Learning to Grasp 3D Objects using Deep Residual U-Nets

Yikun Li, Lambert Schomaker, S. Hamidreza Kasaei

***Abstract*— Grasp synthesis is one of the challenging tasks for any robot object manipulation task. In this paper, we present a new deep learning based grasp synthesis approach for 3D objects. In particular, we propose an end-to-end Convolutional Neural Network to predict the graspable areas of the given objects. We named our approach *Res-U-Net* since the architecture of the network is designed based on U-Net structure and residual network-styled blocks. It devised to plan 6-DOF grasps for any desired object, be efficient to compute and use, and be robust against varying point cloud density and Gaussian noise. We have performed a set of extensive experiments to assess the performance of the proposed approach concerning graspable part detection, grasp success rate, and robustness to varying point cloud density and Gaussian noise. Experiments validate the promising performance of the proposed architecture in all aspects. A video showing the performance of our approach in the simulation environment can be found at http://youtu.be/5_yAJCc8owo**

I. Introduction

Traditional object grasping approaches have been widely used in many industrial settings such as factories assembly lines. In such domains, robots broadly work in tightly controlled conditions to perform object manipulation tasks. Nowadays, service robots are entering human-centric environments. In such unstructured places, generating stable grasp configuration for the household objects is a challenging task due to the high demand for accurate and real-time response under changing and unpredictable environmental conditions [1]. In human-centric environments, an object may have many graspable areas/points, where each one can be used to accomplish a specific task. As an example, consider a robotic cutting task using a knife. The knife has two graspable areas: the handle and the blade. The blade is used to cut through material, and the handle is used for grasping the knife. Therefore, the robot must be able to identify all graspable areas and choose the right one to plan the grasp and complete the task appropriately.

In this paper, we formulate the problem of grasp synthesis, i.e., finding a grasp configuration that satisfies a set of criteria relevant for the grasping task [2], as a learning problem. In particular, we propose a novel deep Convolutional Neural Network (CNN) architecture to predict the graspable areas of the given object. We named it as Res-U-Net, since the network is build based on U-Net network architecture and residual blocks. Our approach is designed to be robust and efficient to compute and use. Besides, we propose a method to find the best collision-free path to approach and grasp the object candidate using a parallel-plate robotic gripper. The advantages of our approach over other state-of-the-art are:

- Most of the recent works forced the robot to approach objects from above vertically (e.g., 3/4-DOF grasp [3], [4]). Such approaches simplify the problem of object grasping and are not able to grasp planar objects, e.g., plates. In this paper, we propose a learning-based 6-DOF grasping approach that allows robots to approach the object from arbitrary directions.
- We demonstrate that our approach outperforms state-of-the-art architectures and enables a robot to pick up different types of objects with a success rate of 83.2%. Fig. 1 shows ten examples of our approach. Furthermore, we have tested the proposed method with a set of never-seen-before objects. Results showed that our approach generalises well to new objects while generating only a small number of false predictions.

We extensively evaluate the performance of the proposed approaches in a simulation environment. The remainder of this paper is structured as follows. After reviewing related work, we discuss our proposed Res-U-Net architecture. We then describe our ranking policy to select the best collision-free path, followed by experimental results and a conclusion.

II. Related Work

Object grasping is one of the fundamental robotic tasks. Although an exhaustive survey is beyond the scope of this paper, we will review a few recent efforts.

Herzog et al. [5] assumed the similarly shaped objects could be grasped similarly and introduced a novel grasp selection algorithm which can generate object grasp poses based on previously recorded grasps. Vahrenkamp et al. [6] shown a system that can decompose novel object models by shape and local volumetric information, and label them with semantic information, then plan the corresponding grasps. Song et al. [7] developed a framework for estimating graspable part of the objects from 2D images (texture and object category are taken into consideration). Kopicki et al. [8] presented a method for one-shot learning of dexterous grasps and grasp generation for novel objects. They trained five basic grasps at the beginning and grasped new objects by generating grasp candidates with contract model and hand-configuration model. Kasaei et al. [9] introduced interactive open-ended learning approach to recognize multiple objects and their grasp affordances. When grasping a new object, they computed the dissimilarity between the new object and known objects and found the most similar object. Then,

The authors are with the Faculty of Science and Engineering, Artificial Intelligence and Computer Science, University of Groningen, 9700 AB Groningen, The Netherlands. {yikun.li, l.r.b.schomaker, hamidreza.kasaei}@rug.nl

We are grateful to the NVIDIA corporation for supporting our research through the NVIDIA GPU Grant Program.

