on Fig. 17 for a detailed introduction.

(1) Learning the structured output of grasp parameters

Earlier researchers used the grasp points as a structured output of the grasp parameters. Saxena et al. [94, 95] used synthetic images as a training dataset to predict the location of the grasp points in 2D images by the supervised learning method. Then, they estimated the grasp pose corresponding to the grasp points by taking 2D images from different viewpoints. The method can complete the recognition of the grasp points within 1.2 s with a detection accuracy rate of 94.2% and a grasping success rate of 87.8%, but the grasp has some limitations because the depth images are not used. Rao et al. [96] used 3D data as input and used supervised localization to obtain the graspable segments in the scene, then estimated the target shape using local 3D information and trained the classifier using the SVM algorithm [53] with Gaussian radial basis function (RBF) kernel as a way to find a pair of optimal grasp points, and finally obtained a grasping success rate of 87.5%.

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

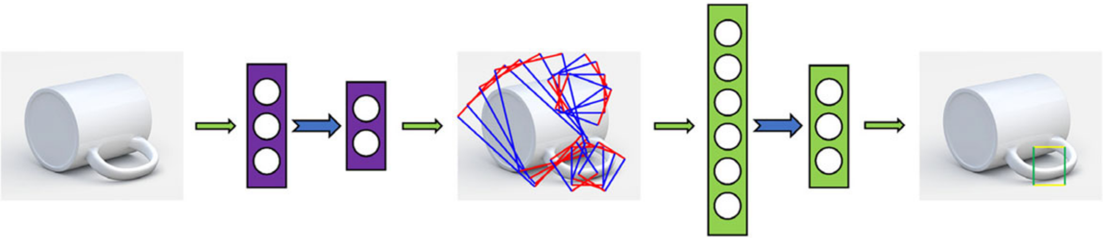

Figure 18. *Two-step cascaded system proposed in ref. [87]. Input an image of an object to grasp and a small deep network is used to exhaustively search for potential rectangles, producing a small group of top-level rectangles. A more extensive deep network is then used to find the top-ranked rectangle to produce the best grasp for the given object.*

The method of using grab points to represent structured output usually has arbitrary support areas (such as the neighborhood centered on these points), which may not match the physical space occupied by the gripper. Therefore, some researchers use the grasp rectangle as the structured output of the grasp parameters, and the most commonly used method to obtain the grasp rectangle is the sliding window method [87].

Jiang et al. [97] proposed a representation of the seven-dimensional grasp rectangle (3D location, 3D orientation, and the gripper opening width), replacing the original representation of grasp points. Lenz et al. [87] proposed a five-dimensional grasp rectangle based on the seven-dimensional grasp rectangle and designed a two-step cascaded system with two deep networks, as shown in Fig. 18. First, a small CNN is used to find all possible grasping rectangles and eliminate some of the rectangles with low scores. Then, a large CNN is used to find the highest score among the retained grasping rectangles as the optimal result to estimate the grasp pose. The grasping success rate of this method reached 75.6%, but the processing time of a single image reached 13.5 s [17], indicating that this method requires massive computation.

In recent years, due to the gradual development of relevant theories in deep learning, more and more researchers can better apply it to robotic grasp detection to improve the efficiency and accuracy of grasping. Ten Pas et al. [98, 99] designed a novel grasp pose detection (GPD) method that can locate the target object’s position directly from sensor data and does not require estimating the grasp pose. In this method, the noise and partially occluded point cloud were taken as input. Then the obtained point cloud was normalized to extract the 12-channel and 15-channel projection features of the robotic grasp closure region and constructs a CNN-based grasp quality evaluation model to generate feasible grasp poses without assuming the object CAD model. Compared with their previous research results [100], the success rate of grasping unknown objects in complex environments was improved from 73% to 93%. Wei et al. [101] proposed a multi-modal deep learning architecture for grasp detection. First, an unsupervised hierarchical extreme learning machine (ELM) was used to achieve feature extraction of RGB and depth images, and a shared layer was developed by combining RGB and depth features. Finally, the ELM was used as a supervised feature classifier for the final grasping decision. Guo et al. [102] proposed a hybrid deep architecture with a mixture of visual and tactile sensing, which uses visual data (RGB images) as the main input and tactile data as a supplement to assess the grasping stability, as shown in Fig. 19. For feature extraction, they employed the ZF model [103] to extract features from the input image and used the reference rectangle to identify all possible graspable regions in an image. In the intermediate layer, they concatenated the visual and tactile features as a joint layer. Finally, a 1 × 1 kernel is applied to slide across the joint layer, yielding the grasp detection results.

Although the sliding window method is simple in principle, it may repeatedly scan the graspable region of the image during the sliding process [17], resulting in a long processing time. Therefore, the one-shot detection method [104] emerged, which does not require iterative scanning but uses a direct

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

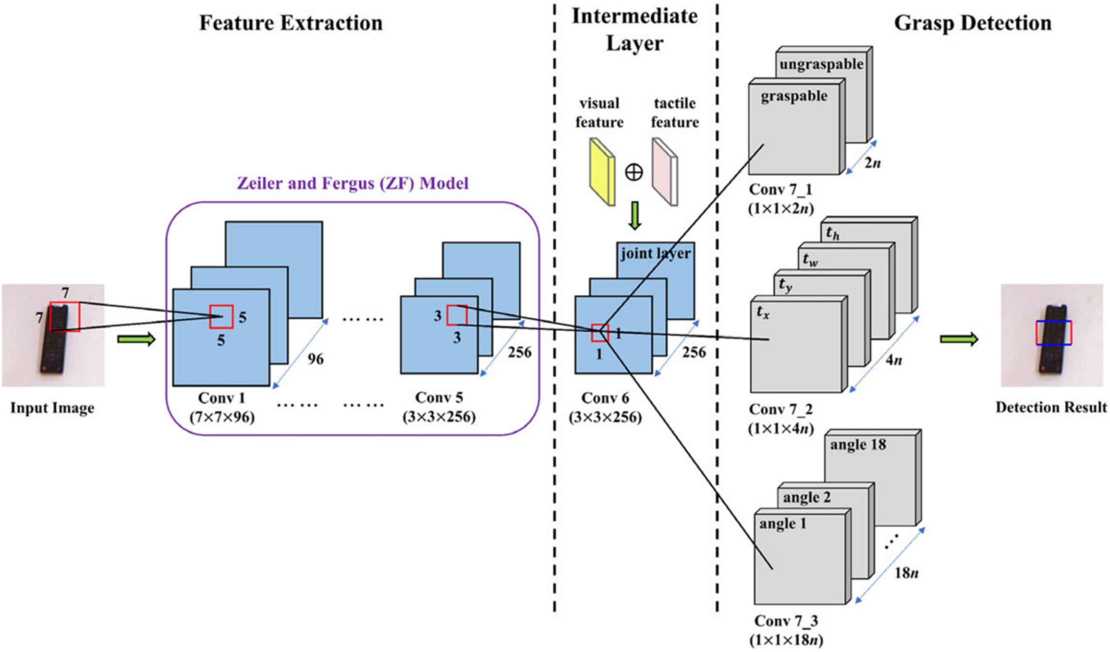

Figure 19. The architecture of the deep visual network for grasp detection proposed in ref. [102]. In grasp detection, there are two classes of labels (graspable and ungraspable) for each reference rectangle, $\{t_x, t_y, t_w, t_h\}$ are the offset coordinates for the predicted initial grasp rectangle, $\{0°, 10°, \cdots, 170°\}$ are the 18 labels for the rotation angle, and n is the number of the reference rectangle used in each location.

regression method to predict the structured output of the grasp parameters, which improves the real-time performance.

Zhang et al. [105] focused on the RGB features and depth features of images and proposed a multimodal fusion method to achieve regression of robotic grasp configuration from RGB-D images. The calculation time of a single image was 117 ms, and the accuracy of image-wise split and object-wise split was 88.90% and 88.20%, respectively. Redmon et al. [104] used a CNN to perform grasp prediction of the complete image of an object, which does not use the standard sliding windows or region proposal networks but performs single-stage regression directly on the grasp rectangle, with a processing speed of 76 ms on the GPU for a single image and a detection accuracy of 88%. In addition, this method can predict multiple grasp poses and classify them by using a locally constrained prediction mechanism. However, it cannot evaluate these grasp poses quantitatively and still has some drawbacks. Using a similar approach, Kumra et al. [11] proposed a novel multi-modal grasp detection system that uses deep CNNs to extract features from images and predict the grasp pose of the target object by shallow CNNs, achieving a detection accuracy of 89.21% on the Cornell dataset and running at 16.03 FPS, it is about 4.8 times faster than the method proposed in ref. [104].

In order to evaluate the structured grasp output more effectively, Depierre et al. [106] used a scorer to score the grasping ability of a certain position in the image. Based on this scorer, an advanced deep neural network (DNN) was extended to connect the regression of grasp parameters with the score of grasp ability (as shown in Fig. 20). The architecture achieves detection accuracies of 95.2% and 85.74% on the Cornell and Jacquard datasets, respectively, and is applied to actual robotic grasp with a grasping success rate of around 92.4%. It is not difficult to find that most of the one-shot detection methods adopt deep transfer learning techniques to use pretrained deeper convolutional networks to predict the grasp candidates from images [107, 108] and finally achieve good results.

