robotics

MDPI

Article

Grasping Complex-Shaped and Thin Objects Using a Generative Grasping Convolutional Neural Network †

Jaeseok Kim, Olivia Nocentini * and Muhammad Zain Bashir and Filippo Cavallo

Department of Industrial Engineering, University of Florence, 50139 Florence, Italy

* Correspondence: olivia.nocentini@unifi.it

† This paper is an extended version of our paper published in Olivia, N.; Jaeseok, K.; Zain, M. B.; et al. An architecture for grasping complex-shaped, thin and reflective objects. In Proceedings of the Second Italian Conference on Robotics and Intelligent Machines, Rome, Italy, 10–12 December 2020; DOI:10.5281/zenodo.4781336.

Abstract: Vision-based pose detection and grasping complex-shaped and thin objects are challenging tasks. We propose an architecture that integrates the Generative Grasping Convolutional Neural Network (GG-CNN) with depth recognition to identify a suitable grasp pose. First, we construct a training dataset with data augmentation to train a GG-CNN with only RGB images. Then, we extract a segment of the tool using a color segmentation method and use it to calculate an average depth. Additionally, we apply and evaluate different encoder–decoder models with a GG-CNN structure using the Intersection Over Union (IOU). Finally, we validate the proposed architecture by performing real-world grasping and pick-and-place experiments. Our framework achieves a success rate of over 85.6% for picking and placing seen surgical tools and 90% for unseen surgical tools. We collected a dataset of surgical tools and validated their pick and place with different GG-CNN architectures. In the future, we aim to expand the dataset of surgical tools and improve the accuracy of the GG-CNN.

Keywords: generative grasping convolutional neural networks; surgical tools

Citation: Kim, J.; Nocentini, O.; Bashir, M.Z.; Cavallo, F. Grasping Complex-Shaped and Thin Objects Using a Generative Grasping Convolutional Neural Network. *Robotics* **2023**, *12*, 41. https://doi.org/10.3390/robotics12020041

Academic Editor: Luigi Fortuna

Received: 16 January 2023
Revised: 6 March 2023
Accepted: 9 March 2023
Published: 15 March 2023

1. Introduction

Grasping has been a long and extensively studied area in Robotics. In an era of automation, robotic grasping allows for the quick, reliable, efficient, and reproducible handling of objects in many environments. While humans beings are able to grasp various kinds of objects under various environmental conditions, replicating the same skills in robots is not an easy task. The first step in a robotic grasping process is to detect an object's pose, which is still challenging. Although active research by the computer vision and machine learning community has provided solutions to this problem, numerous challenges stem from the object's properties of shape, appearance, color, occlusions, etc. An object such as a surgical tool is difficult to detect using traditional vision approaches. The tool has a complex shape, a reflective surface, and thinness, making it tough to infer its grasp pose using traditional vision-based methods confidently. Consequently, existing works have focused on machine learning and deep learning approaches to propose the grasp poses of such objects. In this regard [1,2] proposed a novel neural network architecture called the Generative Grasping Convolutional Neural Network to select grasp points and to predict the quality and pose of grasps. Although this method has been able to propose good grasp candidates for adversarial and household objects, it fails to address the problem of the thinness associated with surgical tools, and as a result, it performs poorly on such objects. We have addressed this issue by training a GG-CNN2 with only RGB images instead of only depth images as used by [2]. Our network's output is still a grasp box in pixel coordinates, and its center can be taken as the grasp point. Moreover, the depth estimation of the grasp point is important for the grasping task. A color segmentation method with average

depth calculation could provide the depth information between a camera and a tool in our experimental setup. We integrated the Generative Grasping Convolutional Neural Network with the segmentation method that could generate a reliable grasping point and experimented with it in an actual environment (Figure 1).

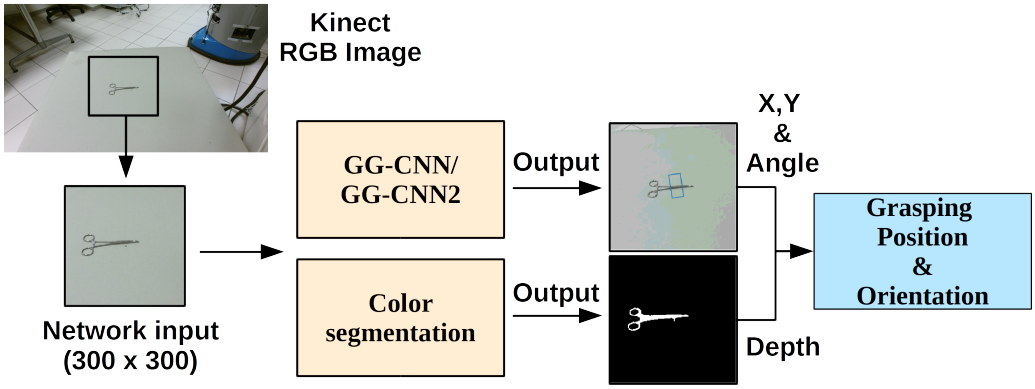

Figure 1. Overview of our proposed architecture: From the Kinect, an RGB image is cropped by the network to obtain a 300 × 300 image as input to the GG-CNN/GG-CNN2 network for tool detection, grasping position (x and y), and orientating calculation. The cropped image is also used as input to the color segmentation method to obtain tool segmentation, and the segmentation is used to calculate average depth. We used the OpenCV library to obtain the color segmentation using HSV method to obtain a black and white image where the surgical tool is white and the rest of the image is black. The average depth of objects calculated by the white image matched in the depth image.

Concretely, the contributions of our work are the following:

- We created a new dataset of surgical tools composed of only complex-shaped and thin objects that are usually more difficult to grasp due to the challenging depth estimation resulting from their thinness;
- We proposed an architecture for grasping complex-shaped and thin objects, such as surgical tools, using the GG-CNN/GG-CNN2 with a segmentation method that provides the depth of the surgical tools images;
- We compared the performance of the GG-CNN model with the dataset by applying different encoder–decoder models with GG-CNN/GGCNN2 structures and evaluating the models with the IOU.
- We conducted preliminary experiment tests for validating the GG-CNN architecture for grasping the surgical tools of seen and unseen in noncluttered or cluttered environments.

This paper is organized as follows: Section 2 presents the related works. Section 3 describes the surgical tools dataset and the GG-CNN. Section 4 introduces the description of the experimental setup. Section 5 discusses the experiments, and finally, Section 6 concludes the paper.

2. Related Works

The grasp synthesis problem has been studied for decades and it can be divided into empirical and analytical methods [3,4]. Empirical methods [5–7] focus on using models and experience-based approaches. These techniques work with known items, associating good grasp points with an offline database of object models or shapes [8–10], or familiar items, based on object classes [11] or object parts [12], but they are unable to generalize to new objects. Analytic methods [13,14], instead, use mathematical and physical models of geometry, kinematics, and dynamics to calculate grasps that are stable [15,16]. However, they tend to not transfer well to the real world due to the difficulty in modeling physical interactions between a manipulator and an object [4,15,17].

Analytical metrics have been rigorously tested by comparing their predictions to the outcomes of real-world experiments [18,19], and the findings of these studies indicate that analytical metrics are not very good predictors of grasp success in the real world. This is why we decided to determine the optimal grasping area by evaluating the grasping quality based on the geometric position relationship. Another reason to avoid using analytical models is that they are costly in terms of computational resources [4].