Fig. 1: Examples of predicted grasp by the proposed Res-U-Net network on ten sample objects. See the supplements for videos of the grasping trials.

they try to adopt corresponding grasp configuration. If the dissimilarity is larger than the preset threshold, a new class will be created and learned. Kasaei et al. [10] proposed a data-driven grasp approach to grasp the household object by using top and side grasp strategies. It has been reported that they cannot be applied to grasp challenging objects, e.g., objects that should be grasped by their handle or grasped vertically as for instance a plate [11].

Over the past few years, extraordinary progress has been made in robotic application with the emergence of deep learning approaches. Nguyen et al. [12] researched on detecting graspable parts of the objects using RGB-D images and got satisfactory results. They trained a deep Convolutional Neural Network to learn depth features for object grasping from the camera images, which is proved to outperform the other state-of-the-art methods. Qi et al. [13] studied deep learning on point sets, and they proved the deep neural network can efficiently and robustly learn from point set features. Kokic et al. [14] utilized convolutional neural networks for encoding and detecting graspable parts of the object, class and orientation to formulate grasp constraints. Mahler et al. [3] used a synthetic dataset to train a Grasp Quality Convolutional Neural Network (GQ-CNN) model which can predict the probability of success of grasps from depth images.

Most of these grasp synthesis approaches mount an RGB-D camera on top of the workspace and use the color or depth image to predict the grasp configuration as an oriented rectangle in the image frame (3-DOF). Therefore, such approaches necessitate the gripper pose to be perpendicular to the image plane, which leads to a set of drawbacks. The most important one is that picking up a planar object might be impossible given the top-down vertically approaching the object and also other constraints imposed by the arm or task. In contrast to these approaches, our approach tackles the problem of predicting the 6-DOF grasp pose.

III. Graspable Part Detection

Grasp synthesis refers to the formulation of a stable robotic grasp for a given object. In this paper, we formulate grasp synthesis as a cascaded approach, first predicting the graspable areas using Res-U-Net architecture, and then choosing the best collision-free path to approach and grasp the object. The input to our grasp synthesis framework is a point cloud of an object, which is extracted from a 3D scene using object segmentation algorithms such as [15], [16]. A point cloud of an object, O, is represented as a set of points, $p_i : i \in \{1, \dots n\}$, where each point is described by their 3D coordinates $[x, y, z]$ and RGB information. In this work, we only use geometric information of the object and discard the color data. We therefore represent an object as a fixed occupancy grid of size $32 \times 32 \times 32$ voxels. The obtained representation is then used as the input to the Res-U-Net architecture to predict the graspable parts of the object, g, where $g \in O$. In the following subsections, we describe the detail of the Res-U-Net and the details of the finding the collision-free path to grasp the object is addressed in section IV.

A. Baseline Networks

To make our contribution transparent, we build two baselines including autoencoder architecture [17], and U-Net network [18] and highlight the similarities and differences between these approaches and our Res-U-Net. All the networks contain two essential parts: one is the encoder network, and the other is the decoder network.

The architecture of the encoder-decoder network [17] is

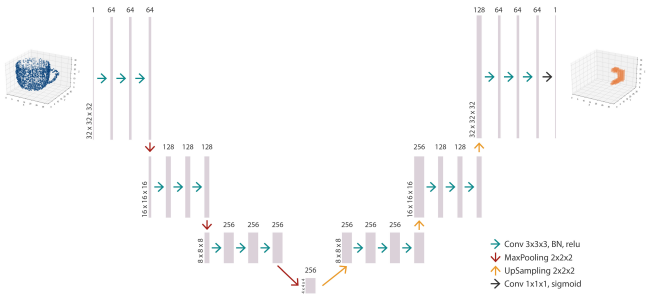

Fig. 2: Structure of the encoder-decoder network. Each grey box stands for a multi-channel feature map. The number of channels is shown on the top of the feature map box. The shape of each feature map is denoted at the lower left of the box. The different color arrows represent various operations shown in the legend.

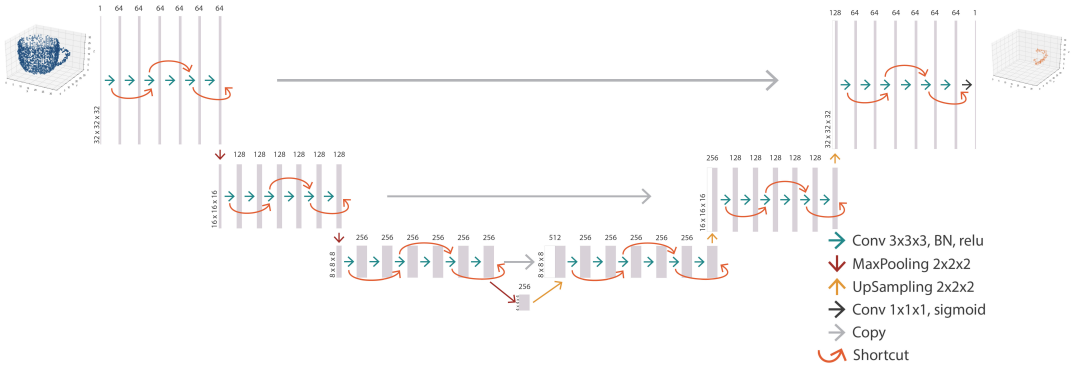

Fig. 3: Structure of the proposed Res-U-Net: compared to the U-Net, we replace the residual blocks with 3D convolutional layers and skipping over layers. This skipping over layers effectively simplifies the network and speeds learning by reducing the impact of vanishing gradients.

