electronics

Article

Deep Learning-Based Ensemble Approach for Autonomous Object Manipulation with an Anthropomorphic Soft Robot Hand

Edwin Valarezo Añazco [1], Sara Guerrero [2], Patricio Rivera Lopez [3], Ji-Heon Oh [3], Ga-Hyeon Ryu [3] and Tae-Seong Kim [4,*]

[1] Faculty in Electricity and Computation FIEC, Escuela Superior Politecnica del Litoral ESPOL, Guayaquil 090202, Ecuador; edgivala@espol.edu.ec

[2] Faculty of Architecture and Design, Universidad Espiritu Santo UEES, Samborondón 092301, Ecuador; saguerrero@uees.edu.ec

[3] Department of Electronics and Information Convergence Engineering, Kyung Hee University, Yongin 17104, Republic of Korea; patoalejor@khu.ac.kr (P.R.L.); dhwlgjs3@khu.ac.kr (J.H.O.); yugacandy@khu.ac.kr (G.H.R.)

[4] Department of Biomedical Engineering, Kyung Hee University, Yongin 17104, Republic of Korea

* Correspondence: tskim@khu.ac.kr

Abstract: Autonomous object manipulation is a challenging task in robotics because it requires an essential understanding of the object's parameters such as position, 3D shape, grasping (i.e., touching) areas, and orientation. This work presents an autonomous object manipulation system using an anthropomorphic soft robot hand with deep learning (DL) vision intelligence for object detection, 3D shape reconstruction, and object grasping area generation. Object detection is performed using Faster-RCNN and an RGB-D sensor to produce a partial depth view of the objects randomly located in the working space. Three-dimensional object shape reconstruction is performed using U-Net based on 3D convolutions with bottle-neck layers and skip connections generating a complete 3D shape of the object from the sensed single-depth view. Then, the grasping position and orientation are computed based on the reconstructed 3D object information (e.g., object shape and size) using U-Net based on 3D convolutions and Principal Component Analysis (PCA), respectively. The proposed autonomous object manipulation system is evaluated by grasping and relocating twelve objects not included in the training database, achieving an average of 95% successful object grasping and 93% object relocations.

Keywords: deep learning; 3D robot vision; autonomous object grasping; autonomous robots

check for updates

Citation: Valarezo Añazco, E.; Guerrero, S.; Rivera Lopez, P.; Oh, J.-H.; Ryu, G.-H.; Kim, T.-S. Deep Learning-Based Ensemble Approach for Autonomous Object Manipulation with an Anthropomorphic Soft Robot Hand. *Electronics* **2024**, *13*, 379. https://doi.org/10.3390/electronics13020379

Academic Editor: Jian Sun

Received: 17 October 2023
Revised: 12 December 2023
Accepted: 20 December 2023
Published: 17 January 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Autonomous visuomotor manipulation by a robot hand system is one of the most widely investigated topics in robotics because of the growing demand for delicate manipulation with robots [1–3]. Enhancing the autonomy of robots could allow service applications for robots and improve human–robot interaction [4–6]. Autonomous object manipulation requires various form of machine intelligence such as automatic detection or identification of the object position [7], 3D shape information [8,9], object orientation, and grasping (touching) areas [10–12].

Object detection algorithms allow robots to detect and identify objects randomly located in the working space. Deep learning is widely investigated for object detection from RGB images due to self-feature extraction [13]. Faster-RCNN is a well-defined object detector used for real-time applications such as autonomous object manipulation because of its object detection accuracy and inference speed [14,15]. Recent research focuses on integrating object detectors with object shape inference to enrich the preserved information [16]. For instance, Sudharkar et al. [17] propose the integration of R-3D-YOLOv3 on an embedded vision for efficient animal detection and 3D shape reconstruction in dynamic environments. Sudharkar applied the model to approximately 1600 images of Indian stray and wild animals, achieving a commendable accuracy of 84.18%. The results demonstrate the model's effectiveness in identifying and reconstructing 3D views of moving animals near roads, showcasing the potential of ensemble systems to generate scene information with DL-based algorithms.

Once an object is detected based on color information (RGB images), a partial depth view of the object (i.e., a single view of point clouds) can be obtained with an RGB-D camera [18]. It is useful for robot motor control when 3D shape or volume information of an object becomes available. Three-dimensional shape reconstruction was initially attempted by classifying the partial views of objects in categories of known objects (i.e., a database of objects) to produce a complete object shape via shape matching [19,20]. Xialong et al. [17] proposed a system for scene and human reconstruction based on non-rigid deformation and 2D–3D feature fusion modules. Xialong applied the method to address large-scale motion challenges. The results demonstrate improved clarity in object identification and highlight the effectiveness of capturing dynamic scenes with enhanced feature extraction. These initial works [17,19,20] are limited since only objects in the database can be matched or reconstructed. Another approach reconstructs 3D shapes using images of the same object from different angles and patching all the images together. For instance, Xinpeng et al. [21] proposed a system to capture images of the target scene with nine cameras and reconstruct the 3D scene using U-Net-based architectures. Instead of matching or using multiple images, 3D object shapes could be reconstructed using DL. For instance, Valarezo et al. [22] proposed a 3D object shape reconstructor based on U-Net 3D-CNN with a bottle-neck skip connection block (3D U-Net-based BNSC) to reconstruct trained and untrained objects from a single partial view. The idea is to reconstruct the 3D shapes of objects using 3D U-Net-based BNSC to generate the grasping position and orientation angle.

Recently, DL has also been investigated to produce machine intelligence for generating grasping information, i.e., object grasping areas and grasping orientation. For instance, Brahmbhatt et al. [23] proposed an Object Grasping Areas Generator (OGAG) based on U-Net to estimate the grasping areas over the object using a database of human grasping demonstrations. Brahmbhatt reported an average matching error of 11.64%. Choi et al. [24] used a convolutional neural network (CNN) to estimate the most likely grasping direction and wrist orientation from a collected database of successful grasping samples. Choi's system was evaluated by grasping ten objects with a soft robot hand from six grasping directions and four orientations. These previous works [23,24] show the potential of DL to estimate object grasping areas. However, they are limited to grasping objects from a discrete number of wrist orientation angles.