In the past [10,11], most works used human-designed features to represent grasps in images or required the full 3D model of objects to generate grasps [20,21]. These methods were very popular, but they all faced the challenges of non-robust features or lacked the full 3D models when used in real-world applications. Moreover, these methodologies depended to a great extent on the the pre-built database, not generalizing well to novel objects.

Grasp Proposals Using Deep Learning Methods

Recently, deep learning methods [22] have proved effective in robotic grasp generation [23]. Neural networks are used to extract features from images and apply them to detect grasps to be executed by robots. Several of these techniques share a common pipeline: classifying grasp candidates sampled from an image or point cloud, and then ranking them individually using CNNs [2]. However, such approaches provide two key challenges. The first is the availability of diverse and high-quality data, which is widely recognized as an important prerequisite for training robust and generalizable models. The second one provides standardized, reproducible, and comparable evaluation methods and metrics [24]. In order to accelerate the detection speed, the recent GG-CNN proposed by Morrison et al. [1] directly evaluated the grasp quality and the pose of grasps at every pixel. This architecture has a high processing speed, overcoming the limitations of the current deep learning grasping techniques by avoiding the discrete sampling of grasp candidates. Moreover, the GG-CNN is orders of magnitude smaller while detecting stable grasps with an equivalent performance to the current state-of-the-art techniques [1]. The GG-CNN was used by Yu et al. [25] to build a regression network that is part of an efficient cascaded deep learning framework which has a real-time performance and guarantees grasp robustness for robotic manipulation in an unstructured environment. Wang et al. [26] modified the GG-CNN, developing an efficient neural network model to generate robotic grasps with high-resolution images, while Mahajan et al. [27] augmented the GG-CNN architecture with a decoder structure used in the vector-quantized variational autoencoder (VQ-VAE) model with the intuition that it should help to regress in the vector-quantized latent space.

The same authors improved the GG-CNN, creating a new network for real-time grasp prediction called GG-CNN2 (Figure 2). This architecture is a fully convolutional network based on the semantic segmentation architecture from Yu and Koltun [28], which uses dilated convolutional layers to provide an improved performance in semantic segmentation tasks. The GG-CNN2 uses the same input and outputs as the GG-CNN, contains 66,000 parameters, and has an average inference time of 3 ms, with the entire grasping pipeline taking on average 20 ms. In the GG-CNN2, the authors vary three parameters: the filter sizes, number of filters, and size of the dilated convolutions. The GG-CNN2 is used in [29] as a part of the ORientation AtteNtive Grasp synthEsis (ORANGE) framework that jointly solves a bin classification problem and a real-value regression. In [30], a comparison was made between the GG-CNN2 and the performance in grasping of a network presented by the authors. They proposed a novel Generative Residual Convolutional Neural Network (GR-ConvNet) model that can generate robust antipodal grasps from n-channel input at real-time speeds, and they stated that their network had a better performance in grasping than the GG-CNN2. In [31], the authors proposed a method for centering the object of interest in the field of view using visual serving based on RGB-D image data acquired from an eye-in-hand camera. Their method was designed to reduce propagation errors and eliminate the need for complex hand tracking algorithms, image segmentation, or 3D reconstruction. Regardless of the geometric complexity and physical properties of the object in question, the proposed approach can efficiently generate reliable multi-view object

grasps. The system architecture proposed allows for simple and effective path generation as well as real-time tracking control. Furthermore, the system is modular, dependable, and precise in end-effector path generation and control. In another work [32], the authors took a different approach to grasp detection by treating it as image-space keypoint detection. Rather than a triplet or quartet of corner points, the deep network detects each grasp candidate as a pair of keypoints convertible to the grasp representation $g = \{x, y, w\}^T$. This increases the performance by reducing the detection difficulty by grouping keypoints into pairs. A non-local module is incorporated into the network design to encourage the capture of the dependencies between the keypoints. A final filtering strategy based on discrete and continuous orientation prediction eliminates false correspondences, improving the grasp detection performance even further. Finally, in [33], starting with a 3D partial view of the object, the authors proposed an end-to-end learning solution for generating 6-DOF parallel jaw grasps. Their Learning to Grasp (L2G) method extracts information from an input point cloud using a novel procedure that combines a differentiable sampling strategy to identify visible contact points with a feature encoder that takes advantage of local and global cues. Overall, L2G is guided by a multi-task objective that optimizes contact point sampling, grasp regression, and grasp classification to generate a diverse set of grasps.

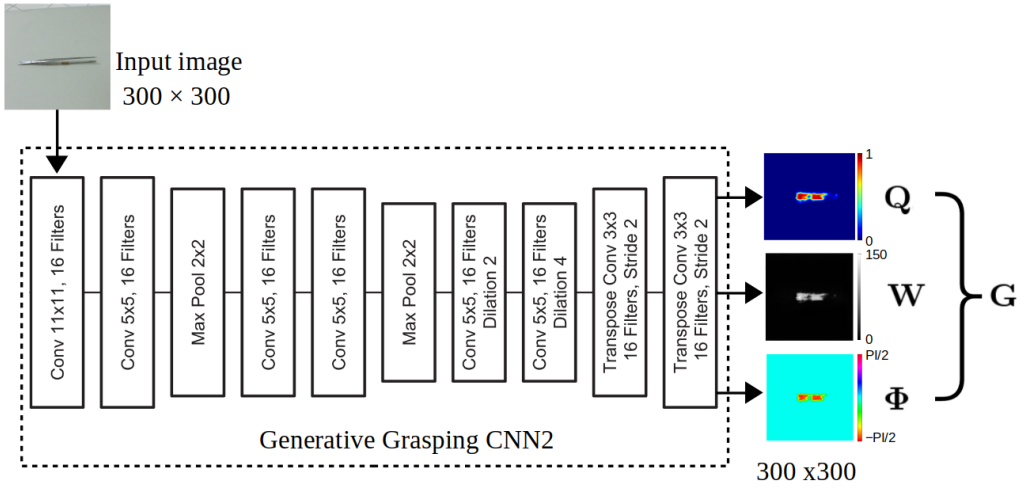

Figure 2. GG-CNN2 network architecture [2]. The GG-CNN2 directly generates a grasp pose G from grasp quality Q, grasp width W, and grasp angle Φ.

3. Methods

3.1. Grasping Generative Convolutional Neural Network

In [34–36], the grasp representation proposed by [37], and then simplified by [38], was used to generate antipodal robotic grasps using RGB-D images of objects. The grasp representation is defined by the following formula:

$$g = \{x, y, \phi, h, w\}, \tag{1}$$

g represents a five-dimension rectangle perpendicular to x-y plane that includes the center of the rectangle (x,y), the orientation of the rectangle relative to the horizontal axis of the image ϕ, its width (w), and height (h) . Morrison et al. [1] proposed a new representation of robotic grasps that changes g into this representation:

$$g = \{p, \phi, w, q\}, \tag{2}$$

where $p = (x, y, z)$ is the center position of the gripper, ϕ is the rotation angle relative to the horizontal axis of the image plane, w is the gripper width, and q is the grasp quality, which represents the chances of grasp success. Concerning the computation of q, each grasping rectangle has set the corresponding area of q to a value of 1. All other pixels are 0.