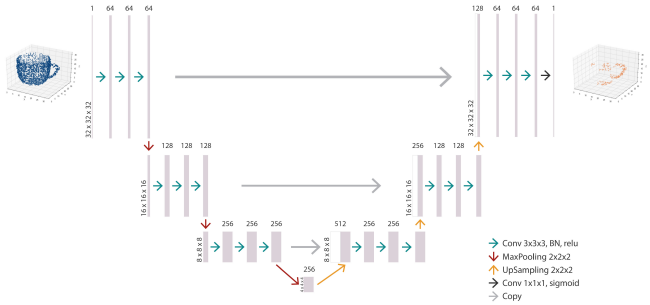

Fig. 4: Structure of the U-Net: compared to the encoder-decoder network, the last feature map of each layer in the encoder part is copied and concatenated to the first feature map of the same layer in the decoder part.

depicted in Fig. 2. This architecture is the lightest one among the selected architectures in terms of the number of parameters and computation, making the network easier and faster to learn. The encoder part of this network has nine 3D convolutional layers (all of them are $3 \times 3 \times 3$), and each of them is followed by batch normalisation and ReLU layer. At the end of each encoder layer, there is a 3D max-pooling layer of $2 \times 2 \times 2$ to produce a dense feature map. Each encode layer is corresponding to a decoder layer. It also has nine 3D convolutional layers. The difference is that instead of having 3D max-pooling layers, at the beginning of each layer, an up-sampling layer is utilised to produce a higher resolution of the feature map. Besides, a $1 \times 1 \times 1$ convolutional layer and a sigmoid layer is attached after the final decoder to reduce the multi-channels to 1.

The architecture of U-Net [18] is shown in Fig. 4. The basic structure of the U-Net and the described encoder-decoder network are almost the same. The main difference is that, in U-Net architecture, the dense feature map is first copied from the end of each encoder layer to the beginning of each decoder layer, and then the copied layer and the up-sampled layer are concatenated.

B. Proposed Res-U-Net Network

The architecture of our approach is illustrated in Fig. 3. As shown in this figure, the network architecture is a combination of U-Net and residual network [19]. We therefore call this network Res-U-Net. We come up with this architecture to retain more information from the input layer and dig more features, inspired by the residual network [19]. Compared to the U-Net, we replace the residual blocks with 3D convolutional layers and skipping over layers. The main motivation is to avoid the problem of vanishing gradients, by reusing activation from a previous layer until the adjacent layer learns its weights. Benefiting from the residual blocks, the network can go deeper since it simplifies the network, using fewer layers in the initial training stages. The encoder and decoder parts are jointly trained to minimises the average reconstruction loss $\mathcal{L}(g^{'}, g)$ between the predicted graspable areas, $g^{'}$, and the ground truth areas, g, over a training set.

IV. Ranking Collision-free Paths for Grasping

To discuss the problem better, we provide a representative example of the proposed approach in Fig.5. As shown in Fig.5 (*a*), we assume that a given object is laying on a planar surface. The object is then extracted from the scene and fed to the Res-U-Net (see *b-c*). After detecting the graspable area of the given object, the point cloud of the object is further processed to determine an appropriate 6-DOF grasp configuration (i.e., the position and the orientation of end-effector in 3D space). In particular, the predicted graspable part of the object is first segmented into m clusters using the K-means algorithm, where m is defined based on the size of graspable part of the object and robot's griper. The centroid of each cluster indicates one grasp candidate (see Fig. 5 (*d*)). Each centroid is considered as the desire pose of the approaching path. We create a pipeline for each grasp candidate and process the object further to define the starting pose of the collision-free approaching path. Inside each pipeline, we generate a Fibonacci sphere by putting the center of the sphere at the grasp candidate and then randomly select N points on the sphere. We then define N linear approaching paths by calculating lines using selected points and the grasp candidate point (i.e., the center of the sphere). In our current setup, N has been set to 256 points which are shown by red lines Fig. 5. In this study, we use the following