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

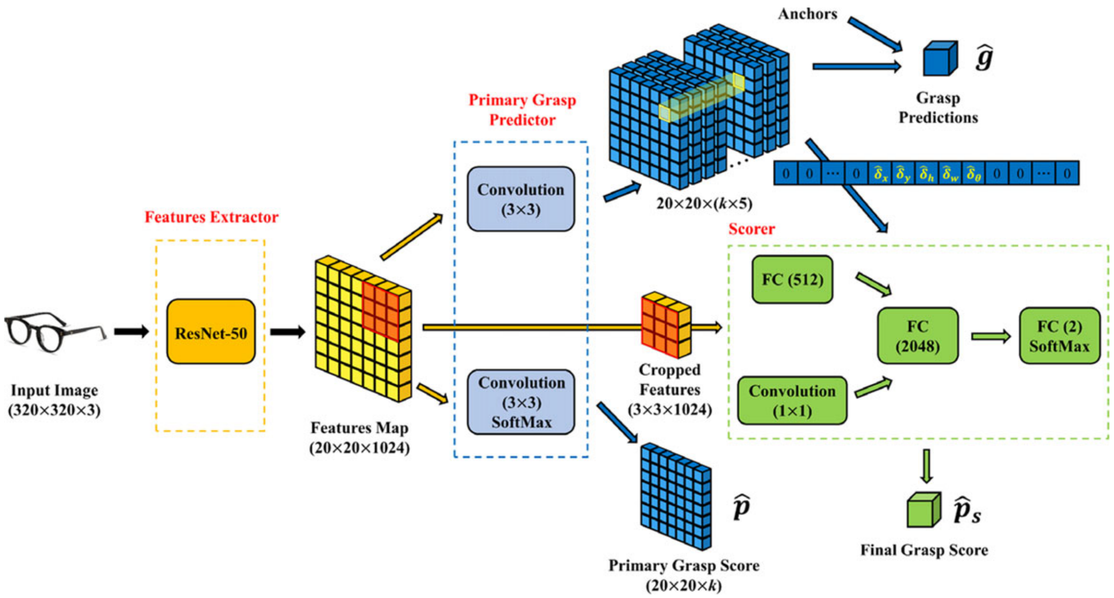

Figure 20. *A state-of-the-art grasp detection architecture proposed in ref. [106]. The architecture takes RGB-D images as input and has three components: the feature extractor, the intermediate grasp predictor, and the scorer network.*

Although using direct regression to predict the structured output of the grasp parameter is the mainstream method for one-shot detection, many researchers have combined the classification and regression techniques for one-shot detection in some special cases.

Chu et al. [109] transformed the regression problem into a combination of region detection and orientation classification problems, using RGB-D images as the input to a DNN to predict grasp candidates for a single object or multiple objects. This method achieved 96.0% image-wise split accuracy and 96.1% object-wise split accuracy on the Cornell dataset when dealing with a single object. As for dealing with multiple objects, the method has a specific generalization capability and achieves 89.0% grasping success rate when grasping a group of household objects, and the processing time of a single image is less than 0.25 s. Based on ref. [102], Zhang et al. [110] proposed a real-time robotic grasp method based on the fully CNN and used the oriented anchor boxes to predefine the region of the image. As shown in Fig. 21, the network mainly consists of a feature extraction part and a grasp prediction part, and the feature extraction part mainly takes RGB or RGB-D images as input to generate feature mappings for grasp detection. At the same time, the grasp prediction part is divided into a regression layer and a classification layer, which are mainly responsible for regressing grasp rectangles from predefined oriented anchor boxes and classifying these rectangles into graspable and ungraspable parts. The proposed method effectively improved the performance of capture detection and obtained 98.8% image-wise split accuracy and 97.8% object-wise split accuracy in the Cornell dataset, respectively. In the context of GTX 1080Ti, the fastest running speed can reach 118 FPS.

(2) Learning the grasp robustness evaluation

The grasp robustness is mainly used to describe the grasp probability of a certain position or area in the image [111], and the related grasp robustness function is often used to identify the grasp pose with the highest score as the output. Therefore, learning to grasp robustness evaluation is the core method of many deep grasp detection research. In particular, binary classification [17, 98, 112, 113] is one of the commonly studied methods. It classifies the grasp candidates into valid and invalid poses and then

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

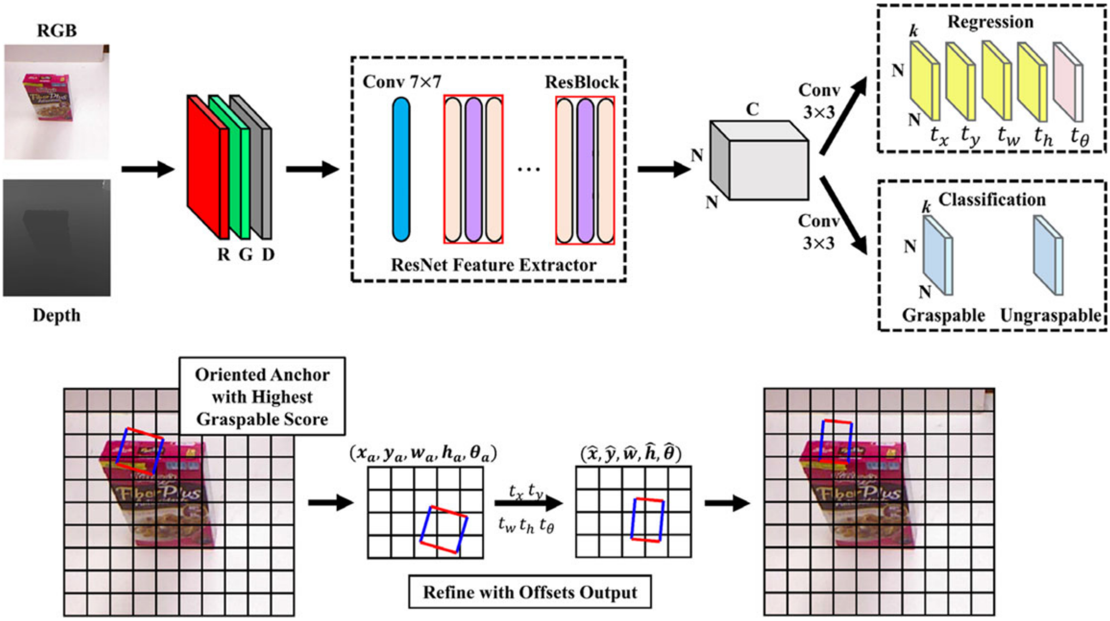
RGB
Depth
R G D
Conv 7×7
ResBlock
...
ResNet Feature Extractor
C
N
N
Conv 3×3
Conv 3×3
Regression
k
N
N
t_x t_y t_w t_h t_θ
Classification
k
N
N
Graspable
Ungraspable
Oriented Anchor with Highest Graspable Score
$(x_a, y_a, w_a, h_a, \theta_a)$
t_x t_y
t_w t_h t_θ
$(\hat{x}, \hat{y}, \hat{w}, \hat{h}, \hat{\theta})$
Refine with Offsets Output

Figure 21. *Top: The network architecture based on ResNet-Conv5 proposed in ref. [110]. The input is an RGD image, and the output includes the regression and classification results. Bottom: The process of using network output to compute the grasp prediction. First, find the oriented anchor box with the highest graspable score according to classification results. Then, the grasp prediction is calculated by the algorithm proposed in the paper.*

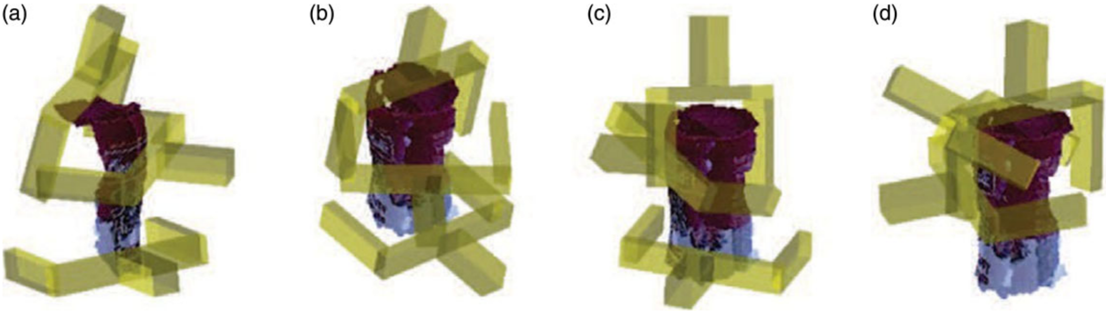

Figure 22. *Illustrations of grasp candidates were found using the algorithm proposed in ref. [98]. Each image shows three examples of a gripper placed at randomly sampled grasp candidate configurations (the figure is from ref. [98]).*

learns and evaluates them based on neural networks and robustness functions to output the best grasp poses. Ten Pas et al. [98] acquired the target point cloud based on two strategies (active and passive) and optimized them, and then generated a large number of grasp candidates by calculating the geometric features such as curvature and normal of the point cloud (Fig. 22 shows several grasp candidates generated based on this method). Subsequently, they used the end-to-end learning method to perform binary classification on the generated grasp candidates to identify the graspable areas of objects in the complex point cloud scene and finally obtained 89% detection accuracy rate and 93% grasping success rate. Chen et al. [112] proposed an edge-based grasp detection strategy, which first used the geometric

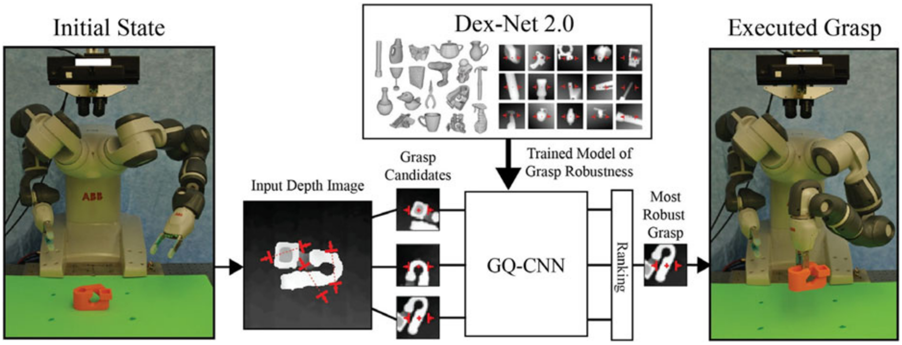

Figure 23. *Dex-Net 2.0 architecture. When performing the grasp operation, a 3D point cloud is obtained with the depth camera, where pairs of antipodal points identify a set of several hundred grasp candidates. Then, GQ-CNN is used to quickly determine the most robust grasp candidate, and the robot will perform the grasp (the figure is from ref. [116]).*

relationship between edge points to determine the approximate area of grasp candidates. Then, they used binary classification to train a lightweight CNN under a limited number of samples to identify feasible grasps. The method uses only RGB images as input, and the training time for a single image on the CPU is only 1.46 s, with a detection accuracy of 93.5%. Li et al. [113] regarded the prediction of grasp stability as a binary classification problem. In order to achieve stable grasping, a training dataset that can reflect the grasp contact force of various objects was constructed by multiple grasp operation feedback from a tactile sensor array. The optimal grasp prediction model under different scenarios was obtained by inputting the training data into different machine learning algorithms.