Robotic grasps are detected in the depth image $I \in \mathbb{R}^{H \times W}$ with height H and width W. In the image space I, the grasp g is represented by

$$\tilde{g} = \{s, \tilde{\phi}, \tilde{w}, q\}, \tag{3}$$

where $s = (u,v)$ represents the center point in pixels coordinates, $\tilde{\phi}$ denotes the rotation relative to the camera frame, $\tilde{w}$ denotes the gripper width in pixels, and q the grasp quality. The grasp map G proposed by Morrison et al. [1] is

$$G = \{\boldsymbol{\Phi}, \boldsymbol{W}, \boldsymbol{Q}\} \in \mathbb{R}^{3 \times H \times W} \tag{4}$$

$\boldsymbol{\Phi}$ is an image of dimension $H \times W$ which describes the angle of a grasp to be executed at each point, W an image which expresses the gripper width of a grasp to be executed at each point, q an image which defines the quality of a grasp executed at each point (u,v). $\boldsymbol{\Phi}$, $\boldsymbol{W}$, $\boldsymbol{Q}$ are each in $\mathbb{R}^{1 \times H \times W}$ and each pixel contains the values $\tilde{\phi}$, $\tilde{w}$, and q, respectively, at each pixel s. Following [1], the authors used this network to generate a grasp g for each pixel in depth image I, which denotes the pixel-wise representation:

$$M(I) = G, \tag{5}$$

where the map function M is a deep NN, and then the best grasp can be found by

$$g = \max_{Q} G. \tag{6}$$

The Mean Squared Error loss is applied to predict grasp pose, grasp quality $\tilde{q}$, gripper width $\tilde{w}$, and rotation $\tilde{\phi}$.

3.1.1. Contractive Networks

A deep contractive network imposes a layer-wise contractive penalty in a feed-forward neural network. The layer-wise penalty approximately minimizes the network outputs variance with respect to perturbations in the inputs, enabling the trained model to achieve "flatness" around the training data points [39]. The objective function of the model is the following:

$$J_{DCN}(\theta) = \sum_{i=1}^{m} L(t^{(i)}, y^{(i)}) + \sum_{j=1}^{H+1} \lambda_j \left\| \frac{\partial h_j^{(i)}}{\partial h_{j-1}^{(i)}} \right\|_2 \tag{7}$$

with outputs $y \in \mathbb{R}^{d_y}$, a target $t \in \mathbb{R}^{d_y}$, h is the hidden representation $\in \mathbb{R}^{d_y}$ with the number of hidden layers H, and λ is a scaling factor that trades off reconstruction objective with contractive objective. $\left\| \frac{\partial y^{(i)}}{\partial x^{(i)}} \right\|_2$ is the Frobenius norm of the Jacobian matrix of $h_{(j)}$ with respect to $h_{(j-1)}$.

This layer-wise contractive penalty computes partial derivatives similarly to a contractive autoencoder and is easily incorporated into the backpropagation procedure. Moreover, it is a computationally efficient way to greedily propagate input invariance through a deep network.

3.1.2. Denoising Networks

The denoising encoder–decoder model introduces some noise to the input before the model starts training. This procedure achieves good representation because corrupted input data will help obtain a robust model useful for recovering the corresponding label data. From the state of the art, the most discussed types of noise in the literature are the additive white Gaussian noise (AWGN) [40], impulse noise [41], quantization noise [42], Poisson noise [43], and speckle noise [44]. For our work, random noise with noise factor are used to produce the noise input data.

3.1.3. Sparse Networks

Sparse neural networks are artificial neural information processing systems restricted in their structure and dynamics to conserve resources. The concept of sparseness refers to the network's connectivity structure, so that each neuron receives inputs from only a limited number of other neurons, and to the network's state which describes the level of activity of the entire neural population, so that only a few neurons are active at any one time [45].

3.1.4. Variational Autoencoder (VAE) Networks

VAE is an artificial neural network architecture introduced by Kingma and Welling, belonging to the families of probabilistic graphical models and variational Bayesian methods. VAE is meant to compress the input information into a constrained multivariate latent distribution (encoding) to reconstruct it as accurately as possible (decoding). In these networks, the input data are sampled by a parametrized distribution, and the encoder and decoder are trained jointly so that the output minimizes a reconstruction error in the sense of the Kullback–Leibler divergence between the parametric posterior and the true posterior [46].

3.2. Proposed Approach

In this paper, we propose an architecture for achieving precise grasping of surgical tools, using computer vision techniques and depth segmentation. The architecture integrates the GG-CNN/GG-CNN2 models with depth segmentation to achieve high-precision grasping of surgical tools (see Figure 1). The input to our system is an RGB image of a single surgical tool on a white flat surface. The image is cropped by the network to provide a better view of the tool, and then analyzed by the GG-CNN/GG-CNN2 models. These models output the x, y, and orientation of the best grasping rectangle for the tool in the camera frame. The GG-CNN/GG-CNN2 models have been shown to achieve state-of-the-art performance in robotic grasping tasks and are well suited to our application of surgical tool grasping. In addition to the GG-CNN/GG-CNN2 models, we incorporate depth segmentation to calculate the average z value of the tool in the camera frame. This segmentation output is a depth image of the tool, which is then transformed into the robot frame. The depth segmentation provides additional information about the tool's position and orientation in 3D space, which is critical for achieving precise grasping. Once we have obtained the x, y, and z values of the tool in the robot frame, we issue a position command to the robot using open-loop position control. This command directs the robot to move to the specified position and orientation in order to grasp the tool with precision. The open-loop position control has the advantage of being simple and fast, which is important for real-time operation in surgical settings. Our proposed architecture has several advantages over existing approaches. First, it combines the strengths of the GG-CNN/GG-CNN2 models with depth segmentation to achieve highly accurate and efficient grasping of surgical tools. Second, it is based on open-loop position control, which is simple, fast, and reliable. We evaluated our proposed architecture on a dataset of surgical tools and achieved high accuracy and efficiency in grasping the tools. We evaluated our proposed architecture on a dataset of surgical tools and achieved high accuracy and efficiency in grasping the tools.

4. Experimental Setup

Our experimental setup uses a Universal Robot (UR5) equipped with a Robotiq gripper, and a Kinect v2 is used to acquire the images of the surgical tools (as shown in Figure 3a,b). The vision sensor is mounted at the top of a tripod, as shown in Figure 3c. We trained the surgical dataset on a PC running Ubuntu 16.04 LTS and using a GTX-1080ti GPU. The code was predominantly written in Python. For what regards the motion planning of the manipulator, we used the Moveit tool [47].