Few previous works performed autonomous object manipulation with anthropomorphic robot hands [7,25]. Ficuciello et al. [25] proposed an autonomous object manipulation using the KUKA lightweight robot 4+, the Schunk five-finger hand, and a single RGB-D camera. Ficuciello first performed object shape classification based on geometric features and then used reinforcement learning (RL) combined with human demonstrations to grasp three objects. Della Santina et al. [7] used the KUKA LWR robot arm, Pisa/IIT anthropomorphic soft robot hand, and a single RGB camera (no depth) for autonomous object grasping. Santina's system used YOLOv2 to detect the objects and deep neural networks (DNNs) to estimate the grasping orientation angle from nine discrete orientations.

This work presents an autonomous object manipulation system with a single RGB-D vision sensor and a flexible anthropomorphic soft robot hand (i.e., the qb Soft Hand [26]) via deep learning (DL)-based vision intelligence. Our proposed autonomous system involves the use of machine vision intelligence, including Faster-RCNN, Principal Component Analysis (PCA), 3D U-Net-based BNSC, and 3D U-Net-based OGAG, to grasp and relocate various objects with different shapes and sizes. Our contribution is the grasping position and orientation estimation. First, the grasping position is calculated from the estimated object grasping areas, which are inferred using the complete object shape and human-like grasping areas. Second, the wrist orientation is a continuous angle from lateral (0°) to top (90°) grasping, adapting to each object. The previous works of Choi et al. [24] and Della Santina et al. [7] only utilized discrete orientation angles such as lateral and top grasping. The proposed autonomous system was tested by manipulating twelve objects not included in the training database. The proposed autonomous object manipulation system achieved an average success rate of 94% in the grasping and relocation tasks.

The rest of the paper is organized as follows. Section 2 describes our proposed autonomous object manipulation system including each database used to train the DL vision intelligence, the implementation details, and the definitions of the manipulation tasks. Section 3 includes qualitative and quantitative results. Section 4 presents a discussion of the experimental results, a comparison with previous manipulation studies, and the limitations. Section 5 states the conclusions of this research article.

2. Materials and Methods

2.1. Visuomotor Robot System Setup

Figure 1 shows the visuomotor robot system and workflow of our autonomous object manipulation system. From the left, the motor components include a UR3 robot arm, the anthropomorphic soft robot hand (i.e., the qb Soft Hand), and a single vision sensor. The qb Soft Hand has 19 degrees of freedom (DoF) controlled with a single motor. The vision sensor is an Intel RealSense D415 RGB-D camera able to sense the color and depth information of an object at a minimum distance of 45 cm with a 2MP resolution. The vision intelligence based on DL is operated in three stages. First, Faster-RCNN detects the object. Its position is found in the working space. The RGB-D sensor captures a partial depth view of the objects. Second, our 3D U-Net-based BNSC reconstructs the complete 3D shape of the object from the partial depth view. Third, our 3D U-Net-based OGAG and PCA estimate the object manipulation parameters of position and orientation, respectively. Finally, the object is automatically grasped and relocated using the UR3 and qb Soft Hand system.

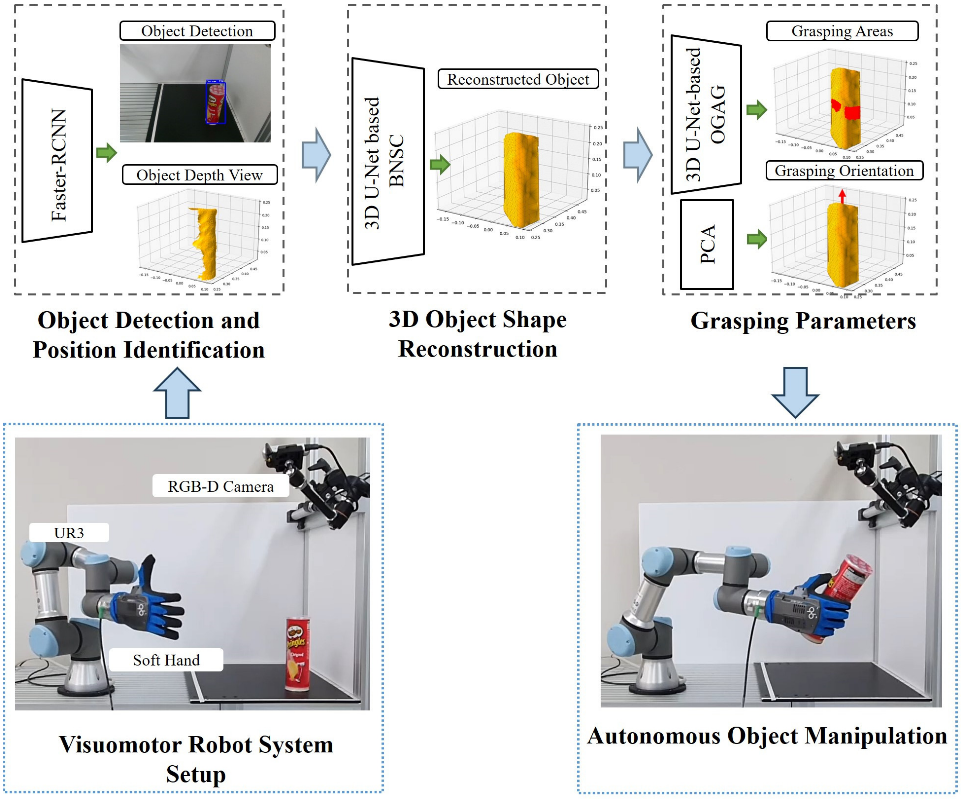


Figure 1. Visuomotor robot system setup and workflow of the proposed autonomous object manipulation system.