Besides the binary classification methods, many researchers also combined neural networks with supervised learning to obtain a robust grasp. Seita et al. [114] used the Monte Carlo sampling estimation algorithm to generate test datasets from the Dex-Net 1.0 dataset [115] and then trained them through supervised learning to estimate the grasp robustness based on the mean absolute error (MAE) and area under the curve (AUC) of the dataset. In the process of sampling estimation, they also adopted two supervised learning methods: deep learning and random forest, which increased the training speed by 1500 times and 7500 times, respectively. Mahler et al. [116] regarded the grasp robustness as a scaler probability in the range of [0,1] and synthesized a dataset named Dex-Net 2.0 (as shown in Fig. 23), which included 6.7 million point clouds and related grasp indicators. The dataset was input into a Grasp Quality Convolutional Neural Network (GQ-CNN) for training as a way to predict the robustness of grasping based on point cloud data, and the grasp detection accuracy of this method was around 93.7%, but the training process was very long due to the huge amount of data. Gariépy et al. [117] improved the Spatial Transform Network (STN) [118] and proposed a one-shot detection network: the Spatial Transform Network of Grab Quality (GQ-STN). Then, they use GQ-CNN as a supervisor and train GQ-STN to obtain grasp candidates with high robustness scores. Compared with ref. [116], it obtained a higher grasp detection accuracy (96.7%) and improved the detection speed by more than 60 times.

Training a suitable neural network for grasp detection usually requires a large amount of manually labeled data, but this requires even higher volumes of domain-specific data [89]. Therefore, some researchers have opted to collect data with self-supervised methods. Fang et al. [119] proposed a Task-Oriented Grasping Network (TOG-Net) model, which realized the joint optimization of grasp robustness and subsequent operational tasks through self-supervised learning. Based on this model, they guided the robot to grasp relevant tools and complete actual tasks (sweeping and hammering, as shown in Fig. 24), achieving a success rate of 71.1% and 80.0%, respectively. ŠEGOTA et al. [120] used the multilayer perceptron (MLP) algorithm to regress the values of grasp robustness from a robotic grasp

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

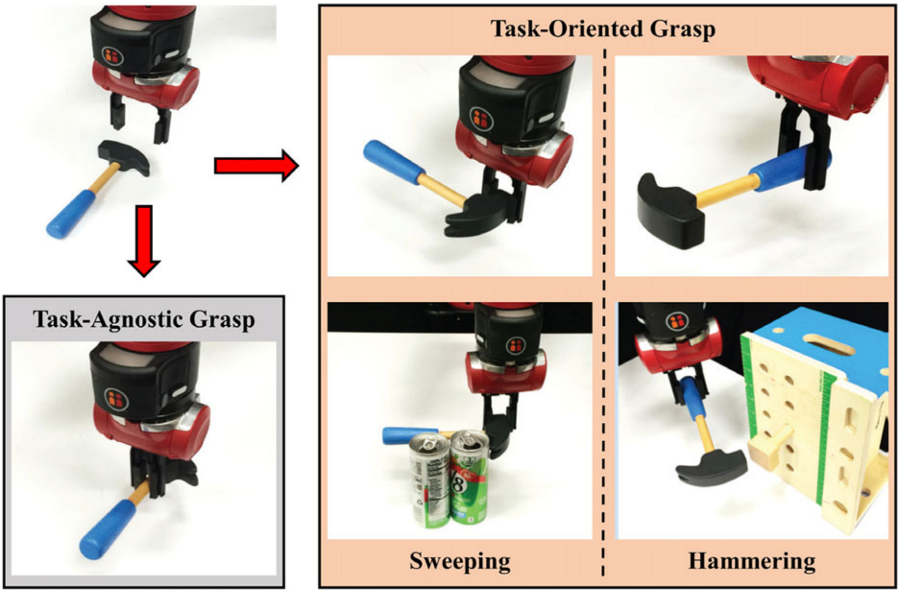

Figure 24. *For different requirements, the robot has different grasp ways. A task-agnostic grasp can lift a hammer, but it may not be appropriate for particular manipulation tasks, such as sweeping or hammering. According to the method proposed in ref. [119], the grasp selection can be directly optimized by jointly selecting a task-oriented grasp and subsequent manipulation actions.*

dataset [121] containing torque, velocity, and position information and finally obtained a high-quality regression model.

(3) The end-to-end grasp based on the visual motion control strategy

The classical machine learning approach is to extract relevant features of the original data and classify them according to the prior knowledge of human beings and then use these features as input to a model for training, which in turn outputs the final result. Usually, the training result depends on the extracted features, so early researchers spent much time on feature extraction. With the development of deep learning, it is often better to use the end-to-end method to let the network model learn by itself and extract the features. Only the original data need to be labeled at the starting stage, and then the original data and the corresponding labels are input into the model for learning to get the final result.

For robotic grasp, end-to-end can be understood as the mapping from image data to the grasp pose. By associating the end-to-end learning method with the visual motion control strategy, the visual motion controller is trained by using deep learning to iteratively correct the grasp points until the robot completes the grasp operation successfully.

In the earlier work, Zhang et al. [122] mapped the grasp operation in the 3D space to the 2D synthetic image and then used the deep Q network (DQN) [123] for training to obtain the poses of the robot's end effector when it reached the target. However, when the authentic images taken by the camera were used as input to the DQN, the probability of successfully reaching the target was only 51%. It can be seen that the learning results in the synthetic scene usually cannot be directly applied to the real environment. In this regard, Zeng et al. [124] proposed a visual control method that can identify and grasp objects in a cluttered environment. First, an object-agnostic grasping framework was used to complete the mapping from visual observation to action, and the dense pixel-wise probability maps of the affordances for four different grasping primitive actions (as shown in Fig. 25) were inferred. Then, it executed the grasp with

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

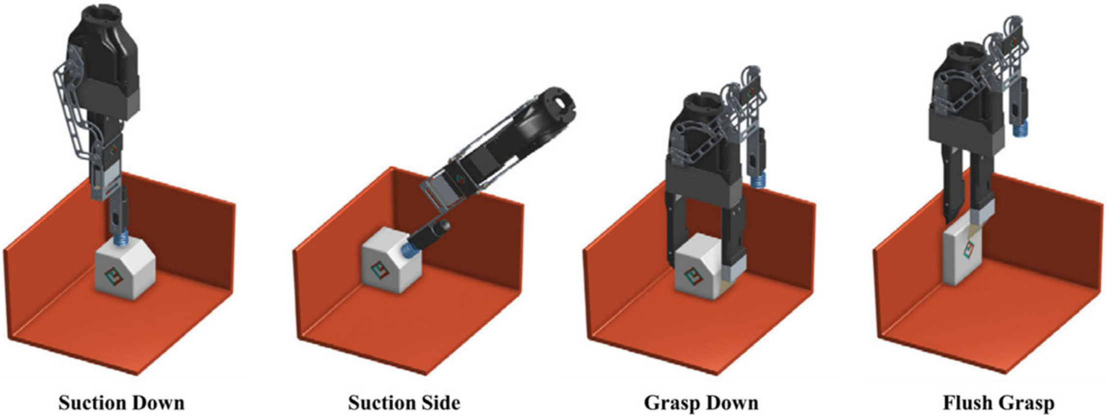

Figure 25. *Four motion primitives proposed in ref. [124] include suction and grasping to ensure successful picking for a wide variety of objects in any orientation (the figure is from ref. [124]).*

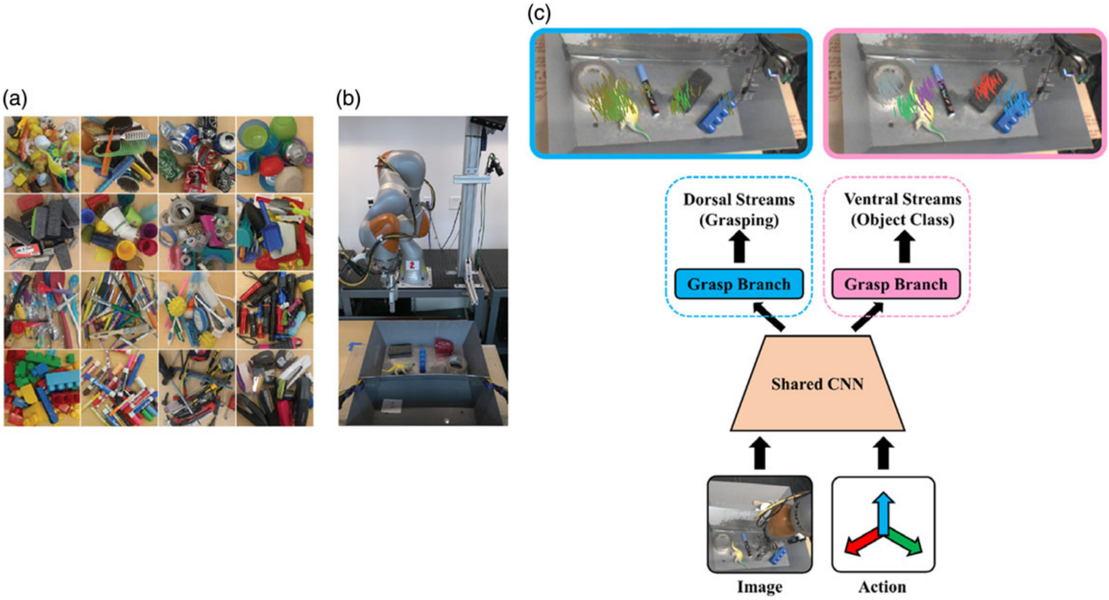

Figure 26. *The semantic grasping model proposed in ref. [125]. (a) Considering the task of learning to pick up objects from 16 object classes. (b) The robotic arm with a two-finger gripper. (c) A two-stream model that shares model parameters between a grasp branch and a class branch, which comprise the dorsal (blue box) and ventral streams (pink box).*

the highest affordance and identified the target object by using the cross-domain image classification framework, which matches observed images to product images without any additional data collection or retraining. The detection accuracy of this method is 88.6%, and the capture success rate is 96.7%. The detection time of a single image is related to the number of graspable angles.