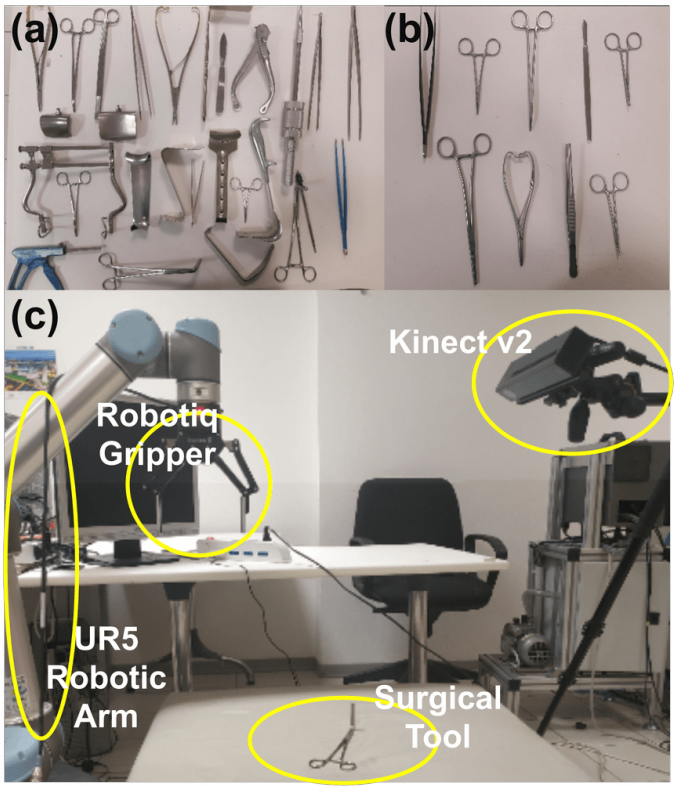

Figure 3. The experimental setup with surgical tools is presented. In (**a**), the different shapes of 27 surgical tools for training and testing (seen objects) are shown. In (**b**), the 9 new surgical tools for validation (unseen objects) are displayed, and in (**c**), the experimental setup is provided. The experimental setup is composed of a UR5 manipulator, a Robotiq gripper, a vision sensor, and a surgical tool.

4.1. Datasets

Because our dataset size is very small compared to datasets such as the Cornell dataset and the Jacquard one, we augmented it using random rotations, zooming, jitter and white noises, and brightness variations. We chose to augment the dataset using random rotations, zooming, jitter and white noises, and brightness variations, because they are more "natural" ways to augment the dataset and because they represent possible variations in how the object can be found in reality. For example, the rotation augmentation is useful because the surgical tool can have different angular positions when it is placed on the table. Moreover, brightness is useful when the object is placed in different environments with several light conditions. Zooming augmentation is also important because the object can be grasped by the manipulator from different angles, and noises are also necessary to simulate different environment conditions.

In the end, we obtained a dataset of 9920 images (the original dataset consists of 320 RGB images). The dataset is composed of images showing a single surgical tool placed on a table-top scene with white background. We decided to have different categories in the training set and validation set to stress the network more and see if it can successfully grasp the new surgical tools even if it has not seen them before. Moreover, the figures are manually labeled with grasping rectangles. The augmented dataset is publicly available (the dataset was released on github: https://github.com/OliviaNocentini/SurgicalKit (accessed on 15 January 2023)).

Then, we used a 80/20% split for training/testing of the GG-CNN2. To determine the best network configuration, we trained the network by changing some parameters (Table 1).

Table 1. Internal parameters of the GG-CNN2.

Batch size	8
Epochs	100
Batches per epochs	1000
Val-batches	250
Optimizer	Adam
Learning rate	0.001
Loss function	MSE

4.2. Experimental Method

During the experimental phase, we tested the grasping performance not only of the GG-CNN2 but also of the GG-CNN to compare the two networks. From the training of the GG-CCN and GG-CNN2, we obtained several models based on different combinations of augmentation data. We used the best models to predict the grasp poses of each surgical tool with the GG-CNN and GG-CNN2, with fixed depth or segmentation. For each configuration, we grasped each surgical tool 10 times and we used seen (used for training) and unseen objects (27 and 9, respectively) to evaluate quantitatively the grasping of the surgical tools using the GG-CNN2 with fixed-depth or with segmentation method (Figure 3a,b).

5. Experimental Results and Discussions

5.1. Quantitative Results

5.1.1. Network Evaluation

We evaluated the Intersection Over Union (IOU) with the GG-CNN/GG-CNN2 during the training. In this paper, the IOU refers to the Intersection Over Union between the ground-truth grasping rectangles, obtained by manually labeling all the trained images, and the predicted grasping rectangles. With regard to the ground-truth grasping rectangles, we annotated a txt file with 4 points for each grasping rectangle. These points represent the vertices of the grasping rectangle and were annotated following the reference paper [1]. The points are used to derive four pieces of information (position, sine and cosine angles, and width) that are used for calculating the loss.

The IOU is described with the following equation:

$$\text{IOU} = \frac{|A \cap B|}{|A \cup B|} \tag{8}$$

where A and B are, respectively, the real and ground-truth rectangles. A success is counted if the grasp rectangle has a 25% IOU with a ground truth and is within 30 degrees [38]. In Figure 4, the ground-truth boxes (in blue) and the real grasping rectangle (in red) are shown. As can be seen, the real grasping rectangle is near the ground-truth rectangle where there is a high q value (yellow color in the center of the right grasping spot light).

Table 2 describes the comparison of the IOU with different encoder–decoder models and GGCNN structures. Normally, the GG-CNN2 outperforms the GG-CNN with different models. Moreover, except for the performance of the VAE-GG-CNN/GG-CNN2 and Denoising-GGCNN, the average IOU of the models is higher than 80%. The original GG-CNN/GG-CNN2 already achieved a good evaluation over approximately 80% (with an MSE loss function), but other models conduct a greater performance. The reason is that contractive GG-CNN/GG-CNN2 models applied different loss functions with penalty terms. It is a kind of regularization technique that could extract useful features. In contrast, the performance of the VAE-GG-CNN/GG-CNN2 is dramatically low compared to the other models. Especially, the output from the encoder of the VAE is composed of latent distribution, which is μ, and covariance as the input to the decoder. It helps to train the

model and generate new data, but the latent distribution missed many of the features from the new dataset during the training. Moreover, we used three MSE loss functions with a GG-CNN structure that is not the same and the original VAE structure was not optimized.

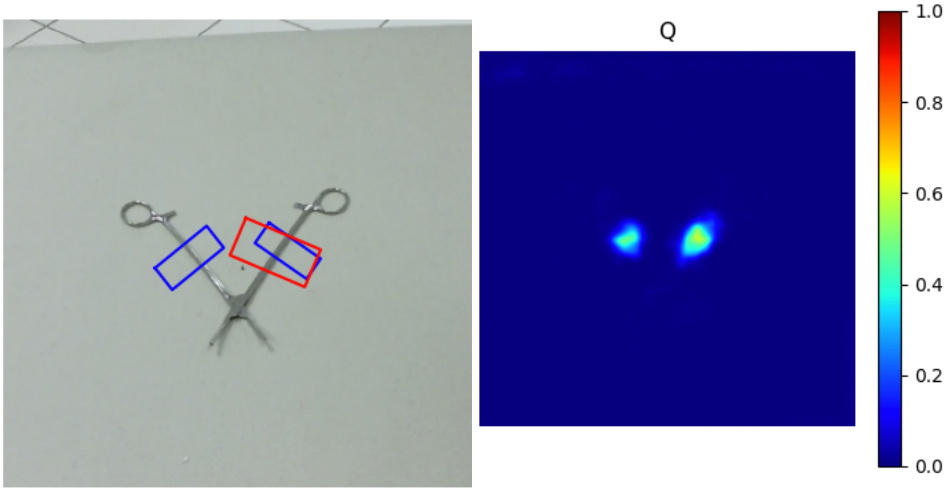

Figure 4. On the left, the real and ground-truth grasping rectangles; on the right, the q value of the image.