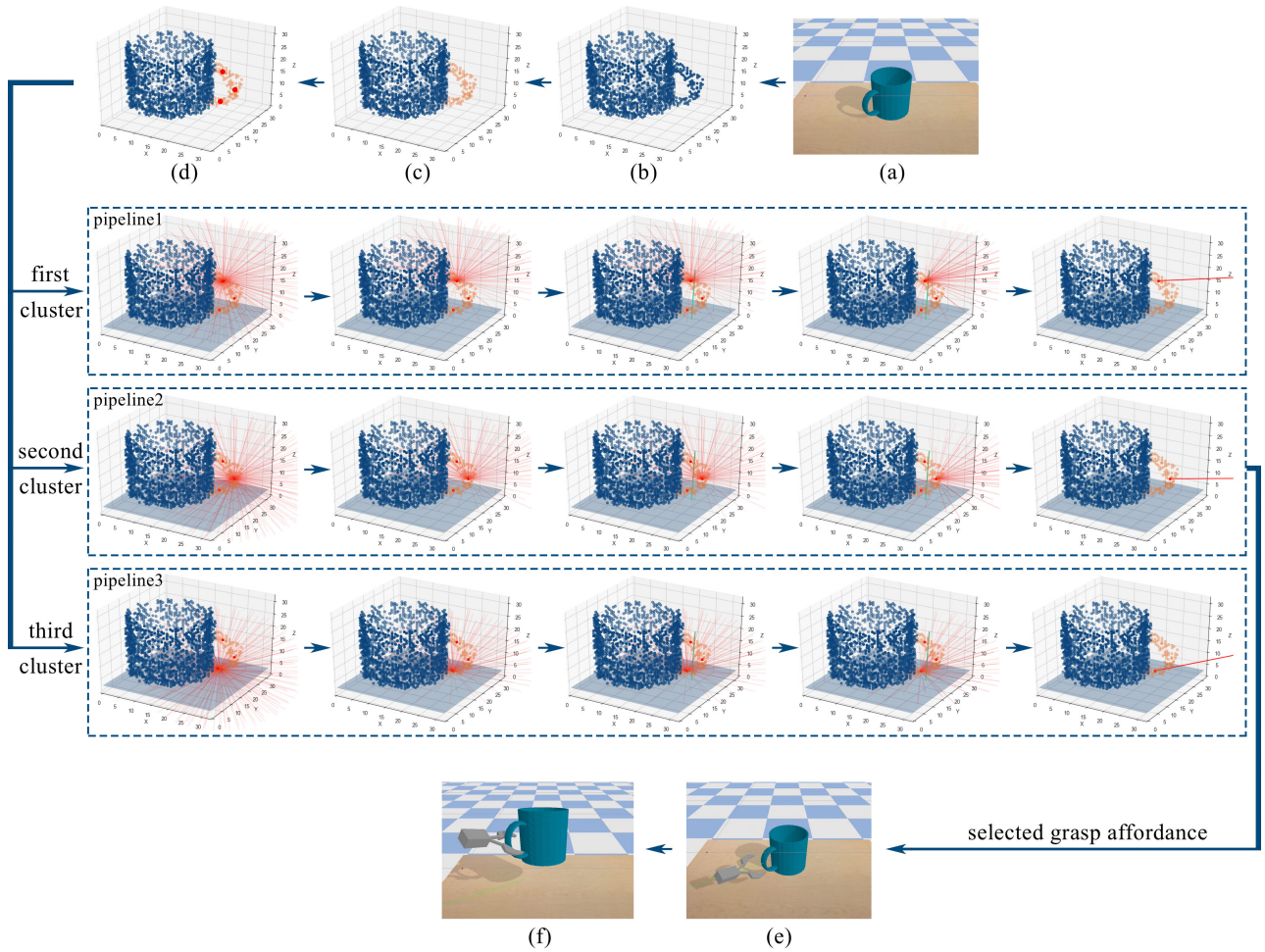

Fig. 5: An illustrative example of the proposed object grasping approach: (*a*) a *Mug* object in our simulation environment; (*b*) point cloud of the object; (*c*) feeding the point cloud to Res-U-Net for detecting the graspable part of object (highlighted by orange color); (*d*) the predicted area is then segmented into three clusters using the K-means algorithm. The centroid of each cluster is considered as a graspable point. Then, the point cloud of the object is further processed in three pipelines to find out an appropriate 6-DoF grasp configuration for each graspable point. In particular, inside each pipeline, a set of approaching paths is first generated based on the Fibonacci sphere (shown by red lines) and the table plane information (shown by a dark blue plane); we then eliminate those paths that go through the table plane. Afterwards, we find the principal axis of the graspable part by performing PCA analysis (the green line shows the main axis), which is used to define the goodness of each approaching path. The best approaching path is finally detected and (*e*) used to perform grasping; (*f*) this snapshot shows a successful example of grasp execution.