2.2. Object Detection and Position Identification

2.2.1. Database

The object detector was trained using Open Images V4 database [27,28]. Open Images V4 has 1.74 million RGB images with 14.6 million bounding boxes for 600 object categories. The manually drawn bounding boxes exist in the database with the labels assigned to the corresponding category. The images from Open Image V4 were divided into 125,436 images for testing, 41,620 images for validation, and the remaining images for the training datasets. The resolution of the images is 1600 × 1200 pixels and the size of the detected objects varies in each image.

2.2.2. Object Detector

Faster-RCNN was adopted for object detection, which is an artificial neural network based on 2D CNN with two stages [15]. First, a regional proposed network (i.e., RPN) generates proposals of the detected bounding boxes with different sizes and objectness scores for each proposal. Second, Fast R-CNN detects the objects based on the proposals generated by RPN [15]. Faster-RCNN was previously evaluated, achieving a mean average precision (mAP) of 42.7% with the MS COCO database [29]. The pixel information from the detected bounding box is paired with the depth information sensed from the RGB-D camera to produce a depth view of the seeing object surface as a point cloud.

2.3. 3D Object Shape Reconstruction

2.3.1. Database

The Grasp database has information on 590 objects sensed and saved as mesh models [30]. The objects are groceries, tools, toys, drugstore products, and household objects. The sizes and materials depend on the object. For instance, tin cans are about 10 cm in height and made of aluminum, cracker boxes are about 25 cm in height and made of carton. All objects in the Grasp database are expressed in a voxel grid 40 × 40 × 40 as binvox files. Binvox is a software that rasterizes 3D object models into 3D voxel grids [31,32]. The binvox object's files were built following the guidelines of [33]. First, the meshes were placed in Gazebo, homepage: https://gazebosim.org/home. Gazebo is a 3D software used for robot simulations [34]. Then, various partial depth views of the object from different angles were created using a virtual depth camera. Finally, a 3D gridded mesh of the visible pixels was produced using Binvox, where label 1 corresponds to object voxels and label 0 to the background.

2.3.2. Three-Dimensional Object Shape Reconstructor

Our object shape reconstructor is based on U-Net with 3D convolutional layers, bottle-neck layers, and skip connections (U-Net-based BNSC 3D-CNN). The 3D convolutional layers are used because of their larger 3D receptive field [35]. The skipped connections propagate the features and training error to the whole network, and the bottle-neck layers reduce the trainable parameters [22].

The 3D U-Net-based BNSC method is an artificial neural network with a contractive path, expensive path, and skip connections [36]. The contractive path captures the object's contextual information (i.e., shape and size) using two BNSC blocks and two max pooling layers. The expansive path reconstructs the object shape using two BNSC blocks and two up-sampling layers. The skip connections connect the layers from the contractive path with their equivalent layers in the expansive path. Also, two skip connections were added around each BNSC block in the contractive path to maximize feature sharing and backpropagation of the training error. The details of the 3D U-Net-based BNSC architecture are available in [22].

2.4. Grasping Parameters

2.4.1. Object Grasping Areas Database

ContactDB is a public database with grasping area information of objects based on human hands (i.e., object grasping areas) [23,37]. ContactDB provides object grasping areas for 50 household objects. The object grasping areas were created by measuring the heat left by the touch of the human hand on the object using a thermal camera. The objects were 3D-printed in their original sizes using thermoplastic polyester (i.e., PLA) to retain the thermal human handprints. The object grasping areas are expressed as voxel grids of 40 × 40 × 40, where the inner area of the object is marked as occupied grids to match the objects from the Grasp database.

2.4.2. Object Grasping Areas Generator (OGAG)

Our 3D U-Net-based OGAG is based on the work of Brahmbhatt et al. [23] because of its average matching error for object grasping area estimation. The original structure was modified by adding the bottle-neck skip connections block (BNSC block) to reduce the network size and improve the feature shearing for training.

Figure 2 shows the structure of the 3D U-Net-based OGAG. A 3D convolutional layer extracts 64 features (@64) from the reconstructed object using a kernel size of 4 × 4 × 4 as the input. The contractive path extracts features using four BNSC blocks and one max pooling layer right after the second BNSC block. Also, skip connections were added around each BNSC block to maximize the depth features shared inside the contractive path and with the expansive path. The expansive path has four BNSC blocks, one up-sampling layer after the second BNSC block, and the skip connections from the contractive path. At the output, a 3D convolutional layer infers the grasping area over the object using a kernel size of 4 × 4 × 4 and one feature (@1).