Table 2. Comparison of the IOU of different networks.

Network	IOU (Max 1)
GG-CNN	0.79
GG-CNN2	0.88
Contractive-GG-CNN	0.78
Contractive-GG-CNN2	**0.98**
Denoising-GG-CNN	0.41
Denoising-GG-CNN2	0.90
Sparse-GG-CNN	0.73
Sparse-GG-CNN2	0.96
VAE-GG-CNN	0.18
VAE-GG-CNN2	0.26

5.1.2. Grasping Single Tool

An example of the outputs of the GG-CNN2 is shown in Figure 5a. In this figure, the RGB images of the surgical tool with the grasping rectangle, the depth image of the object, the q value, and the angle for four different surgical tools are shown. As we depicted in the images, the q value influences the grasping position, as the grasping rectangle location corresponds to a region with a higher value of q. In Figure 5b, the output of the segmentation of each object is shown.

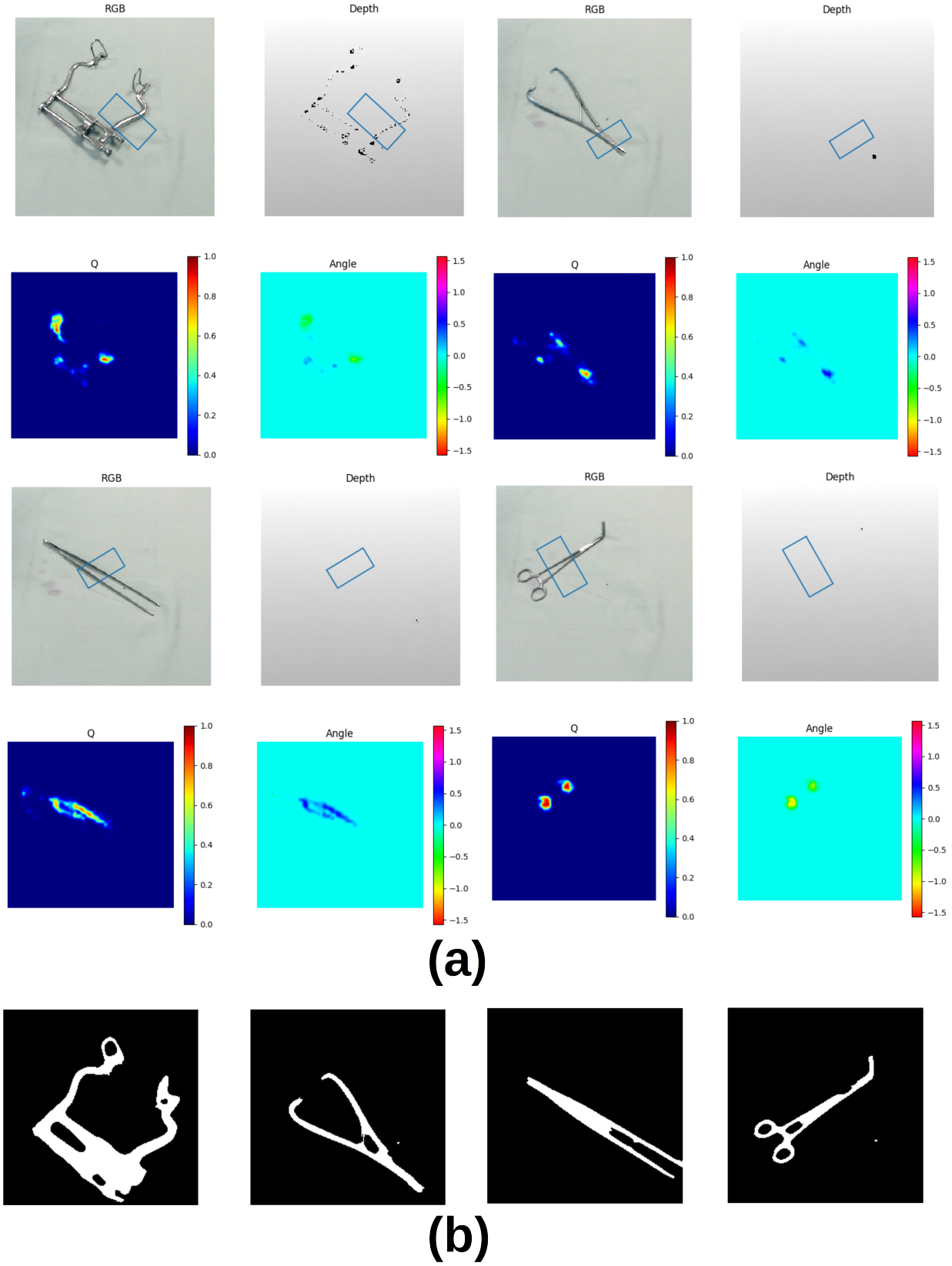

Figure 5. In (**a**), the GG-CNN2 calculates and visualizes four different grasping box rectangles of surgical tools in RGB and depth using the q value and angle. In (**b**), the real-time segmentation of a single surgical tool is matched to the tool in (**a**).

With regard to the GG-CNN with a fixed depth, we achieved a success rate of 62% in grasping both the seen and unseen (73.64%) surgical tools. On the other hand, the GG-CNN2 obtained a success rate of 94.8% and 78.18%, respectively, in grasping the seen and unseen tools. Surprisingly, the performance was better for the unseen objects than for the seen objects. This can be attributed to the fact that the seen surgical tools have a more complex shape than the unseen ones. The GG-CNN2 with fixed depth exhibited a better performance both on the seen and unseen objects compared to the GG-CNN and this is due to its dilated convolutional layers, which provide an improved performance in semantic segmentation tasks. With respect to the GG-CNN and the segmentation method, we obtained a success rate of 74% for the seen objects and 66.36% for the unseen objects. For the GG-CNN2 with the segmentation method, the performance was 85.6% for the seen objects and 90% for the unseen objects. Therefore, we conclude that the segmentation method, when combined with grasping for unseen surgical tools, produces better results than a fixed depth (see Table 3).

Table 3. To evaluate the performance of GGCNN and GGCNN2 for grasping in various environments, we use several datasets, including our Surgical tools (unseen only), Cornell, Daily supplies, Jacquard, and Restaurant object.

Models \ Dataset	Surgical	Cornell	Daily	Jacquard	Restaurant
GGCNN	66.36%	85.39% [48], 88% [49], 91% [1]	91.45% [50]		94.14% [51]
GGCNN2	90%	95.5% [52]		94% [2]	

Our architecture's reliability and effectiveness have been validated and discussed in Table 3. We compared the accuracy of the different grasping tasks using the GGCNN/GGCNN2 models. Based on the results, the models have proven to be reliable and effective for grasping tasks in various environments, with an accuracy of over 80% in actual environments. Given these results, our architecture could be applied to grasp surgical instruments in different environments. However, there are still limitations and improvements needed.

Limitations:

- Despite using a few variations in the training dataset, this model would still not adapt to novel objects due to limited variations in lighting conditions or other environmental factors.
- This model may struggle with objects that have complex geometries or are occluded, as it relies on a simple geometric grasping position.