procedures to define the best collision-free approaching path:

- **Discard infeasible paths:** by considering the table information, we remove infeasible approaching paths, i.e., those paths that their start point is under the table or colliding with table before reaching the object (*see the second image in each pipeline*).
- **Compute the main axis of the predicted graspable part:** using Principal Component Analysis (PCA). The axis with maximum variance is considered as the main-axis (shown by a green line in the third image of each pipeline).
- **Rank each of the approaching path:** we propose the following equation to rank each of the remaining approaching paths:

$$score = 2\frac{\pi - a}{\pi}\sum_{i=1}^{n} min(1, \frac{1}{d^2 + \epsilon}) \qquad (1)$$

where n represents the number of points of the object, d stands for the distance between the specific approaching path and one of the points in a point cloud model, ϵ is equal to 0.01, and a is the angle between approaching path line and the main axis of the graspable part of the object, ranging from 0 to $\frac{\pi}{2}$. Since [20] has shown that humans tend to grasp object orthogonal to the principal axis, we then calculate $(2*\frac{\pi-a}{\pi})$ in the formula to reduce the score when the path is orthogonal to the principal axis. The lower score means the distances between the approaching path to all points of the objects are farther.

Therefore, the path with the lowest score is selected as a final approaching path for each grasp point candidate. The approaching paths with scores' influence are shown as the fourth image in each pipeline. It is visible that all paths with deeper color represent proper approaching paths. Finally, the best approaching path is selected as the approaching path for the given grasp point (*last figure in each pipeline*).

After calculating the best collision-free approaching path, we instruct the robot to follow the path. Towards this end, we first transform the approaching path from object frame to world frame and then dispatch the planned trajectory to the robot to be executed (Fig. 5 (*e* and *f*)). It is worth to mention, in some situation it is possible that the fingers of the gripper get in contact with the table (which stops the gripper from moving forward). To handle this point, we do slight roll rotation on the gripper to find a better angle between gripper and table to keep gripper moving forward. An illustrative example of the proposed object grasping approach is depicted in Fig. 5.

V. Experimental Results

A set of experiments was carried out to evaluate the proposed approach. In this section, we first describe our experimental setup and then discuss the obtained results.

A. Dataset and Evaluation Metrics

In these experiments, we mainly used a subset of ShapeNetCore [21] containing 500 models from five categories including *Mug, Chair, Knife, Guitar*, and *Lamp*. For each category, we randomly selected 100 object models and convert them into complete point clouds with the pyntcloud package. We then shift and resize the point clouds data and convert them into a $32 \times 32 \times 32$ array as the input size of networks.

We manually labelled an graspable parts for each object to provide ground truth data. Part annotations are represented as point labels. A set of examples of labelled graspable part for different objects is depicted in Fig. 6 (graspable parts are highlighted by orange color). It should be noted that we augment the dataset with by rotating the point clouds along the z-axis for 90, 180 and 270 degrees and flip the point clouds vertically and horizontally from the top view to augment the training and validation data. We obtain 2580 training, 588 validation and 100 test data for evaluation.

We mainly used Average Intersection over Union (IoU) as the evaluation metric. We first compute IoU for each part on each object. Afterwards, for each category, IoU is computed by averaging the per part IoU across all parts on all objects of category. To evaluate the grasping part, we used *success rate* metric, which is defined as the ratio of successful grasps to all performed grasp experiments.

B. Training

All the proposed networks are trained from scratch through `RMSprop` optimizer with the ρ setting to 0.9. We initially set the learning rate to 0.001. If the validation loss does not decrease in 5 epochs, the learning rate is decayed by multiplying the square root of 0.1 until it reaches the minimum learning rate of 0.5×10^{-6}. The `binary cross-entropy` loss is employed in training and the batch size is set to 16. We mainly use Python and Keras library in this study. The training process takes around two days on our NVIDIA Tesla K40m GPU, depending on the complexity of the network.

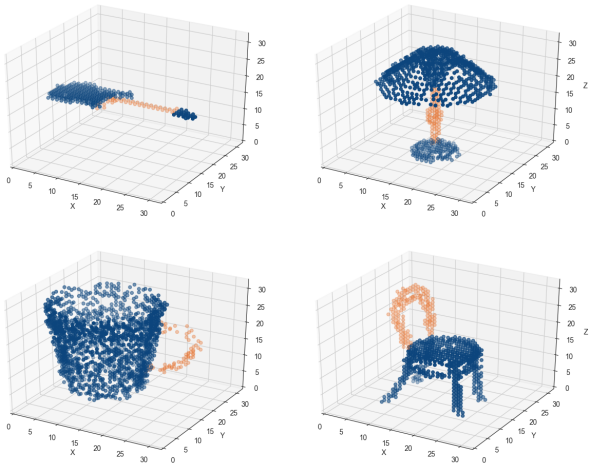

Fig. 6: Examples of labelling graspable part for *Guitar*, *Lamp*, *Mug*, and *Chair* objects: point cloud of the object is shown by dark blue and graspable parts are highlighted by orange color.