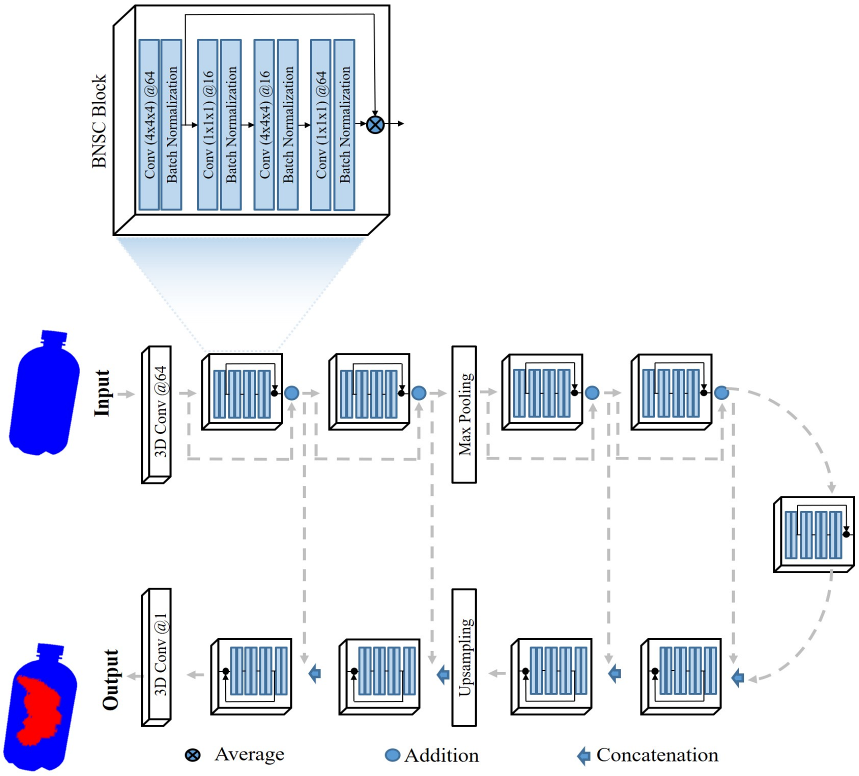


Figure 2. Details of 3D U-Net-based OGAG architecture.

Additionally, the 3D U-Net-based OGAG uses average, addition, and concatenation layers to ensemble the extracted features, as shown in Figure 2. The average layer performs an averaging between the features extracted from the first and last layers of the BNSC block.

The addition layer is used to ensemble the features extracted between the BNSC block of the contractive path. Finally, the concatenation layers groups the features extracted from the contractive path with the features from the expansive path.

Each BNSC block is composed of four convolutional layers, including bottle-neck layers, as shown in Figure 2. The bottle-neck layers are convolutional layers with a kernel size 1 × 1 × 1 used to reduce or restore the feature maps [38,39]. The main advantage of bottle-neck layers is reducing the network size because of their 1 × 1 × 1 kernel. The first convolutional layer of the BNSC block extracts 64 features using a kernel size of 4 × 4 × 4. The second convolutional layer is a bottle-neck layer that reduces the number of features from @64 to @16. The third convolutional layer extracts 16 features using a kernel size of 4 × 4 × 4. At the output of the BNSC block, a bottle-neck layer restores the features from @16 to @64. Batch normalization is used after each convolutional layer.

The qb Soft Hand does not allow independent control of each finger (i.e., one motor controls the entire grasping motion of the fingers). Thus, the inferred object grasping areas are reduced to a single centroid point (i.e., grasping position). The anthropomorphic soft robot hand closes to grasp the objects after reaching the grasping position and opens to release the object in the relocation target position.

2.4.3. Grasping Orientation Angle

The grasping orientation angle aligned the anthropomorphic soft robot hand to the object for power grasping. Power grasping is defined as grasping the object with most of the fingers around it to maximize the area in contact. The grasping orientation is the angle of the principal component regarding the robot frame (i.e., the world frame). The principal component is computed using PCA over the point cloud of the 3D-reconstructed object. Then, the soft robot hand can be aligned in a continuous range from 0° (for lateral grasping) to 90° (for top grasping) according to the direction of the object's principal component.

2.5. Implementation Details

The Intel RealSense RGB-D camera is set up using the Python library Pyrealsense2 from Intel [40]. The UR3 robot arm is connected to a PC using a TCP/IP via an Ethernet socket. UR3 moves to the grasping position using the inverse kinematics package from the URX Python library [41]. The qb Soft Hand is connected via a USB cable and controlled using the C++ library from qb Robotics [26]. The grasping enclosure force was set to 80% of its capacity. Grasping positions less than 6.5 cm high (Z-axis) were not executed to avoid collision with the table.

2.6. Manipulation Tasks and Objects

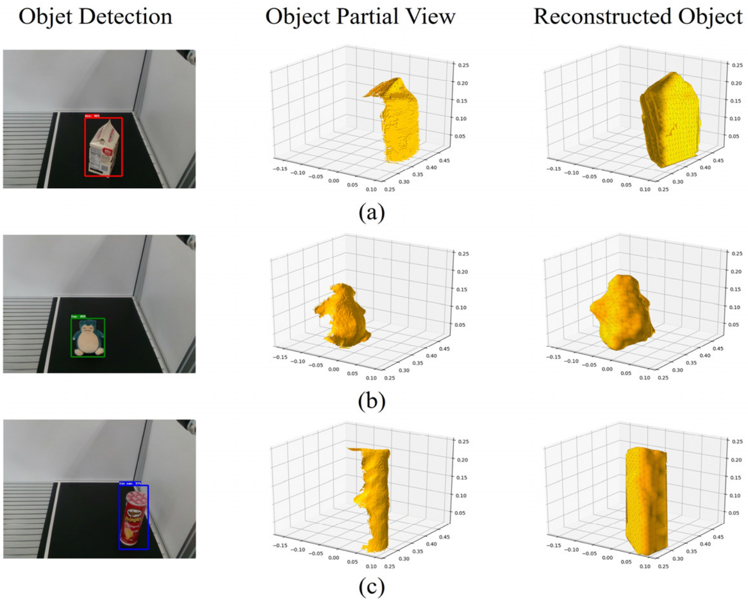
The object manipulation tasks involved the grasping and relocation of twelve objects. The grasping task is defined as approaching the object and lifting it. The relocation task is defined as repositioning the object at the target position without dropping it. The objects manipulated with the proposed autonomous system are a Pringles can, tea box, air can, box, thermos, milk box, tiger, toy, ball, Monster can, flashlight, and bottle. The twelve manipulated objects were not included in the training datasets of 3D U-Net-based BNSC or 3D U-Net-based OGAG. However, the training datasets include some objects similar to the twelve tested objects in their shape and size.

3. Results

This section describes the validation of each stage of the proposed autonomous object manipulation system separately. The results include qualitative and quantitative validation. The qualitative validation shows time series frames from a complete grasping and relocation attempt for some representative objects. Also, pictures from different grasping attempts for three objects are shown. The quantitative validation shows the number of successful grasping and relocation attempts over ten attempts using random initial positions for each object.