Improvements:

- Improving the training dataset by incorporating more diverse object shapes, textures, and sizes.
- Exploring new learning techniques such as transfer learning or domain adaptation methods for handling new objects or environments [53,54].
- Using other types of sensors, such as tactile or additional information, provides the model with more information for grasping [55].

5.2. Qualitative Results

5.2.1. Grasping Multiple Tools

Regarding the grasping of multiple surgical tools concurrently, we decided to conduct the following experiments. We used 9 unseen surgical tools and attempted to grasp them using 5 configurations for each tool, resulting in 45 grasping attempts in total. This experiment is illustrated in Figure 6. We applied this protocol to the GG-CNN and GG-CNN2 with fixed depth and with segmentation. We only obtained good results for the GG-CNN with fixed depth, which achieved a success rate of 68.89%. The other combinations resulted in lower success rates for grasping the surgical tools.

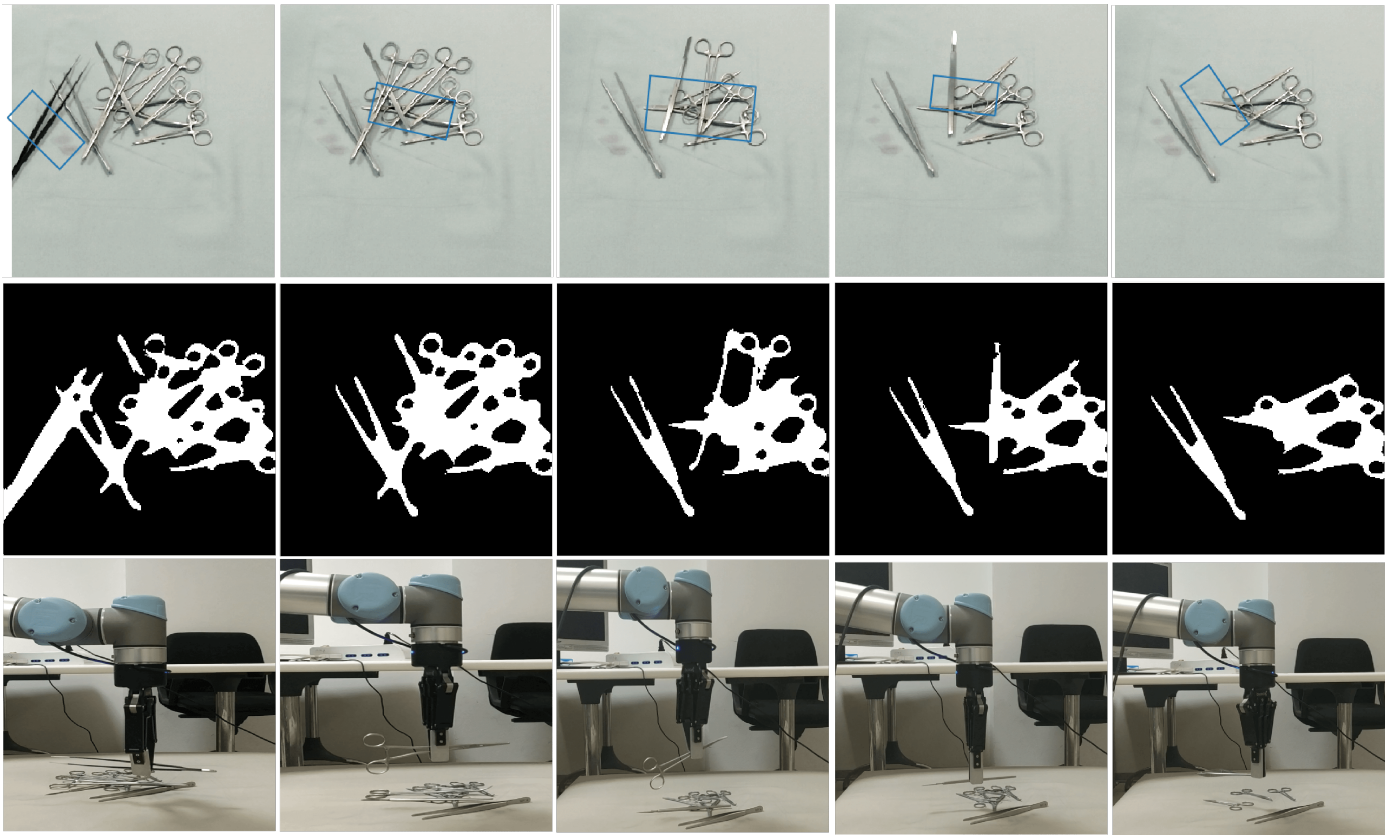

Figure 6. The process of grasping multiple surgical tools in a cluttered environment. During the grasping performance, GG-CNN2 searched a new grasping rectangle, and with color-based segmentation on the scene in the first and the second rows. In the last row, the robotic arm and gripper grasp the tools based on the networks' information.

The architecture is able to grasp the tools, but it has a problem of not completing the task. The issue is described in the following section.

5.2.2. Failure Grasping Examples

During the experimental part involving a single surgical tool, we encountered several common problems. The most frequent issue was the tool falling after being grasped. This was due to the large length of the objects and the lack of consideration of their center of mass. Moreover, the higher weight of the surgical tools and the material they are made of contributed to the problem. The shape of the surgical tools also played a critical role in the success or failure of grasping, as strange shapes are more difficult to grip and thicker objects are easier to hold than thinner ones. We also encountered two failure cases where the GG-CNN2 could not find the grasping box from the input data with differently colored tools (see Figure 7a) or where the segmentation was incorrect (see Figure 7b). Both of these issues arose from the lack of training examples. Additionally, we occasionally encountered an error related to the Moveit! software, which prevented us from grasping the surgical tool. This issue was resolved by running the grasping program a second time.

As for the issues we faced in grasping multiple surgical tools, we encountered the following problems. First, the grasping box was sometimes inaccurately located (see Figure 7c), preventing us from grasping any surgical tool. Second, due to the material or weight of the surgical tool, it would fall after being grasped (see Figure 7d). Finally, the computation of the surgical tool depth was sometimes incorrect because we used the average of the segmentation model's output depth, leading to failures in grasping the object.