C. Graspable Part Prediction

Figure 7 shows the progress of the proposed networks over 100 epochs. By comparing all the experiments, it is visible that the encoder-decoder network performs much worse than the other approaches. In particular, the final IoU of the encoder-decoder network was 28.9% and 22.3% on training and validation data respectively. The U-Net performs much better than the encoder-decoder network. Its final IoU is 80.1% on training and 71.4% on validation data. The proposed Res-U-Net architecture clearly outperformed the others by a large margin. The final IoU of Res-U-Net was 95.5% and 77.6% on training and validation data respectively. Particularly, in the case of training, it was 15.4 percentage points (p.p.) better than U-Net and 66.6 p.p. better

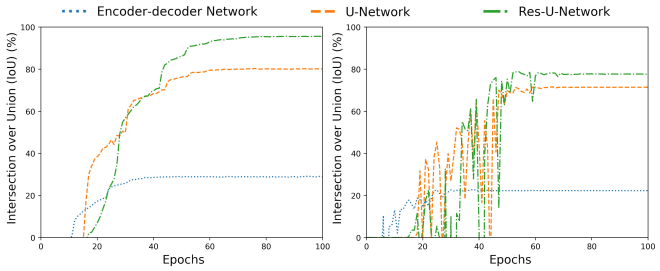

Fig. 7: Train and validation learning curves of different approaches: (*left*) Line plots of IoU over training epochs; (*right*) Line plots of IoU over validation epochs.

than the encoder-decoder network, in the case of validation, it was 6.2 p.p., and 55.3 p.p. better than U-Net and encoder-decoder network respectively.

D. Grasping Results

We evaluate our grasp methodology using a simulated robot[1]. In particular, we build a simulation environment to verify the capability of the proposed grasp approach. The simulation is developed based on the Bullet physics engine. We only consider the end-effector pose $(x, y, z, roll, pitch, yaw)$ to simplify the complexity and concentrate on evaluating the proposed approach.

We design a grasping scenario that the simulated robot first grasps the object and then picks it up to a certain height to see if the object slips due to bad grasp or not. A particular grasp was considered a success if the robot is able to complete the task. In this experiment, we randomly selected 50 different objects for each of the five mentioned categories. In each experiment, we randomly place the object on the table region and also rotate it along the z-axis. It is worth to mention that all test objects were not used for training the neural networks. We achieved a grasp success rate of 83.2% (i.e., 208 success out of 250 trials). The detailed outcomes of the experiments are summarised in Table I. Figure 1 shows the grasp detection results of ten example objects. A video of this experiment is available online at **http://youtu.be/5_yAJCc8owo**

Two sets of experiments were carried out to examine the robustness of the proposed approach with respect to varying point cloud density and Gaussian noise. In particular, in the first set of experiments, the original density of training objects was kept and the density of testing objects was reduced (downsampling) from 1 to 0.5. In the second set of experiments, nine levels of Gaussian noise with standard deviations from 1 to 9 mm were added to the test data. The results are summarised in Fig. 8.

From experiments of reducing density of test data (i.e. Fig.8 (*left*), it was found that our approach is robust to low-level downsampling i.e., with 0.9 point density, the success rate remains the same. In the case of mid-level downsampling resolution (i.e. point density between 0.6 and 0.8), the grasp success rate dropped around 20%. It can be concluded from

TABLE I: Grasp success rate on five categories

Category	Success rate	Success / Total
Mug	0.86	43 / 50
Chair	0.80	40 / 50
Knife	0.84	42 / 50
Guitar	0.80	40 / 50
Lamp	0.86	43 / 50
Average	0.832	208 / 250

[1]Given COVID-19 lockdown situation, it was not possible to test the work in a real-robot scenario. To address the gap between simulation and reality, we will try our best to add real-robot experiments in the camera-ready version as we have done such experiments for the other projects [22], [23]. Previously, it has been shown that models trained exclusively with synthetic grasp data can perform successfully in the real world [3].

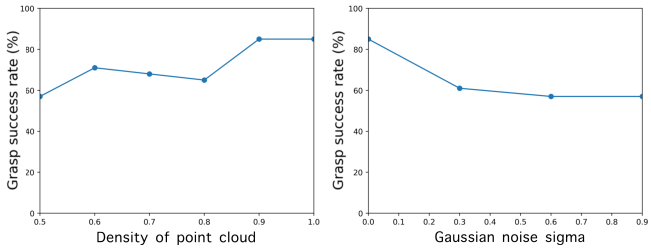

Fig. 8: The robustness of the Res-U-Net to different level of Gaussian noise and varying point cloud density: (*left*) grasp success rate against down-sampling probability; (*right*) grasp success rate against Gaussian noise sigma.