3.1. Three-Dimensional Object Shape Reconstruction

The object shape reconstructor is validated qualitatively using samples of some representative objects. Extensive quantitative validation of 3D U-Net-based BNSC is available in [22], where the Jaccard similarity index (i.e., Intersection over Union) was used to assess the reconstruction performance, achieving an average reconstruction performance of 72.17% with the ShapeNet database and 87.03 with the Grasp database. Figure 3 shows the detected objects, sensed partial view of each object, and the reconstructed objects via 3D U-Net-based BNSC. Figure 3 presents the identified objects in the first column, the corresponding sensed partial views in the second column, and the reconstructed objects in the third column. Figure 3a–c depict the reconstruction results for the milk box, toy, and Pringles can, respectively.



(a)

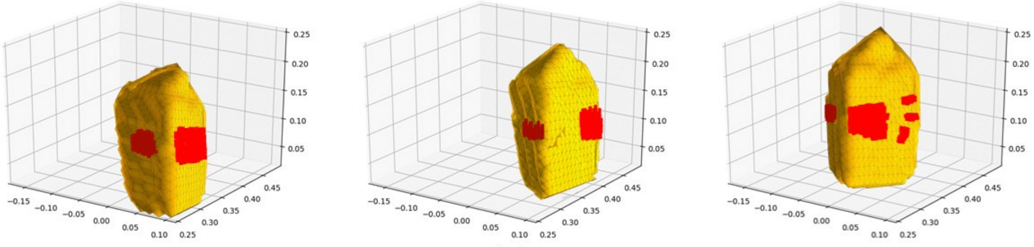
(b)

(c)

Figure 3. Samples of the objects reconstructed with 3D U-Net-based BNSC. Elements (**a**), (**b**), and (**c**) show the reconstruction results for the milk box, toy, and Pringles can, respectively.

3.2. Object Grasping Areas

Figure 4 shows the estimated object grasping areas. Figure 4a shows samples of the object grasping areas generated from the reconstructed milk box. The reconstructed 3D object is shown in yellow. The red points are the generated object grasping areas identified using 3D U-Net-based OGAG. Figures 4b and 4c present samples of the object grasping areas for the reconstructed toy and Pringles can, respectively.



(a)

Figure 4. *Cont.*

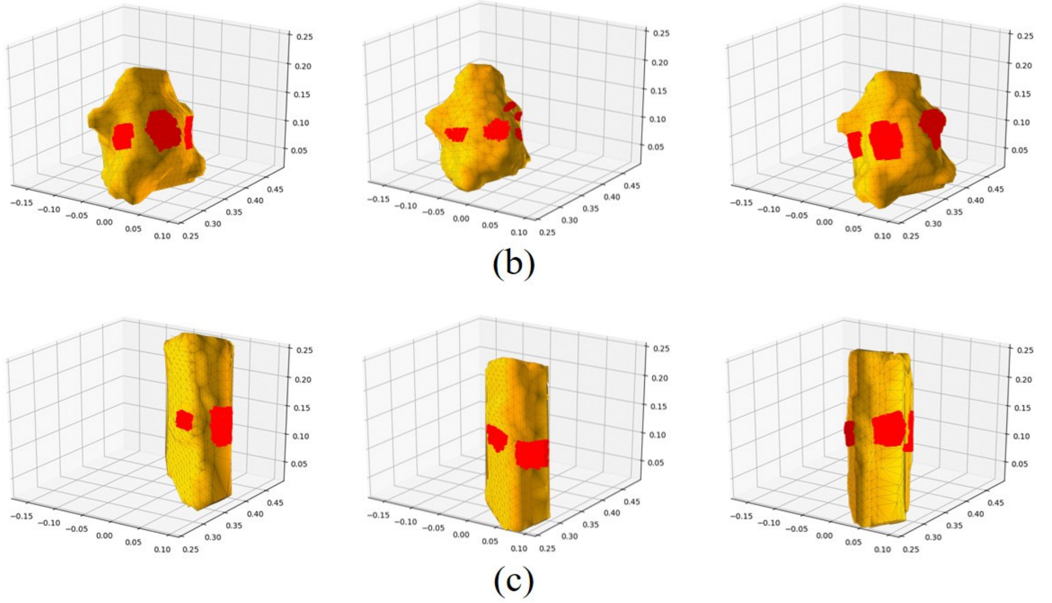


Figure 4. Illustrations of object grasping areas produced by 3D U-Net-based OGAG are presented. In particular, (**a**), (**b**), and (**c**) showcase the outcomes for the reconstructed milk box, toy, and Pringles can, respectively.

3.3. Grasping Orientation Angle

Figure 5 shows samples of the objects' principal components for some representative objects. Figure 5a shows some reconstructed milk boxes and the direction of the principal component as a vector in red. Figure 5b and Figure 5c show the direction of the objects' principal components for the reconstructed toy and Pringles can, respectively.