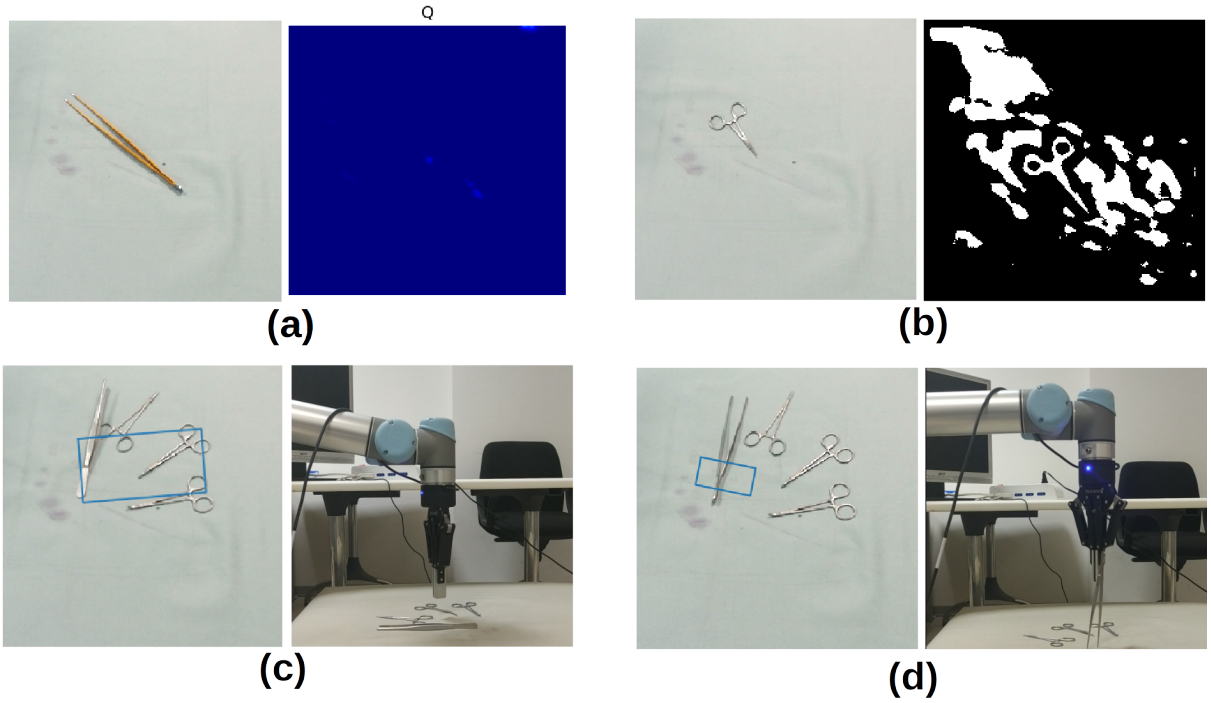

Figure 7. Failure examples: (**a**) no grasping box detection based on q value, (**b**) wrong segmentation as well as no grasping box detection, (**c**) wrong grasping box, and (**d**) tool falling in a cluttered environment.

6. Conclusions

Grasping thin and complex-shaped objects is a challenging task in the field of manipulation. In this paper, we proposed an architecture that utilizes the GG-CNN with a segmentation method for grasping such objects. In the future, we plan to expand the dataset by using more surgical tools and modifying the GG-CNN architecture to achieve even higher success rates in grasping trials. Additionally, we aim to explore how different encoder–decoder network models with a GG-CNN architecture and segmentation can improve grasping multiple surgical tools based on the different colored height map [56] and with a compliance gripper [57]. Finally, we would like to study in depth the grasping of surgical tools with the Deformable Convolutional Net [58] or by adding different modules to the GG-CNN2.

Author Contributions: Conceptualization, J.K. and O.N.; Methodology, J.K. and O.N.; Software, J.K. and O.N.; Validation, J.K. and O.N.; Formal analysis, J.K. and O.N.; Investigation, J.K. and O.N.; Writing—original draft, J.K. and O.N.; Writing—review & editing, O.N., J.K., M.Z.B. and F.C.; Supervision, F.C.; Funding acquisition, F.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors have no relevant financial or non-financial interest to disclose.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
GG-CNN	Generative Grasping Convolutional Neural Network
IOU	Intersection Over Union

ORANGE	ORientation AtteNtive Grasp synthEsis
GR-ConvNet	Generative Residual Convolutional Neural Network
AWGN	Additive White Gaussian Noise
VAE	Variational Autoencoder
UR	Universal Robot

References

1. Morrison, D.; Corke, P.; Leitner, J. Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach. In Proceedings of the Robotics: Science and Systems (RSS), Pittsburgh, PA, USA, 26–30 June 2018.
2. Morrison, D.; Corke, P.; Leitner, J. Learning robust, real-time, reactive robotic grasping. *Int. J. Robot. Res.* **2020**, *39*, 183–201. [CrossRef]
3. Bohg, J.; Morales, A.; Asfour, T.; Kragic, D. Data-driven grasp synthesis—A survey. *IEEE Trans. Robot.* **2013**, *30*, 289–309. [CrossRef]
4. Sahbani, A.; El-Khoury, S.; Bidaud, P. An overview of 3D object grasp synthesis algorithms. *Robot. Auton. Syst.* **2012**, *60*, 326–336. [CrossRef]
5. Mousavian, A.; Eppner, C.; Fox, D. 6-Dof graspnet: Variational grasp generation for object manipulation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 2901–2910.
6. Murali, A.; Mousavian, A.; Eppner, C.; Paxton, C.; Fox, D. 6-Dof grasping for target-driven object manipulation in clutter. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 6232–6238.
7. Depierre, A.; Dellandréa, E.; Chen, L. Jacquard: A large scale dataset for robotic grasp detection. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 3511–3516.
8. Detry, R.; Başeski, E.; Krüger, N.; Popović, M.; Touati, Y.; Piater, J. Autonomous Learning of Object-Specific Grasp Affordance Densities. 2009. Available online: https://iis.uibk.ac.at/public/papers/Detry-2009-SLHR.pdf (accessed on 15 January 2023).
9. Goldfeder, C.; Allen, P.K.; Lackner, C.; Pelossof, R. Grasp planning via decomposition trees. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 4679–4684.
10. Miller, A.T.; Knoop, S.; Christensen, H.I.; Allen, P.K. Automatic grasp planning using shape primitives. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), Taipei, Taiwan, 14–19 September 2003; Volume 2, pp. 1824–1829.
11. Saxena, A.; Driemeyer, J.; Ng, A.Y. Robotic grasping of novel objects using vision. *Int. J. Robot. Res.* **2008**, *27*, 157–173. [CrossRef]
12. El-Khoury, S.; Sahbani, A. Handling objects by their handles. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; number POST_TALK.
13. Mahler, J.; Matl, M.; Liu, X.; Li, A.; Gealy, D.; Goldberg, K. Dex-net 3.0: Computing robust robot vacuum suction grasp targets in point clouds using a new analytic model and deep learning. *arXiv* **2017**, arXiv:1709.06670.
14. Zhang, Z.; Zhou, C.; Koike, Y.; Li, J. Single RGB Image 6D Object Grasping System Using Pixel-Wise Voting Network. *Micromachines* **2022**, *13*, 293. [CrossRef]
15. Bicchi, A.; Kumar, V. Robotic grasping and contact: A review. In Proceedings of the Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), San Francisco, CA, USA, 24–28 April 2000; Volume 1, pp. 348–353.
16. Prattichizzo, D.; Trinkle, J.C.; Siciliano, B.; Khatib, O. Springer handbook of robotics. In *Grasping*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 671–700.
17. Rubert, C.; Kappler, D.; Morales, A.; Schaal, S.; Bohg, J. On the relevance of grasp metrics for predicting grasp success. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 265–272.
18. Balasubramanian, R.; Xu, L.; Brook, P.D.; Smith, J.R.; Matsuoka, Y. Physical human interactive guidance: Identifying grasping principles from human-planned grasps. *Hum. Hand Inspir. Robot. Hand Dev.* 2014, 28, 899–910.
19. Weisz, J.; Allen, P.K. Pose error robust grasping from contact wrench space metrics. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 557–562.
20. Ciocarlie, M.; Hsiao, K.; Jones, E.G.; Chitta, S.; Rusu, R.B.; Şucan, I.A. Towards reliable grasping and manipulation in household environments. In *Experimental Robotics*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 241–252.
21. Herzog, A.; Pastor, P.; Kalakrishnan, M.; Righetti, L.; Asfour, T.; Schaal, S. Template-based learning of grasp selection. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 2379–2384.
22. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
24. Morrison, D.; Corke, P.; Leitner, J. EGAD! An Evolved Grasping Analysis Dataset for diversity and reproducibility in robotic manipulation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 4368–4375. [CrossRef]