Jang et al. [125] focused on robot semantic grasping and proposed an end-to-end learning method. By combining spatial and semantic reasoning into a neural network, the network is divided into ventral and dorsal streams, which are used to classify and select good grasping targets, respectively, as shown in Fig. 26.

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

Table II. *Summary of learning-based grasp detection methods.*

Classification of methods		Reference	Detection accuracy or grasping success	Speed	Algorithm or model	Input	Dataset
Learning the structured output of grasping parameters	Grasp points	[95]	Detection accuracy: 94.2% Grasping success: 87.8%	Single image: 1.2 s	Supervised learning (SL)	RGB	Customized dataset
		[96]	Grasping success: 87.5%	—	SL; SVM	RGB-D	Customized dataset
	Grasp rectangles (sliding window methods)	[97]	Detection accuracy: 96.8% Grasping success: 87.9%	—	Two-stage deep network; SL; SVM	RGB-D	Customized dataset
		[87]	Grasping success: 75.6%	Single image: 13.5 s	Two-stage deep network; sparse auto-encoder [135]	RGB-D	Cornell
		[101]	Detection accuracy: 92.58%	—	Extreme learning machine auto-encoder [136]	RGB-D	Cornell
		[102]	Image-wise split: 93.2% Object-wise split: 89.1%	—	Hybrid deep architecture; ZF model [103]	RGB-D	Cornell & THU grasp dataset

Table II. *Continued.*

Classification of methods		**Reference**	**Detection accuracy or grasping success**		**Speed**	**Algorithm or model**	**Input**	**Dataset**
Learning the structured output of grasping parameters	One-shot detection methods	[104]	Image-wise split: 88.0% Object-wise split: 87.1%		Single image: 76 ms	AlexNet [137]	RGB-D	Cornell
		[105]	Image-wise split: 88.9% Object-wise split: 88.2%		Single image: 117 ms	VGG-16 [138]	RGB-D	Cornell
		[11]	Image-wise split: 89.21% Object-wise split: 88.96%		16.03 FPS	ResNet-50 [139]; multi-modal grasp predictor [140]	RGB-D	Cornell
		[106]	Cornell	Detection accuracy: 95.2%	—	ResNet-50; primary grasp predictor [141]	RGB	Cornell Jacquard
			Jacquard	Detection accuracy: 85.74%				
			Grasping success: 92.4%					

Table II. *Continued.*

Classification of methods		Reference	Detection accuracy or grasping success		Speed	Algorithm or model	Input	Dataset
	Combining regression with classification	[109]	Single	Image-wise split: 96.0% Object-wise split: 96.1%	8.33 FPS	ResNet-50; region proposal network [142];	RGB-D	Cornell
			Multi	Grasping success: 89.0%	≤3.0 FPS			
		[110]	VGG-16	Image-wise split: 98.2% Object-wise split: 96.4%	118 FPS	VGG-16; ResNet-50; ResNet-101 [138]; oriented anchor box detection; angle matching	RGB-D	Cornell
			ResNet-50	Image-wise split: 98.8% Object-wise split: 97.0%	105 FPS			
			ResNet-101	Image-wise split: 98.8% Object-wise split: 97.8%	67 FPS			

Table II. Continued.

Classification of methods		Reference	Detection accuracy or grasping success		Speed	Algorithm or model	Input	Dataset
Learning the grasp robustness	Binary classification	[98]	Active strategy	Detection accuracy: 89.0% Grasping success: 93.0%	Generate grasp poses: 0.8∼1.7 s/ 1000 pcs; Grasp poses classification: 0.3∼6.2 s/ 1000 pcs	Geometric algorithms; force-closure analysis; classification learning based on CNN	3D point clouds	BigBird
			Passive strategy	Detection accuracy: 77.0% Grasping success: 84.0%				
		[112]	Detection accuracy: 93.5%		Single image: 1.46 s	Image edge detection; force-closure analysis; grasp recognition based on CNN	RGB	Cornell
		[113]	SVC	Detection accuracy: 98.24%	Single image: 8 ms ∼ 50 ms	Tactile signal perception; SVC [143]; KNN [144]; LR [145]	RGB	Customized dataset
			KNN	Detection accuracy: 95.0%				
			LR	Detection accuracy: 97.4%				

Table II. *Continued.*

Classification of methods		Reference	Detection accuracy or grasping success		Speed	Algorithm or model	Input	Dataset
	Others	[114]	—		Single prediction: 1.05 ms (deep learning) and 0.21 ms (Random Forests)	Monte Carlo sampling; dupervised learning based on deep learning and random forests	3D point clouds	Dex-Net 1.0
		[116]	Dex-Net 2.0	Detection accuracy: 93.7%	—	GQ-CNN	3D point clouds	Dex-Net 2.0 Cornell
			Cornell	Detection accuracy: 93.0%				
Learning the grasp robustness	Others	[117]	Detection accuracy: 96.7% Robust probability: 61.7%		Single image: 24 ms	GQ-STN; GQ-CNN	RGB-D	Dex-Net 2.0
		[119]	Sweeping	Grasping success: 71.1%	—	TOG-Net; self-supervised learning	RGB-D	Dex-Net 1.0 MPI [18]
			Hammering	Grasping success: 80.0%				

Table II. Continued.

Classification of methods	Reference	Detection accuracy or grasping success		Speed	Algorithm or model	Input	Dataset
End-to-end grasping based on visual motion control strategy	[124]	Detection accuracy: 88.6% Grasping success: 96.7%		Single image: $N \times 0.05$ s	Two-stream CNN	RGB-D	Customized dataset
	[126]	Detection accuracy: 93.3% Grasping success: 85.0%		—	GP Net; PointNet++	3D point clouds	Customized dataset
	[128]	ShapeNetPart	Grasping success: 93.0%	Single prediction: 0.365 s	L2G; DeCo feature encoder; self-supervised learning	3D point clouds	ShapeNet-Part YCB [146]
		YCB-8	Grasping success: 53.4%				
		YCB-76	Grasping success: 43.9%				

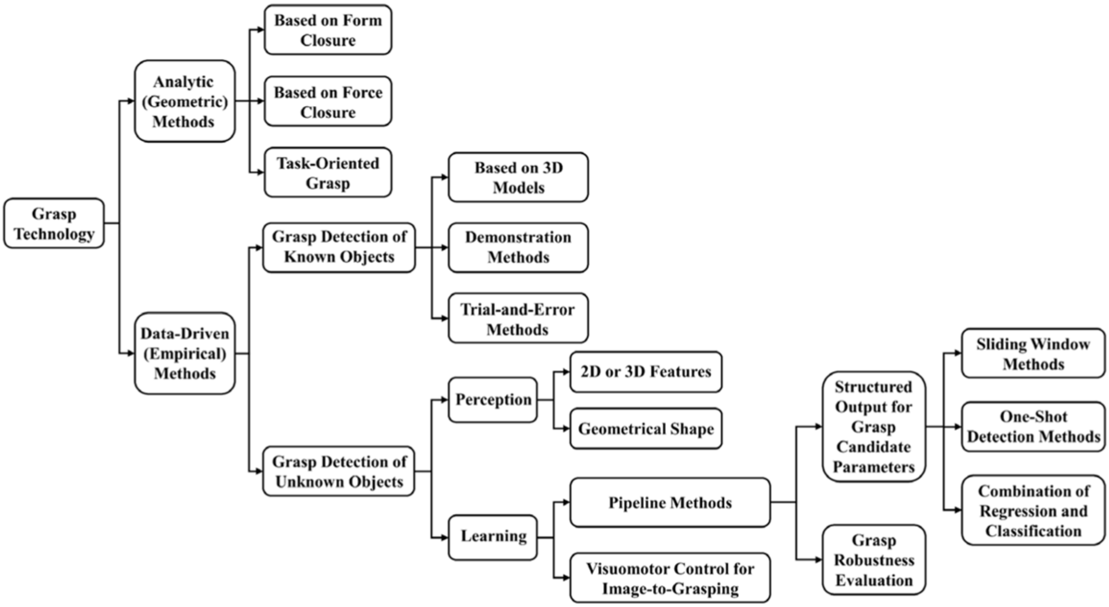
Grasp Technology
Analytic (Geometric) Methods
Based on Form Closure
Based on Force Closure
Task-Oriented Grasp
Data-Driven (Empirical) Methods
Grasp Detection of Known Objects
Based on 3D Models
Demonstration Methods
Trial-and-Error Methods
Grasp Detection of Unknown Objects
Perception
2D or 3D Features
Geometrical Shape
Learning
Pipeline Methods
Visuomotor Control for Image-to-Grasping
Structured Output for Grasp Candidate Parameters
Grasp Robustness Evaluation
Sliding Window Methods
One-Shot Detection Methods
Combination of Regression and Classification

Figure 27. *The classification method of robotic grasp detection technology proposed in this review.*

Wu et al. [126] pointed out the shortcomings of heuristic sampling grasp strategies and proposed an end-to-end Grasp Proposal Network (GP Net) for predicting 6-DOF grasps of unknown objects from monocular cameras. With point cloud data as input and PointNet++ [91] as feature encoder, the network constructs the grasp proposal set by connecting the defined grasp anchor with the point cloud successively and then trains the grasp proposal set in terms of antipodal validity [127], regress grasp prediction, and score grasp confidence, and finally an ideal result is obtained. Alliegro et al. [128] referred to the working mechanism of GP Net and proposed a more efficient end-to-end learning strategy L2G, which is used for the 6-DOF grasp of local target point clouds. The method uses a differentiable sampling strategy to identify visible contact points and a feature encoder [129] that combines local and global cues for encoding and then generates a grasp set by optimizing contact point sampling, grasp regression, and grasp classification. They used a self-supervised learning method on the ShapeNetPart dataset [130] to train the generated grasp set, and a grasping success rate of 93.0% with a prediction time of around 0.365 s for a single grasp is achieved. L2G is slightly higher than GP Net in grasping success rate but takes much less time than GP Net and has a certain generalization ability. It is more suitable for large and diversified grasp datasets.