Fig.8 (*left*) that when the level of downsampling increases to 0.5, the grasp success rate dropped to 57% rapidly.

In the second round of experiment, Gaussian noise is independently added to the X, Y and Z-axes of the given test object. As shown in Figure 8 (*right*), performance decrease when the standard deviation of the Gaussian noise increases. In particular, when we set the sigma to 0.3, 0.6 and 0.9, the success rates are dropped to 61%, 57%, and 57% respectively.

Our approach was trained to grasp five object categories. In this experiment, we examine the performance of our approach by a set of 50 never-seen-before objects. In most of the cases, the robot could detect an appropriate grasp configuration for the given object and completed the grasping scenario. This observation showed that the proposed Res-U-Net could use the learned knowledge to grasp most of the unknown objects correctly. In particular, a never-seen-before object that is similar to one of the known ones (i.e., they are familiar) can be grasped similarly. Figure 9 shows the steps taken by the robot to grasp a set of unknown objects in our experiments.

In both experiments (i.e., grasping known and unknown objects), we have encountered three types of failure modes. First, Res-U-Net may fail to predict an appropriate part of the object for grasping. Second, grasping may fail because of the collision between gripper, object, and table, or the predicted graspable part is too small for the given object (e.g., Guitar), or too large to fit in the robot's gripper (e.g., the body of Mug), or if the object is too big or slippery (e.g., Chair and Lamp). The third case of failure happens when the finger of gripper is tangent to one of the surfaces of the object. In such cases, although the graspable part of the object is correctly predicted, the robot pushes the object instead of grasping it.

Another set of experiments was performed to estimate the execution time of the proposed approach. Three components mainly make the execution time: perception, graspable part prediction, and finding the best collision-free approaching path. We measured the run-time for ten instances of each. Perception of the environment and converting the point cloud of the object to appropriate voxel-based representation (on average) takes 0.15 seconds. graspable part prediction by Res-U-Net requires an average of 0.13 seconds, and finding

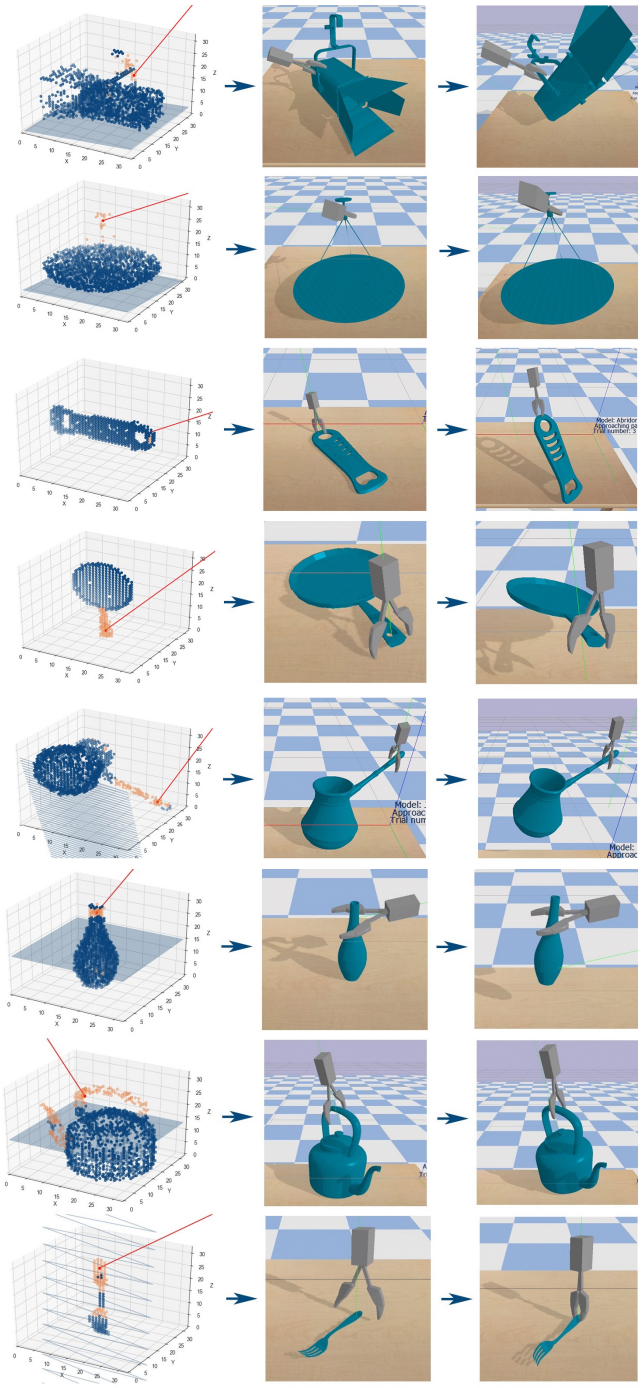

Fig. 9: Examples of grasping unknown objects: Our approach is able to predict the appropriate graspable part and collision-free path for each object that lead to successful pickups of unknown objects.