25. Yu, H.; Lai, Q.; Liang, Y.; Wang, Y.; Xiong, R. A Cascaded Deep Learning Framework for Real-time and Robust Grasp Planning. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 1380–1386.
26. Wang, S.; Jiang, X.; Zhao, J.; Wang, X.; Zhou, W.; Liu, Y. Efficient fully convolution neural network for generating pixel wise robotic grasps with high resolution images. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 474–480.
27. Mahajan, M.; Bhattacharjee, T.; Krishnan, A.; Shukla, P.; Nandi, G. Semi-supervised Grasp Detection by Representation Learning in a Vector Quantized Latent Space. *arXiv* **2020**, arXiv:2001.08477.
28. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.
29. Gkanatsios, N.; Chalvatzaki, G.; Maragos, P.; Peters, J. Orientation Attentive Robot Grasp Synthesis. *arXiv* **2020**, arXiv:2006.05123.
30. Kumra, S.; Joshi, S.; Sahin, F. Antipodal Robotic Grasping using Generative Residual Convolutional Neural Network. *arXiv* **2019**, arXiv:1909.04810.
31. Sayour, M.H.; Kozhaya, S.E.; Saab, S.S. Autonomous robotic manipulation: Real-time, deep-learning approach for grasping of unknown objects. *J. Robot.* **2022**, *2022*, 2585656. [CrossRef]
32. Xu, R.; Chu, F.J.; Vela, P.A. Gknet: Grasp keypoint network for grasp candidates detection. *Int. J. Robot. Res.* **2022**, *41*, 361–389. [CrossRef]
33. Alliegro, A.; Rudorfer, M.; Frattin, F.; Leonardis, A.; Tommasi, T. End-to-end learning to grasp via sampling from object point clouds. *IEEE Robot. Autom. Lett.* **2022**, *7*, 9865–9872. [CrossRef]
34. Redmon, J.; Angelova, A. Real-time grasp detection using convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1316–1322.
35. Kumra, S.; Kanan, C. Robotic grasp detection using deep convolutional neural networks. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 769–776.
36. Park, D.; Chun, S.Y. Classification based grasp detection using spatial transformer network. *arXiv* **2018**, arXiv:1803.01356.
37. Jiang, Y.; Moseson, S.; Saxena, A. Efficient grasping from rgbd images: Learning using a new rectangle representation. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3304–3311.
38. Lenz, I.; Lee, H.; Saxena, A. Deep learning for detecting robotic grasps. *Int. J. Robot. Res.* **2015**, *34*, 705–724. [CrossRef]
39. Gu, S.; Rigazio, L. Towards deep neural network architectures robust to adversarial examples. *arXiv* **2014**, arXiv:1412.5068.
40. Buades, A.; Coll, B.; Morel, J.M. A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.* **2005**, *4*, 490–530. [CrossRef]
41. Awad, A. Denoising images corrupted with impulse, Gaussian, or a mixture of impulse and Gaussian noise. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 746–753. [CrossRef]
42. Ling, B.W.K.; Ho, C.Y.F.; Dai, Q.; Reiss, J.D. Reduction of quantization noise via periodic code for oversampled input signals and the corresponding optimal code design. *Digit. Signal Process.* **2014**, *24*, 209–222. [CrossRef]
43. Rajagopal, A.; Hamilton, R.B.; Scalzo, F. Noise reduction in intracranial pressure signal using causal shape manifolds. *Biomed. Signal Process. Control.* **2016**, *28*, 19–26. [CrossRef] [PubMed]
44. Ilesanmi, A.E.; Idowu, O.P.; Chaumrattanakul, U.; Makhanov, S.S. Multiscale hybrid algorithm for pre-processing of ultrasound images. *Biomed. Signal Process. Control.* **2021**, *66*, 102396. [CrossRef]
45. Liu, B.; Wang, M.; Foroosh, H.; Tappen, M.; Pensky, M. Sparse convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 806–814.
46. An, J.; Cho, S. Variational autoencoder based anomaly detection using reconstruction probability. *Spec. Lect. IE* **2015**, *2*, 1–18.
47. Coleman, D.; Sucan, I.; Chitta, S.; Correll, N. Reducing the barrier to entry of complex robotic software: A moveit! case study. *arXiv* **2014**, arXiv:1404.3785.
48. Mahajan, M.; Bhattacharjee, T.; Krishnan, A.; Shukla, P.; Nandi, G.C. Robotic grasp detection by learning representation in a vector quantized manifold. In Proceedings of the 2020 International Conference on Signal Processing and Communications (SPCOM), Bangalore, India, 19–24 July 2020; pp. 1–5.
49. Feng, K.; Wei, W.; Yu, Q.; Liu, Q. Grasping Prediction Algorithm Based on Full Convolutional Neural Network. *J. Phys. Conf. Ser.* **2021**, *1754*, 012214. [CrossRef]
50. Zhang, C.; Zheng, L.; Pan, S. Suction Grasping Detection for Items Sorting in Warehouse Logistics using Deep Convolutional Neural Networks. In Proceedings of the 2022 IEEE International Conference on Networking, Sensing and Control (ICNSC), Shanghai, China, 15–18 December 2022; pp. 1–6.
51. Navarro, R. Learning to Grasp 3D Objects Using Deep Convolutional Neural Networks. Ph.D. Thesis, University of Groningen, Groningen, The Netherlands, 2020.
52. Shukla, P.; Kushwaha, V.; Nandi, G.C. Development of a robust cascaded architecture for intelligent robot grasping using limited labelled data. *arXiv* **2021**, arXiv:2112.03001.
53. Kim, J.; Cauli, N.; Vicente, P.; Damas, B.; Bernardino, A.; Santos-Victor, J.; Cavallo, F. Cleaning tasks knowledge transfer between heterogeneous robots: A deep learning approach. *J. Intell. Robot. Syst.* **2020**, *98*, 191–205. [CrossRef]
54. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 23–30.

55. Maus, P.; Kim, J.; Nocentini, O.; Bashir, M.Z.; Cavallo, F. The Impact of Data Augmentation on Tactile-Based Object Classification Using Deep Learning Approach. *IEEE Sensors J.* **2022**, *22*, 14574–14583. [CrossRef]
56. Zeng, A.; Song, S.; Yu, K.T.; Donlon, E.; Hogan, F.R.; Bauza, M.; Ma, D.; Taylor, O.; Liu, M.; Romo, E.; et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1–8.
57. Kim, J.; Mishra, A.K.; Radi, L.; Bashir, M.Z.; Nocentini, O.; Cavallo, F. SurgGrip: A compliant 3D printed gripper for vision-based grasping of surgical thin instruments. *Meccanica* **2022** , 1–16. [CrossRef]
58. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.