After the previous introduction, we can find that early studies [94, 95] mainly focused on 2D image data and used relevant learning algorithms to detect the grasp points from 2D images to estimate the grasp pose. In this paper, these methods are called 2D grasp detection methods, which usually use ordinary RGB cameras to capture images and do not require additional sensing equipment, and the data acquisition is inexpensive and convenient. In addition, 2D grasp detection has low algorithm complexity and wide application. Due to its early appearance, many research results have been produced [106, 112, 113]. However, 2D grasp detection cannot directly obtain the depth information of objects and is more sensitive to the change of illumination and viewing angle. In complex environments (such as occlusion and overlap), the detection effect is usually not ideal. In contrast, 3D data has significantly improved this problem [131]. With the advent of low-cost RGB-D sensors and structured light cameras, using RGB-D or 3D point cloud data in robotic grasp has become more common [11, 101, 102, 104, 105, 114, 128]. Although 3D grasp detection has higher detection accuracy and is more robust in occlusion or overlapping occasions, it also has the characteristics of high computational complexity and cost.

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

Therefore, we need to choose the appropriate method according to the specific application requirements and environments.

Table II summarizes the mentioned methods, including the detection accuracy rate and grasping success rate of the methods, as well as the training datasets and related algorithms used, for the convenience of readers.

5. Conclusion

According to the previous description, current robotic grasp detection techniques can be classified according to Fig. 27. It can be mainly divided into analytic and data-driven methods. Among them, the analytic methods were widely used in early research, but considering the defects of this kind of method (see Section 3.3), it was gradually replaced by the data-driven method.

The data-driven methods can be applied to the grasp detection for known and unknown objects. The grasp detection methods for known objects must require the target object to have a complete 3D model. This kind of method has *a* simple principle, high detection accuracy, and easy implementation, but it has significant limitations in the complex unstructured environment. Therefore, the grasp detection methods for unknown objects have become a current research hotspot.

The classification of unknown object grasp detection techniques and related methods are shown in Table II, which can be mainly classified into perception-based and learning-based methods. Perception-based methods generate and evaluate grasp candidates by recognizing structures or features in image data, which are usually only for objects of specific shapes, and the detection time and accuracy cannot be guaranteed, so the practical application is limited. Learning-based methods are the focus of current research in robotic grasping. Such methods are not limited to the complexity of the environment and the shape of the target objects and can be divided into two categories: the pipeline methods and end-to-end grasp based on visual motion control strategies. The former can be subdivided according to the learning content into two approaches: learning the structured output of grasp parameters and learning grasp robustness evaluation.

The methods of learning the structured output of grasp parameters do not consider the intermediate steps of grasping and only focuses on the grasp results. Typical strategies include sliding window methods and one-shot detection methods. The sliding window methods have the advantages of simple structure, good detection accuracy, and certain generalization ability. The main limitation is that the optimal solution is obtained by traversal search, so the efficiency is low. It has been improved by using the one-shot detection methods, which mainly adopt the ResNet-based network model and greatly reduce the detection time while ensuring higher accuracy. Although the one-shot detection methods do not require an iterative search, it still consumes a lot of training time when the network model structure is complex and needs to be trained on large datasets. Therefore, how to streamline the network structure while ensuring detection accuracy is the development trend of learning-based robot grasp detection methods. In this regard, several ideas are proposed for the general reader: using lightweight models or sparsely connected deep networks to improve detection speed; maintaining high detection accuracy by adding residual modules to the network; using network pretraining to avoid overfitting, etc.

The methods of learning grasp robustness evaluation are able to observe intermediate steps of grasp detection with high detection accuracy, but the computational process is more complex. In contrast to the one-shot detection method, this kind of method usually consumes a lot of time during detection and spends less time on post-training. Therefore, future research direction can focus on improving the speed of the detection process.

The end-to-end grasp based on visual motion control strategies, which do not require independent configuration for the planning control system, can directly realize the mapping from images to grasp actions. A more representative strategy is to apply reinforcement learning to the method without pre-labeling the dataset, but this approach will dramatically increase the training time. Therefore, further work is needed to develop more time-efficient methods.

5.1. Future work

Since this review focuses on data-driven methods, the discussion of future research directions is also based on this. In the future, robots may be used more to grasp unknown objects in unstructured environments, which is a great test for robotic grasp detection technology. In recent years, there have been significant advances in robotic grasp detection, particularly in machine learning and computer vision. However, there are still several challenges that need to be addressed to improve the performance and robustness of robotic grasp detection systems. Some of the future research directions in this field include

1. **Learning-based approaches.** One of the key research areas in robotic grasp detection is the development of learning-based methods to improve the accuracy and efficiency of grasp detection. Deep learning techniques such as CNNs and recurrent neural networks (RNNs) have shown promise in this regard, and further research is needed to explore the potential of these techniques.
2. **Processing of large datasets.** The training of grasp detection networks usually requires a large amount of manually labelled data, which is not readily available. At present, supervised learning has been used to collect data, but there are still some drawbacks. In the future, self-supervised learning will be more applied to data acquisition, and domain adaptation techniques [132, 133] may be applied to generate application-specific datasets from 3D simulations. In addition, very few researchers have autonomously created larger datasets by incorporating reinforcement learning techniques, which is also a potential research direction.
3. **Transfer learning.** By using pretrained models and transfer learning techniques, it is possible to improve the performance of the grasp detection system with limited training data.
4. **Multi-modal sensing.** Another important research direction is the integration of multiple sensing modalities such as vision, touch, and force sensing to improve the accuracy and reliability of grasp detection. Multi-modal sensing can provide additional information about the object and its properties, which can be used to improve the grasp detection algorithm.
5. **Real-time performance.** This is a critical requirement for robotic systems that interact with the environment. Future research in robotic grasp detection should focus on developing algorithms and techniques that can operate in real time and provide reliable and efficient grasp detection in unstructured environments.

Overall, future research directions in robotic grasp detection are focused on improving the accuracy, reliability, and efficiency of grasp detection systems, as well as developing algorithms and techniques that can operate in real time and in unstructured environments.

Author contributions. Author Minglun Dong was responsible for the conception, literature review, writing and typesetting of the review paper. Author Jian Zhang is responsible for the improvement and modification of the content.

Financial support. This work was supported by the National Key R&D Program of China (No.2018YFB1308400).

Competing interests. The authors declare no competing interests exist.

Ethical approval. Not applicable.

References

[1] D. Jiang, G. Li, Y. Sun, J. Hu, J. Yun and Y. Liu, "Manipulator grabbing position detection with information fusion of color image and depth image using deep learning," *J. Ambient Intell. Humaniz. Comput.* **12**(12), 10809–10822 (2021). doi: 10.1007/s12652-020-02843-w.

[2] G. Du, K. Wang, S. Lian and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review," *Artif. Intell. Rev.* **54**(3), 1677–1734 (2020). doi: 10.1007/s10462-020-09888-5.

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

[3] S. Liu, G. Tian, Y. Zhang, M. Zhang and S. Liu, "Service planning oriented efficient object search: A knowledge-based framework for home service robot," *Expert Syst. Appl.* **187**, 115853 (2022). doi: 10.1016/j.eswa.2021.115853.

[4] J. Sanchez, J. A. Corrales, B. C. Bouzgarrou and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey," *Int. J. Robot. Res.* **37**(7), 688–716 (2018). doi: 10.1177/0278364918779698.

[5] D. Morrison, P. Corke and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *Int. J. Robot. Res.* **39**(2-3), 183–201 (2019). doi: 10.1177/0278364919859066.

[6] L. Antanas, P. Moreno, M. Neumann, R. P. De Figueiredo, K. Kersting, J. Santos-Victor and L. De Raedt, "Semantic and geometric reasoning for robotic grasping: A probabilistic logic approach," *Auton. Robot.* **43**(6), 1393–1418 (2018). doi: 10.1007/s10514-018-9784-8.

[7] M. Q. Mohammed, K. L. Chung and C. S. Chyi, "Review of deep reinforcement learning-based object grasping: Techniques, open challenges, and recommendations," *IEEE Access* **8**, 178450–178481 (2020). doi: 10.1109/access.2020.3027923.

[8] H. J. Gong, S. Ling, X. Dong and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Trans. Cybern.* **43**(5), 1318–1334 (2013). doi: 10.1109/tcyb.2013.2265378.

[9] A. Zabatani, V. Surazhsky, E. Sperling, S. B. Moshe, O. Menashe, D. H. Silver, Z. Karni, A. M. Bronstein, M. M. Bronstein and R. Kimmel, "Intel® RealSenseTM SR300 coded light depth camera," *IEEE Trans. Pattern Anal. Mach. Intell.* **42**(10), 2333–2345 (2020). doi: 10.1109/tpami.2019.2915841.

[10] K. Kleeberger, R. Bormann, W. Kraus and M. F. Huber, "A survey on learning-based robotic grasping," *Curr. Robot. Rep.* **1**(4), 239–249 (2020). doi: 10.1007/s43154-020-00021-6.

[11] 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)* (2017) pp. 769–776. doi: 10.1109/iros.2017.8202237.

[12] E. Al-Gallaf, A. Allen and K. Warwick, "A survey of multi-fingered robot hands: Issues and grasping achievements," *Mechatronics* **3**(4), 465–491 (1993). doi: 10.1016/0957-4158(93)90018-w.