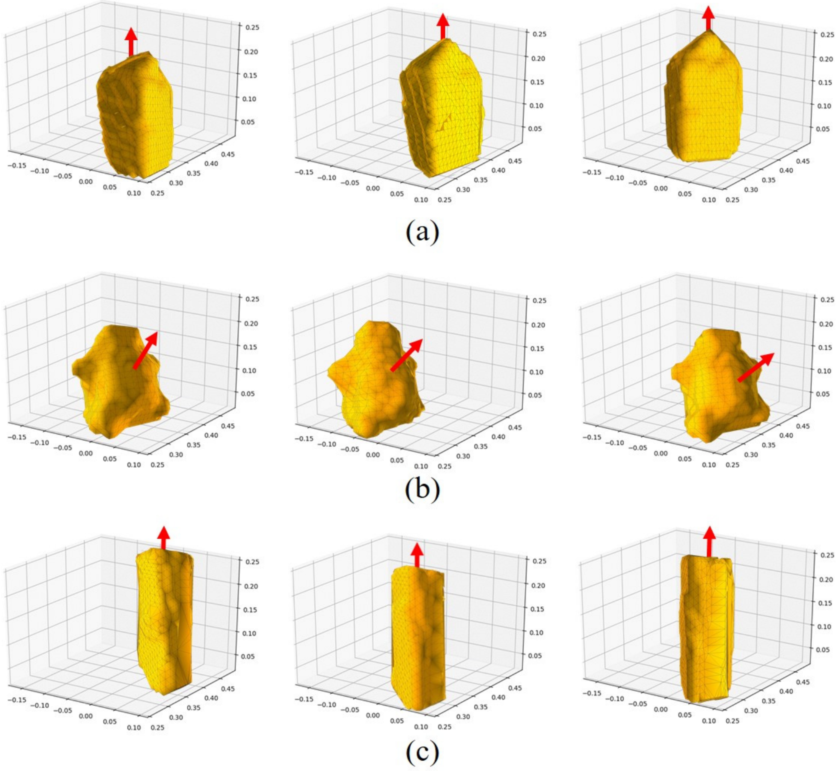


Figure 5. Samples of principal components computed using PCA for the reconstructed milk box (**a**), toy (**b**), and Pringles can (**c**).

3.4. Autonomous Object Grasping and Relocation with Soft Hands

Figure 6 shows time series frames from each attempt at object grasping and relocation for three objects by our proposed autonomous system. Figures 6a and 6b show the soft robot hand approaching the tiger and toy, respectively. Then, the soft hand rotates to grasp the objects and relocate them out of the working space. Figure 6c shows the soft hand grasping and relocating the milk box.

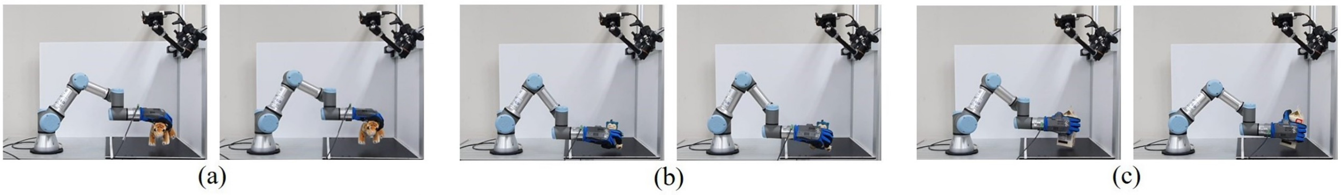


Figure 6. Samples of autonomous grasping of the tiger (**a**), toy (**b**), toy, and milk box (**c**).

Figure 7 shows samples of the autonomous grasping for three representative objects. Figure 7a shows the soft hand grasping the tiger from the top, i.e., a hand orientation angle around 90° (for top grasping). Figure 7b shows the soft hand grasping the toy according to the object's shape, i.e., a hand orientation angle between 0° and 90°. Figure 7c presents the soft hand grasping the milk box from the side of the object, i.e., a hand orientation angle around 0° (for lateral grasping).

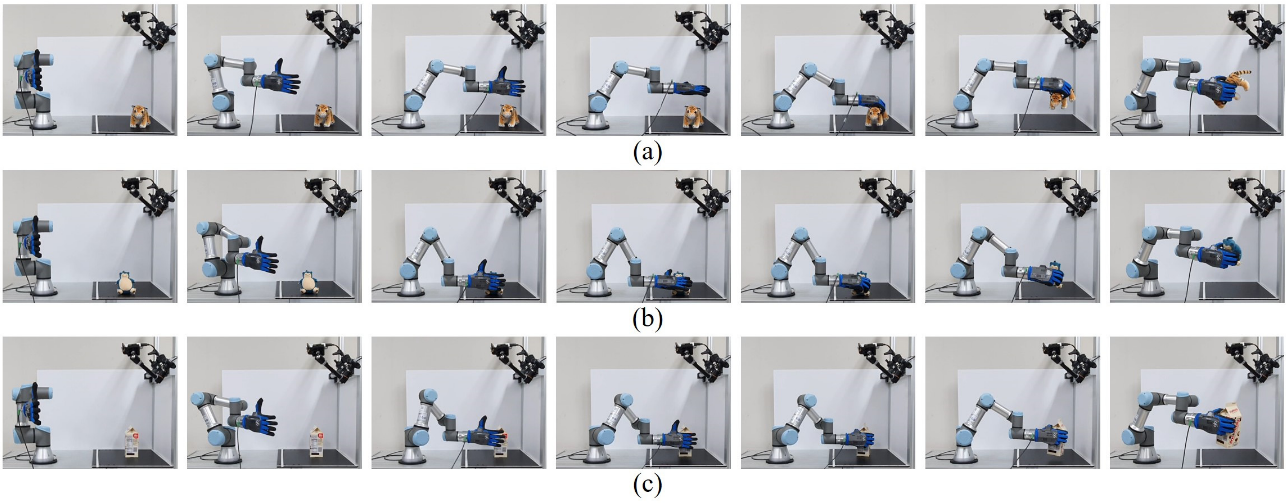


Figure 7. Time series frames of one grasping and relocating attempt by our autonomous object manipulation system for the tiger (**a**), toy (**b**), and milk box (**c**).

Table 1 shows the results of the quantitative validation. For the grasping task, the ball, thermos, flashlight, bottle, tiger, and milk box were successfully grasped ten times over ten attempts. The Pringles can, air can, box, tea box, Monster can, and toy were successfully grasped nine times over ten attempts. For the relocation task, the ball, thermos, flashlight, and bottle were successfully relocated ten times over ten attempts. The Pringles can, air can, box, tea box, Monster can, tiger, toy, and milk box were successfully relocated nine times over ten attempts.