suitable grasp configuration demands another 1.32 seconds. Therefore, finding a complete grasp configuration for a given object on average takes about 1.60 seconds.

VI. Conclusion and Future Work

In this paper, we present a novel deep convolutional neural network named Res-U-Net to detect graspable parts of 3D Objects. The point cloud of the object is further processed to determine an appropriate grasp configuration for the predicted graspable parts of the object. To validate our approach, we built a simulation environment and conducted an extensive set of experiments. Results show that the overall performance of the proposed Res-U-Net is clearly better than the best results obtained with the U-Net and Autoencoder approaches. We also test our approaches by a set of never-seen-before objects. It was observed that, in most of the cases, our approach was able to detect graspable parts of the objects correctly and perform the proposed grasp scenario successfully. In the continuation of this work, we plan to evaluate the proposed approach in clutter scenarios such as clearing a pile of toy objects. We would also like to investigate the possibility of considering Res-U-Net for task-informed grasping scenarios.

References

[1] J. J. Gibson, *The ecological approach to visual perception: classic edition*. Psychology Press, 2014.

[2] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[3] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[4] D. Morrison, P. Corke, and J. Leitner, "Closing the Loop for Robotic Grasping: A Real-time, Generative Grasp Synthesis Approach," in *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[5] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour, and S. Schaal, "Template-based learning of grasp selection," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2379–2384.

[6] N. Vahrenkamp, L. Westkamp, N. Yamanobe, E. E. Aksoy, and T. Asfour, "Part-based grasp planning for familiar objects," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 919–925.

[7] H. O. Song, M. Fritz, D. Goehring, and T. Darrell, "Learning to detect visual grasp affordance," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 798–809, 2015.

[8] M. Kopicki, R. Detry, M. Adjigble, R. Stolkin, A. Leonardis, and J. L. Wyatt, "One-shot learning and generation of dexterous grasps for novel objects," *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 959–976, 2016.

[9] S. H. Kasaei, M. Oliveira, G. H. Lim, L. S. Lopes, and A. M. Tomé, "Towards lifelong assistive robotics: A tight coupling between object perception and manipulation," *Neurocomputing*, vol. 291, pp. 151–166, 2018.

[10] S. H. Kasaei, N. Shafii, L. S. Lopes, and A. M. Tomé, "Object learning and grasping capabilities for robotic home assistants," in *Robot World Cup*. Springer, 2016, pp. 279–293.

[11] N. Shafii, S. H. Kasaei, and L. S. Lopes, "Learning to grasp familiar objects using object view recognition and template matching," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2895–2900.

[12] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Detecting object affordances with convolutional neural networks," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2765–2770.

[13] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[14] M. Kokic, J. A. Stork, J. A. Haustein, and D. Kragic, "Affordance detection for task-specific grasping using deep learning," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 91–98.

[15] S. H. Kasaei, J. Sock, L. S. Lopes, A. M. Tomé, and T.-K. Kim, "Perceiving, learning, and recognizing 3d objects: An approach to cognitive service robots," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] J. Sock, S. Hamidreza Kasaei, L. Seabra Lopes, and T.-K. Kim, "Multi-view 6d object pose estimation and camera motion planning using rgbd images," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[17] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang, "Object contour detection with a fully convolutional encoder-decoder network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 193–202.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] R. Balasubramanian, L. Xu, P. D. Brook, J. R. Smith, and Y. Matsuoka, "Human-guided grasp measures improve grasp robustness on physical robot," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2294–2301.

[21] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Stanford University — Princeton University — Toyota Technological Institute at Chicago, Tech. Rep. arXiv:1512.03012 [cs.GR], 2015.

[22] S. H. Kasaei, M. Ghorbani, J. Schilperoort, and W. van der Rest, "Investigating the importance of shape features, color constancy, color spaces and similarity measures in open-ended 3d object recognition," *arXiv preprint arXiv:2002.03779*, 2020.

[23] S. Luo, H. Kasaei, and L. Schomaker, "Accelerating reinforcement learning for reaching using continuous curriculum learning," *arXiv preprint arXiv:2002.02697*, 2020.