[13] K. B. Shimoga, "Robot grasp synthesis algorithms: A survey," *Int. J. Robot. Res.* **15**(3), 230–266 (1996). doi: 10.1177/027836499601500302.

[14] A. Sahbani, S. El-Khoury and P. Bidaud, "An overview of 3D object grasp synthesis algorithms," *Robot. Auton. Syst.* **60**(3), 326–336 (2012). doi: 10.1016/j.robot.2011.07.016.

[15] A. Bicchi and V. Kumar, "Robotic Grasping and Contact: A Review," **In:** *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, vol. **1** (2000) pp. 348–353. doi: 10.1109/robot.2000.844081.

[16] G. Du, K. Wang, S. Lian and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review," *Artif. Intell. Rev.* **54**(3), 1677–1734 (2021). doi: 10.1007/s10462-020-09888-5.

[17] S. Caldera, A. Rassau and D. Chai, "Review of deep learning methods in robotic grasp detection," *Multimodal Technol. Interact.* **2**(3), 57 (2018). doi: 10.3390/mti2030057.

[18] J. Bohg, A. Morales, T. Asfour and D. Kragic, "Data-driven grasp synthesis—A survey," *IEEE Trans. Robot.* **30**(2), 289–309 (2013). doi: 10.1109/tro.2013.2289018.

[19] A. Bicchi, "Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity," *IEEE Trans. Robot. Autom.* **16**(6), 652–662 (2000). doi: 10.1109/70.897777.

[20] J. K. Salisbury and B. Roth, "Kinematic and force analysis of articulated mechanical hands," *J. Mech. Transm. Autom. Des.* **105**(1), 35–41 (1983). doi: 10.1115/1.3267342.

[21] Y. H. Liu, "Qualitative test and force optimization of 3-D frictional form-closure grasps using linear programming," *IEEE Trans. Robot. Autom.* **15**(1), 163–173 (1999). doi: 10.1109/70.744611.

[22] D. Ding, Y. H. Liu and S. Wang, "Computing 3-D Optimal Form-Closure Grasps," **In:** *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, vol. **4** (2000) pp. 3573–3578. doi: 10.1109/robot.2000.845288.

[23] D. Prattichizzo and J. C. Trinkle, "Grasping," **In:** *Springer Handbook of Robotics* (Springer, Berlin/Heidelberg, 2008) pp. 671–700. doi: 10.1007/978-3-540-30301-5_29.

[24] B. S. Baker, S. Fortune and E. Grosse, "Stable Prehension with Three Fingers," **In:** *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing* (1985) pp. 114–120. doi: 10.1145/22145.22158.

[25] X. Markenscoff and C. H. Papadimitriou, "Optimum grip of a polygon," *Int. J. Robot. Res.* **8**(2), 17–29 (1989). doi: 10.1177/027836498900800202.

[26] V. D. Nguyen, "Constructing force-closure grasps," *Int. J. Robot. Res.* **7**(3), 3–16 (1988). doi: 10.1177/027836498800700301.

[27] J. Ponce and B. Faverjon, "On computing three-finger force-closure grasps of polygonal objects," *IEEE Trans. Robot. Autom.* **11**(6), 868–881 (1995). doi: 10.1109/70.478433.

[28] J. Cornella and R. Suárez, "On Computing Form-Closure Grasps/Fixtures for Non-Polygonal Objects (ISATP 2005)," **In:** *The 6th IEEE International Symposium on Assembly and Task Planning: From Nano to Macro Assembly and Manufacturing* (2005) pp. 138–143. doi: 10.1109/isatp.2005.1511463.

[29] B. Faverjon and J. Ponce, "On Computing Two-Finger Force-Closure Grasps of Curved 2D Objects," **In:** *Proceedings. 1991 IEEE International Conference on Robotics and Automation* (1991) pp. 424–429. doi: 10.1109/robot.1991.131614.

[30] A. Bicchi, "On the closure properties of robotic grasping," *Int. J. Robot. Res.* **14**(4), 319–334 (1995). doi: 10.1177/027836499501400402.

[31] W. S. Howard and V. Kumar, "On the stability of grasped objects," *IEEE Trans. Robot. Autom.* **12**(6), 904–917 (1996). doi: 10.1109/70.544773.

[32] J. C. Trinkle, "On the stability and instantaneous velocity of grasped frictionless objects," *IEEE Trans. Robot. Autom.* **8**(5), 560–572 (1992). doi: 10.1109/70.163781.

[33] X. Y. Zhu and J. Wang, "Synthesis of force-closure grasps on 3-D objects based on the Q distance," *IEEE Trans. Robot. Autom.* **19**(4), 669–679 (2003). doi: 10.1109/tra.2003.814499.

[34] Y. H. Liu, "Computing N-Finger Force-Closure Grasps on Polygonal Objects," **In:** *Proceedings. 1998 IEEE International Conference on Robotics and Automation*, vol. **3** (1998) pp. 2734–2739. doi: 10.1109/robot.1998.680759.

[35] M. Ciocarlie, A. Miller and P. Allen, "Grasp Analysis Using Deformable Fingers," **In:** *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2005) pp. 4122–4128. doi: 10.1109/iros.2005.1545525.

[36] C. Rosales, R. Suárez, M. Gabiccini and A. Bicchi, "On the Synthesis of Feasible and Prehensile Robotic Grasps," **In:** *2012 IEEE International Conference on Robotics and Automation* (2012) pp. 550–556. doi: 10.1109/icra.2012.6225238.

[37] P. Jia, W.li Li, G. Wang and S. Y. Li, "Optimal grasp planning for a dexterous robotic hand using the volume of a generalized force ellipsoid during accepted flattening," *Int. J. Adv. Robot. Syst.* **14**(1), 172988141668713 (2017). doi: 10.1177/1729881416687134.

[38] Q. Lin, J. W. Burdick and E. Rimon, "A stiffness-based quality measure for compliant grasps and fixtures," *IEEE Trans. Robot. Autom.* **16**(6), 675–688 (2000). doi: 10.1109/70.897779.

[39] C. Ferrari and J. Canny, "Planning Optimal Grasps," **In:** *Proceedings 1992 IEEE International Conference on Robotics and Automation* (1992) pp. 2290–2295. doi: 10.1109/robot.1992.219918.

[40] M. O. H., "Planning of grasping with multi-fingered hands based on the maximal external wrench," *J. Mech. Eng.* **45**(3), 258–262 (2009). doi: 10.3901/jme.2009.03.258.

[41] M. Cutkosky and P. Wright, "Modeling Manufacturing Grips and Correlations with the Design of Robotic Hands," **In:** *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. **3** (1986) pp. 1533–1539. doi: 10.1109/robot.1986.1087525.

[42] Z. Li and S. S. Sastry, "Task-oriented optimal grasping by multi-fingered robot hands," *IEEE J. Robot. Autom.* **4**(1), 32–44 (1988). doi: 10.1109/56.769.

[43] N. S. Pollard, "Parallel algorithms for synthesis of whole-hand grasps," *Proc. Int. Conf. Robot. Autom.* **1**, 373–378 (1997). doi: 10.1109/robot.1997.620066.

[44] C. Borst, M. Fischer and G. Hirzinger, "Grasp Planning: How to Choose a Suitable Task Wrench Space," **In:** *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04*, vol. **1** (2004) pp. 319–325.

[45] S. El-Khoury, R. de Souza and A. Billard, "On computing task-oriented grasps," *Robot. Auton. Syst.* **66**, 145–158 (2015). doi: 10.1016/j.robot.2014.11.016.

[46] Z. Deng, X. Zheng, L. Zhang and J. Zhang, "A learning framework for semantic reach-to-grasp tasks integrating machine learning and optimization," *Robot. Auton. Syst.* **108**, 140–152 (2018).

[47] M. Wiering and M. van Otterlo, "Reinforcement learning, *Adapt. Learn. Optim.* **12**(3), 729 (2012). doi: 10.1007/978-3-642-27645-3.

[48] R. Bormann, B. F. de Brito, J. Lindermayr, M. Omainska and M. Patel, "Towards Automated Order Picking Robots for Warehouses and Retail," **In:** *Lecture Notes in Computer Science* (2019) pp. 185–198. doi: 10.1007/978-3-030-34995-0_18.

[49] A. T. Miller, S. Knoop, H. I. Christensen and P. K. Allen, "Automatic Grasp Planning Using Shape Primitives," **In:** *2003 IEEE International Conference on Robotics and Automation*, vol. **2** (2003) pp. 1824–1829. doi: 10.1109/robot.2003.1241860.

[50] A. T. Miller and P. K. Allen, "Graspit!: A Versatile Simulator for Grasp Analysis," **In:** *Proc. of the ASME Dynamic Systems and Control Division*, vol. **2** (2000) pp. 1251–1258. doi: 10.1115/imece2000-2439.

[51] C. Goldfeder, P. K. Allen, C. Lackner and R. Pelossof, "Grasp Planning via Decomposition Trees," **In:** *Proceedings 2007 IEEE International Conference on Robotics and Automation* (2007) pp. 4679–4684. doi: 10.1109/robot.2007.364200.

[52] R. Pelossof, A. Miller, P. Allen and T. Jebara, "An SVM Learning Approach to Robotic Grasping," **In:** *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04*, vol. **4** (2004) pp. 3512–3518. doi: 10.1109/robot.2004.1308797.

[53] C. Cortes, V. Vapnik and S.-V. Networks, "Support-vector networks," *Mach. Learn.* **20**(3), 273–297 (1995). doi: 10.1007/bf00994018.

[54] M. Nieuwenhuisen, J. Stückler, A. Berner, R. Klein and S. Behnke, "Shape-Primitive Based Object Recognition and Grasping," **In:** *ROBOTIK 2012; 7th German Conference on Robotics* (2012) pp. 1–5.

[55] M. R. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Trans. Robot. Autom.* **5**(3), 269–279 (1989). doi: 10.1109/70.34763.