Table 1. Number of successful object manipulations out of ten attempts.

Object	**Pringles Can**	**Ball**	**Air Can**	**Thermos**	**Flashlight**	**Bottle**	**Box**	**Tea Box**	**Monster Can**	**Tiger**	**Toy**	**Milk Box**
Grasping	9	10	9	10	10	10	9	9	9	10	9	10
Relocation	9	10	9	10	10	10	9	9	9	9	9	9

4. Discussion

Three-dimensional object shape reconstruction is one of the crucial components for autonomous object manipulation because it provides complete object information to compute the grasping parameters (i.e., position and orientation). The reconstructed 3D objects' shapes are affected by the detected and segmented partial view of the objects. For instance, the 3D U-Net-based BNSC reconstructs the arms and legs of the toy because the object's partial view includes some information about them, as shown in Figures 3b, 4b and 5b.

The manipulated objects are not included in the training dataset of 3D U-Net-based BNSC, nor for 3D U-Net-based OGAG. The proposed autonomous object manipulation system generates grasping positions and orientation angles suitable for grasping all tested objects based on the knowledge developed with the trained objects. Thus, the DL algorithms estimate grasping parameters for manipulating objects with similar characteristics to those in the training datasets.

Our proposed autonomous object manipulation system is scalable to manipulate variously shaped objects because the continuous orientation angles (i.e., in the range of 0° to 90°) are calculated using PCA. The object manipulation results show the soft robot hand adopting an angle around 90° (for top grasping) to orientate the hand for grasping the tiger, as shown in Figure 6a. The soft hand rotates between 0° and 90° to grasp the toy according to the object shapes (in between top and lateral grasping), as shown in Figure 6b. The soft hand slightly rotates to manipulate the milk box, adopting an angle around 0° (for lateral grasping), as shown in Figure 6c.

Most successful grasping attempts lead to successful relocations for all tested objects because the grasping position and orientation angle are computed according to the reconstructed and complete object shape. Then, the soft robot hand adapts to the object. Only the tiger and milk box slipped out of the robot's hand in one attempt each before finishing the relocation motion.

Previous works performed autonomous object manipulation using different frameworks to compute the grasping parameters and hardware setups [7,25]. For example, Della Santina et al. [7] sensed objects using only an RGB camera (no depth) and grasped them using a Soft Hand. Della Santina's system first detects the object using YOLOv2 from color images and then estimates the grasping parameters via DNN, achieving an average grasping rate of 81.1% with 36 objects. The objects included a mug, salt shaker, bottle, box, glass, book, ball, container, screwdriver, knife, etc. Ficuciello et al. [25] utilized an RGB-D camera and the Schunk hand with 20 DoFs to grasp untrained objects. Ficuciello's system first classified the objects based on criteria of similarity between the sensed point cloud and the spherical or cylindrical shapes using RANSAC models. Then, RL was used to estimate a grasping strategy, achieving an average of 4.6/5 successful grasping attempts with three objects. Ficuciello's system was evaluated using a ball, bottle, and plastic strawberry. Our proposed DL-based vision intelligence uses object detection and 3D shape reconstruction to generate human-like grasping areas and continuous hand-grasping orientation angles. For the manipulation tasks, our system achieved an average success rate of 95% and 93% for the grasping and relocation tasks with the twelve objects, respectively. Despite not being a direct comparison with previous work due to the use of different robot setups and the number of objects, this comparison shows the usefulness of the proposed system.

The average inference time of Faster-RCNN is 5.1 s, that of 3D U-Net-based BNSC is 3.05 s, and that of 3D U-Net-based OGAG is 6.2 s. One of the limitations of our proposed autonomous object manipulation system is that the estimated object grasping areas are not fully utilized due to the hardware limitations of the qb Soft hand; i.e., it does not allow independent control of the fingers. Future work should test our proposed methodology using an anthropomorphic hand with full DoFs for much more delicate object manipulation.

5. Conclusions

In conclusion, our research represents a stride in the realm of robotic visuomotor manipulation. Rather than simply showcasing an automated robotic system for a specific object position or discrete grasping orientation, our work stands out because of the estimated object grasping areas and because the grasping orientation has a continuous range of angles from lateral to top grasping. We introduce a pioneering approach to autonomous visuomotor manipulation by integrating deep learning algorithms. The crux of our innovation lies in the development of a DL-based vision intelligence system, encompassing object detection, 3D shape reconstruction, and grasping area inference. Our vision intelligence employs Faster-RCNN to detect objects randomly positioned in the working space. Subsequently, the 3D U-Net-based BNSC estimates the reconstruction of complete object shapes, providing full object information for effective robot manipulation. Our anthropomorphic soft robot hand's adaptability to diverse objects underscores our approach's versatility. We achieved an average success rate of 95% and 93% for grasping and relocation tasks across twelve different objects not included in the training database, respectively. This underscores the efficacy and practicality of our DL-based vision intelligence in real-world manipulation scenarios.