[56] H. Kjellstrom, J. Romero and D. Kragic, "Visual Recognition of Grasps for Human-to-Robot Mapping," **In:** *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2008) pp. 3192–3199. doi: 10.1109/iros.2008.4650917.

[57] T. Feix, J. Romero, H. B. Schmiedmayer, A. M. Dollar and D. Kragic, "The grasp taxonomy of human grasp types," *IEEE Trans. Hum. Mach. Syst.* **46**(1), 66–77 (2016). doi: 10.1109/thms.2015.2470657.

[58] F. Cini, V. Ortenzi, P. Corke and M. J. S. R. Controzzi, "On the choice of grasp type and location when handing over an object," *Sci. Robot.* **4**(27), (2019). doi: 10.1126/scirobotics.aau9757.

[59] S. B. Kang and K. Ikeuchi, "Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps," *IEEE Trans. Robot. Autom.* **13**(1), 81–95 (1997). doi: 10.1109/70.554349.

[60] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith and Y. Matsuoka, "Physical human interactive guidance: Identifying grasping principles from human-planned grasps," *IEEE Trans. Robot.* **28**(4), 899–910 (2012). doi: 10.1109/tro.2012.2189498.

[61] S. Ekvall and D. Kragic, "Learning and Evaluation of the Approach Vector for Automatic Grasp Generation and Planning," **In:** *Proceedings 2007 IEEE International Conference on Robotics and Automation* (2007) pp. 4715–4720. doi: 10.1109/robot.2007.364205.

[62] S. Ekvall and D. Kragic, "Receptive Field Cooccurrence Histograms for Object Detection," **In:** *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2015) pp. 84–89. doi: 10.1109/iros.2005.1545588.

[63] S. Ekvall and D. Kragic, "Grasp Recognition for Programming by Demonstration," **In:** *Proceedings of the 2005 IEEE International Conference on Robotics and Automation* (2005) pp. 748–753. doi: 10.1109/robot.2005.1570207.

[64] Y. Lin and Y. Sun, "Robot grasp planning based on demonstrated grasp strategies," *Int. J. Robot. Res.* **34**(1), 26–42 (2014). doi: 10.1177/0278364914555544.

[65] Z. Deng, G. Gao, S. Frintrop, F. Sun, C. Zhang and J. Zhang, "Attention based visual analysis for fast grasp planning with a multi-fingered robotic hand," *Front. Neurorobot.* **13**, 60 (2019). doi: 10.3389/fnbot.2019.00060.

[66] R. Detry, D. Kraft, O. Kroemer, L. Bodenhagen, J. Peters, N. Krüger and J. Piater, "Learning grasp affordance densities," *Paladyn J. Behav. Robot.* **2**(1), 1–17 (2011). doi: 10.2478/s13230-011-0012-x.

[67] K. Dehnad, "Density estimation for statistics and data analysis," *Technometrics* **29**(4), 495–495 (1987). doi: 10.1080/00401706.1987.10488295.

[68] O. B. Kroemer, R. Detry, J. Piater and J. Peters, "Combining active learning and reactive control for robot grasping," *Robot. Auton. Syst.* **58**(9), 1105–1116 (2010). doi: 10.1016/j.robot.2010.06.001.

[69] J. Kober, E. Oztop and J. Peters, "Reinforcement Learning to Adjust Robot Movements to New Situations," **In:** *Robotics: Science and Systems VI* (2010). doi: 10.15607/rss.2010.vi.005.

[70] E. Theodorou, J. Buchli and S. Schaal, "A generalized path integral control approach to reinforcement learning," *J. Mach. Learn. Res.* **11**, 3137–3181 (2010).

[71] F. Stulp, E. Theodorou, M. Kalakrishnan, P. Pastor, L. Righetti and S. Schaal, "Learning Motion Primitive Goals for Robust Manipulation," **In:** *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2011) pp. 325–331. doi: 10.1109/iros.2011.6094877.

[72] F. Stulp, E. Theodorou, J. Buchli and S. Schaal, "Learning to Grasp under Uncertainty," **In:** *2011 IEEE International Conference on Robotics and Automation* (2011) pp. 5703–5708. doi: 10.1109/icra.2011.5979644.

[73] C. Dune, E. Marchand, C. Collowet and C. Leroux, "Active Rough Shape Estimation of Unknown Objects," **In:** *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2008) pp. 3622–3627. doi: 10.1109/iros.2008.4651005.

[74] R. Detry, C. H. Ek, M. Madry, J. Piater and D. Kragic, "Generalizing Grasps Across Partly Similar Objects," **In:** *2012 IEEE International Conference on Robotics and Automation* (2012) pp. 3791–3797. doi: 10.1109/icra.2012.6224992.

[75] K. Hsiao, S. Chitta, M. Ciocarlie and E. G. Jones, "Contact-Reactive Grasping of Objects with Partial Shape Information," **In:** *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010) pp. 1228–1235.

[76] Z. Liu, L. B. Gueta and J. Ota, "Feature Extraction from Partial Shape Information for Fast Grasping of Unknown Objects," **In:** *2011 IEEE International Conference on Robotics and Biomimetics* (2011) pp. 1332–1337. doi: 10.1109/robio.2011.6181473.

[77] A. Morales, P. J. Sanz, A. P. Del Pobil and A. H. Fagg, "Vision-based three-finger grasp synthesis constrained by hand geometry," *Robot. Auton. Syst.* **54**(6), 496–512 (2006). doi: 10.1016/j.robot.2006.01.002.

[78] M. Richtsfeld and M. Vincze, "Grasping of Unknown Objects from a Table Top," **In:** *Workshop on Vision in Action: Efficient Strategies for Cognitive Agents in Complex Environments* (2008).

[79] M. A. Fischler and R. C. B. Bolles, "Random sample consensus," *Commun. ACM* **24**(6), 381–395 (1981). doi: 10.1145/358669.358692.

[80] J. O'Rourke, *Computational Geometry in C* (Cambridge University Press, Cambridge, 1998).

[81] J. Bohg, M. Johnson-Roberson, B. León, J. Felip, X. Gratal, N. Bergström, D. Kragic and A. Morales, "Mind the Gap - Robotic Grasping under Incomplete Observation," **In:** *2011 IEEE International Conference on Robotics and Automation* (2011) pp. 686–693. doi: 10.1109/icra.2011.5980354.

[82] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science* **313**(5786), 504–507 (2006). doi: 10.1126/science.1127647.

[83] L. Bo, X. Ren and D. Fox, "Unsupervised Feature Learning for RGB-D Based Object Recognition," **In:** *Experimental Robotics* (2013) pp. 387–402. doi: 10.1007/978-3-319-00065-7_27.

[84] R. Socher, B. Huval, B. Bath, C. D. Manning and A. Ng, "Convolutional-Recursive Deep Learning for 3D Object Classification," **In:** *Advances in Neural Information Processing Systems*, vol. **25** (2012).

[85] M. Zhou, Y. Bai, W. Zhang, T. Zhao and T. Mei, "Look-into-Object: Self-Supervised Structure Modeling for Object Recognition," **In:** *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020) pp. 11774–11783. doi: 10.1109/cvpr42600.2020.01179.

[86] W. Cai, D. Liu, X. Ning, C. Wang and G. Xie, "Voxel-based three-view hybrid parallel network for 3D object classification," *Displays* **69**, 102076 (2021). doi: 10.1016/j.displa.2021.102076.

[87] I. Lenz, H. Lee and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Robot. Res.* **34**(4-5), 705–724 (2015). doi: 10.1177/0278364914549607.

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

[88] J. Zhang, W. Zhang, R. Song, L. Ma and Y. Li, "Grasp for Stacking via Deep Reinforcement Learning," **In:** *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020) pp. 2543–2549. doi: 10.1109/icra40945.2020.9197508.

[89] D. Quillen, E. Jang, O. Nachum, C. Finn, J. Ibarz and S. Levine, "Deep Reinforcement Learning for Vision-Based Robotic Grasping: A Simulated Comparative Evaluation of Off-Policy Methods," **In:** *2018 IEEE International Conference on Robotics and Automation (ICRA)* (2018) pp. 6284–6291. doi: 10.1109/icra.2018.8461039.

[90] O. M. Pedersen, E. Misimi and F. Chaumette, "Grasping Unknown Objects by Coupling Deep Reinforcement Learning, Generative Adversarial Networks, and Visual Servoing," **In:** *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020) pp. 5655–5662. doi: 10.1109/icra40945.2020.9197196.

[91] C. R. Qi, L. Yi, H. Su and L. J. Guibas, "Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," **In:** *Advances in Neural Information Processing Systems*, vol. **30** (2017).

[92] L. Jiao and J. Zhao, "A survey on the new generation of deep learning in image processing," *IEEE Access* **7**, 172231–172263 (2019). doi: 10.1109/access.2019.2956508.

[93] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *Int. J. Robot. Res.* **37**(4-5), 421–436 (2017). doi: 10.1177/0278364917710318.

[94] A. Saxena, J. Driemeyer, J. Kearns and A. Y. Ng, "Robotic Grasping of Novel Objects," **In:** *Advances in Neural Information Processing Systems*, vol. **19** (2006). doi: 10.7551/mitpress/7503.003.0156.

[95] A. Saxena, J. Driemeyer and A. Y. Ng, "Robotic grasping of novel objects using vision," *Int. J. Robot. Res.* **27**(2), 157–173 (2008). doi: 10.1177/0278364907087172.

[96] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang and A. Y. Ng, "Grasping Novel Objects with Depth Segmentation," **In:** *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2010) pp. 2578–2585. doi: 10.1109/iros.2010.5650493.

[97] 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* (2011) pp. 3304–3311.

[98] A. Ten Pas, M. Gualtieri, K. Saenko and R. Platt, "Grasp pose detection in point clouds," *Int. J. Robot. Res.* **36**(13-14), 1455–1473 (2017). doi: 10.1177/0278364917735594.

[99] M. Gualtieri, A. ten Pas, K. Saenko and R. Platt, "High Precision Grasp Pose Detection in Dense Clutter," **In:** *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2016) pp. 598–605. doi: 10.1109/iros.2016.7759114.

[100] A. Ten Pas and R. Platt, "Using Geometry to Detect Grasp Poses in 3D Point Clouds," **In:** *Springer Proceedings in Advanced Robotics* (2017) pp. 307–324. doi: 10.1007/978-3-319-51532-8_19.

[101] J. Wei, H. Liu, G. Yan and F. Sun, "Robotic grasping recognition using multi-modal deep extreme learning machine," *Multidim. Syst. Signal Process.* **28**(3), 817–833 (2016). doi: 10.1007/s11045-016-0389-0.

[102] 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)* (2017) pp. 1609–1614. doi: 10.1109/icra.2017.7989191.

[103] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," **In:** *Computer Vision – ECCV 2014* (2014) pp. 818–833. doi: 10.1007/978-3-319-10590-1_53.

[104] J. Redmon and A. Angelova, "Real-Time Grasp Detection Using Convolutional Neural Networks," **In:** *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015) pp. 1316–1322. doi: 10.1109/icra.2015.7139361.

[105] Q. Zhang, D. Qu, F. Xu and F. Zou, "Robust robot grasp detection in multimodal fusion," *MATEC Web of Conf.* **139**, 00060 (2017). doi: 10.1051/matecconf/201713900060.

[106] A. Depierre, E. Dellandréa and L. Chen, "Scoring Graspability Based on Grasp Regression for Better Grasp Prediction," **In:** *2021 IEEE International Conference on Robotics and Automation (ICRA)* (2021) pp. 4370–4376.

[107] M. Basalla, F. Ebert, R. Tebner and W. Ke, Grasping for the Real World (Greifen mit Deep Learning) (2017). https://www.frederikebert.de/abgeschlossene-projekte/greifen-mit-deep-learning

[108] J. Watson, J. Hughes and F. Iida, "Real-World, Real-Time Robotic Grasping with Convolutional Neural Networks," **In:** *Towards Autonomous Robotic Systems* (2017) pp. 617–626. doi: 10.1007/978-3-319-64107-2_50.

[109] F-J. Chu, R. Xu and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robot. Autom. Lett.* **3**(4), 3355–3362 (2018). doi: 10.1109/lra.2018.2852777.

[110] H. Zhang, X. Zhou, X. Lan, J. Li, Z. Tian and N. Zheng, "A real-time robotic grasping approach with oriented anchor box," *IEEE Trans. Syst. Man Cybern. Syst.* **51**(5), 3014–3025 (2019). doi: 10.1109/tsmc.2019.2917034.

[111] N. Anđelić, Z. Car and M. Šercer, "Prediction of robot grasp robustness using artificial intelligence algorithms," *Tehnicki Vjesnik - Technical Gazette* **29**(1), 101–107 (2022). doi: 10.17559/tv-20210204092154.

[112] L. Chen, P. Huang, Y. Li and Z. Meng, "Edge-dependent efficient grasp rectangle search in robotic grasp detection," *IEEE/ASME Trans. Mechatron.* **26**(6), 2922–2931 (2020). doi: 10.1109/tmech.2020.3048441.

[113] T. Li, X. Sun, X. Shu, C. Wang, Y. Wang, G. Chen and N. Xue, "Robot grasping system and grasp stability prediction based on flexible tactile sensor array," *Machines* **9**(6), 119 (2021). doi: 10.3390/machines9060119.

[114] D. Seita, F. T. Pokorny, J. Mahler, D. Kragic, M. Franklin, J. Canny and K. Goldberg, "Large-Scale Supervised Learning of the Grasp Robustness of Surface Patch Pairs," **In:** *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)* (2016) pp. 216–223. doi: 10.1109/simpar.2016.7862399.

[115] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kroger, J. Kuffner and K. Goldberg, "Dex-NET 1.0: A Cloud-Based Network of 3D Objects for Robust Grasp Planning Using a Multi-Armed Bandit Model with

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press

Correlated Rewards," **In:** *2016 IEEE International Conference on Robotics and Automation (ICRA)* (2016) pp. 1957–1964. doi: 10.1109/icra.2016.7487342.

[116] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Aparicio 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 XIII* (2017). doi: 10.15607/rss.2017.xiii.058.

[117] A. Gariépy, J. C. Ruel, B. Chaib-Draa and P. Giguere, "GQ-STN: Optimizing One-Shot Grasp Detection Based on Robustness Classifier," **In:** *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2019) pp. 3996–4003. doi: 10.1109/iros40897.2019.8967785.

[118] M. Jaderberg, K. Simonyan and A. Zisserman, "Spatial Transformer Networks," **In:** *Advances in Neural Information Processing Systems*, vol. **28** (2015).

[119] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei and S. Savarese, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," *Int. J. Robot. Res.* **39**(2-3), 202–216 (2020). doi: 10.1177/0278364919872545.

[120] S. B. Šegota, N. Anđelić, C. A. R. Z. and M. Šercer, "Prediction of robot grasp robustness using artificial intelligence algorithms," *Technical Gazette* **29**(1), 101–107 (2022). doi: 10.17559/tv-20210204092154.

[121] Kaggle, "Grasping Dataset," **In:** *Ugocupcic* (2017). https://www.kaggle.com/datasets/ugocupcic/grasping-dataset

[122] F. Zhang, J. Leitner, M. Milford, B. Upcroft and P. Corke, "Towards vision-based deep reinforcement learning for robotic motion control," *ArXiv Preprint* (2015).

[123] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature* **518**(7540), 529–533 (2015). doi: 10.1038/nature14236.

[124] A. Zeng, S. Song, K-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. Chavan Dafle, R. Holladay, I. Morona, P. Q. Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser and A. Rodriguez, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *Int. J. Robot. Res.* **41**(7), 690–705 (2022). doi: 10.1177/0278364919868017.

[125] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz and S. Levine, "End-to-end learning of semantic grasping," *ArXiv Preprint* (2017).

[126] C. Wu, J. Chen, Q. Cao, J. Zhang, Y. Tai, L. Sun and K. Jia, "Grasp proposal networks: An end-to-end solution for visual learning of robotic grasps," *Adv. Neural Inf. Process. Syst.* **33**, 13174–13184 (2020).

[127] I. M. Chen and J. W. Burdick, "Finding antipodal point grasps on irregularly shaped objects," *IEEE Trans. Robot. Autom.* **9**(4), 507–512 (1993). doi: 10.1109/70.246063.

[128] A. Alliegro, M. Rudorfer, F. Frattin, A. Leonardis and T. Tommasi, "End-to-end learning to grasp via sampling from object point clouds," *IEEE Robot. Autom. Lett.* **7**(4), 9865–9872 (2022). doi: 10.1109/lra.2022.3191183.

[129] A. Alliegro, D. Valsesia, G. Fracastoro, E. Magli and T. Tommasi, "Denoise and Contrast for Category Agnostic Shape Completion," **In:** *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021) pp. 4629–4638. doi: 10.1109/cvpr46437.2021.00460.

[130] L. Yi, V. G. Kim, D. Ceylan, I-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer and L. Guibas, "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.* **35**(6), 1–12 (2016). doi: 10.1145/2980179.2980238.

[131] A. Saxena, L. L. Wong and A. Y. Ng, "Learning grasp strategies with partial shape information," *AAAI* **3**(2), 1491–1494 (2008).

[132] A. Farahani, S. Voghoei, K. Rasheed and H. R. Arabnia, "A Brief Review of Domain Adaptation," **In:** *Advances in Data Science and Information Engineering* (2021) pp. 877–894. doi: 10.1007/978-3-030-71704-9_65.

[133] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing* **312**, 135–153 (2018). doi: 10.1016/j.neucom.2018.05.083.

[134] B. Wang, L. Jiang, J. W. Li, H. G. Cai and H. Liu, "Grasping Unknown Objects Based on 3D Model Reconstruction," **In:** *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (2005) pp. 461–466. doi: 10.1109/aim.2005.1511025.

[135] I. Goodfellow, H. Lee, Q. Le, A. Saxe and A. Ng, "Measuring Invariances in Deep Networks," **In:** *Advances in Neural Information Processing Systems*, vol. **22** (2009).

[136] E. Cambria and G. Huang, "Extreme learning machines-representational learning with ELMs for big data," *IEEE Intell. Syst.* **28**(6), 30–59 (2013).

[137] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM* **60**(6), 84–90 (2017). doi: 10.1145/3065386.

[138] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv Preprint* (2014).

[139] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," **In:** *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016) pp. 770–778. doi: 10.1109/cvpr.2016.90.

[140] M. Schwarz, H. Schulz and S. Behnke, "RGB-D Object Recognition and Pose Estimation Based on Pre-Trained Convolutional Neural Network Features," **In:** *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015) pp. 1329–1335. doi: 10.1109/icra.2015.7139363.

[141] 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)* (2018) pp. 7223–7230. doi: 10.1109/iros.2018.8594116.

[142] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(6), 1137–1149 (2017). doi: 10.1109/tpami.2016.2577031.

[143] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," *IEEE Intell. Syst. Appl.* **13**(4), 18–28 (1998). doi: 10.1109/5254.708428.

[144] K. Fukunaga and P. M. Narendra, "A branch and bound algorithm for computing K-nearest neighbors," *IEEE Trans. Comput.* **100**(7), 750–753 (1975). doi: 10.1109/t-c.1975.224297.

[145] I. Ruczinski, C. Kooperberg and M. LeBlanc, "Logic regression," *J. Comput. Graph. Stat.* **12**(3), 475–511 (2003). doi: 10.1198/1061860032238.

[146] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel and A. M. Dollar, "Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set," *IEEE Robot. Autom. Mag.* **22**(3), 36–52 (2015). doi: 10.1109/mra.2015.2448951.

Cite this article: M. Dong and J. Zhang (2023). "A review of robotic grasp detection technology", Robotica **41**, 3846–3885. https://doi.org/10.1017/S0263574723001285

https://doi.org/10.1017/S0263574723001285 Published online by Cambridge University Press