Our autonomous object manipulation system lays the groundwork for the development of more dexterous anthropomorphic robot systems. In fact, our research marks a pivotal step forward, not merely in showcasing robotic capabilities, but in advancing the theoretical underpinnings of autonomous visuomotor manipulation.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, writing—original draft, E.V.A.; writing—review and editing, formal analysis, S.G. and P.R.L.; data curation, preparation, visualization, J.H.O. and G.H.R.; resources, supervision, project administration, and funding acquisition, T.-S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (NRF-2023R1A2C100358511). This material is based upon work supported by the Ministry of Trade, Industry & Energy (MOTIE, Republic of Korea) under Industrial Technology Innovation Program (RS-2023-00232141).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, X.; Sun, Y.; Zhang, Q.; Liu, F. Two-Stage Grasp Strategy Combining CNN-Based Classification and Adaptive Detection on a Flexible Hand. *Appl. Soft Comput. J.* **2020**, *97*, 106729. [CrossRef]
2. Valarezo-Añazco, E.; Rivera-Lopez, P.; Park, N.; Oh, J.; Ryu, G.; Al-antari, M.; Kim, T.-S. Natural Object Manipulation Using Anthropomorphic Robotic Hand Through Deep Reinforcement Learning and Deep Grasping Probability Network. *Appl. Intell.* **2021**, *51*, 1041–1055. [CrossRef]
3. Valarezo-Añazco, E.; Rivera Lopez, P.; Park, H.; Pak, N.; Oh, J.; Lee, S.; Byun, K.; Kim, T.-S. Human-like object grasping and relocation for an anthropomorphic robotic hand with natural hand pose priors in deep reinforcement learning. In Proceedings of the International Conference on Robotics and Computer Vision (ICRCV 2019), Bangkok, Thailand, 4–7 August 2019; pp. 46–50.
4. Billard, A.; Kragic, D. Trends and Challenges in Robot Manipulation. *Science* **2019**, *364*, eaat8414. [CrossRef] [PubMed]
5. Birglen, L.; Laliberte, T.; Gosselin, C.M. *Underactuated Robotic Hands*; Springer: Berlin/Heidelberg, Germany, 2007.
6. Haas, M.; Friedl, W.; Stillfried, G.; Hoppner, H. Human-Robotic Variable-Stiffness Grasps of Small-Fruit Containers Are Successful Even Under Severely Impaired Sensory Feedback. *Front. Neurorobotics* **2018**, *12*, 70. [CrossRef]
7. Della Santina, C.; Arapi, V.; Averta, G.; Damiani, F.; Fiore, G.; Settimi, A.; Catalano, M.; Bacciu, D.; Bicchi, A.; Bianchi, M. Learning From Humans How to Grasp: A Data-Driven Architecture for Autonomous Grasping With Anthropomorphic Soft Hands. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1533–1540. [CrossRef]
8. Collet, A.; Martinez, M.; Srinivasa, S.S. The Moped Framework: Object Recognition and Pose Estimation for Manipulation. *Int. J. Robot. Res.* **2011**, *30*, 1284–1306. [CrossRef]
9. Wang, C.; Xu, D.; Zhu, Y.; Martin-Martin, R.; Lu, C.; Fei-Fei, L.; Savarese, S. DenseFusion: 6D Object Pose Estimation by Iterative Dense Fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3338–3347.
10. Fang, K.; Zhu, Y.; Garg, A.; Kurenkov, A.; Mehta, V.; Fei-Fei, L.; Savarese, S. Learning Task-Oriented Grasping for Tool Manipulation from Simulated Self-Supervision. *Int. J. Robot. Res.* **2019**, *39*, 202–216. [CrossRef]

11. Gupta, A.; Eppner, C.; Levine, S.; Abbeel, P. Learning Dexterous Manipulation for a Soft Robotic Hand from Human Demonstrations. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 3786–3793.
12. Bullock, I.; Ma, R.; Dollar, A. A Hand-Centric Classification of Human and Robot Dexterous Manipulation. *IEEE Trans. Haptics* **2013**, *6*, 129–144. [CrossRef] [PubMed]
13. Mohammed, A.-M.; Mugahed, A.-A.; Jeong-Min, P.; Geon, G.; Tae-Yeon, K.; Rivera, P.; Valarezo, E.; Choi, M.-T.; Seung-Moo, M.; Kim, T.-S. Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system. *Comput. Methods Programs Biomed.* **2018**, *157*, 85–94.
14. Valarezo-Añazco, E.; Rivera Lopez, P.; Park, N.; Oh, J.; Ryu, G.; Kim, T.-S. Fully Autonomous Object Grasping and Relocation System with Anthropomorphic Robotic Hands. In Proceedings of the Korea Communication Society Winter Symposium, Republic of Korea; 2021.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 91–99.
16. Wang, R.; Qin, Y.; Wang, Z.; Zheng, H. Group-Based Sparse Representation for Compressed Sensing Image Reconstruction with Joint Regularization. *Electronics* **2022**, *11*, 182. [CrossRef]
17. Xie, X.; Guo, X.; Li, W.; Liu, J.; Xu, J. Deform2NeRF: Non-Rigid Deformation and 2D–3D Feature Fusion with Cross-Attention for Dynamic Human Reconstruction. *Electronics* **2023**, *12*, 4382. [CrossRef]
18. Sipiran, I.; Gregor, R.; Schreck, T. Approximate Symmetry Detection in Partial 3D Meshes. *Comput. Graph. Forum* **2014**, *33*, 131–140. [CrossRef]
19. Miltra, N.; Guibas, L.; Pauly, M. Partial and Approximate Symmetry Detection for 3D Geometry. *ACM Trans. Graph. TOG* **2006**, *25*, 560–568. [CrossRef]
20. Rothganger, F.; Lazebnik, S.; Schmid, C.; Ponce, J. 3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints. *Int. J. Comput. Vis.* **2006**, *66*, 231–259. [CrossRef]
21. Deng, X.; Qiu, S.; Jin, W.; Xue, J. Three-Dimensional Reconstruction Method for Bionic Compound-Eye System Based on MVSNet Network. *Electronics* **2022**, *11*, 1790. [CrossRef]
22. Valarezo-Añazco, E.; Rivera, L.; Kim, T.-S. Three-dimensional Shape Reconstruction of Objects from a Single Depth View Using Deep U-Net CNN with Bottle-neck Skip Connections. *IET Comput. Vis.* **2021**, *15*, 24–35. [CrossRef]
23. Brahmbhatt, S.; Ham, C.; Kemp, C.; Hays, J. ContactDB: Analyzing and Predicting Grasp Contact via Thermal Imaging. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 8701–8711.
24. Choi, C.; Schwarting, W.; DelPreto, J.; Rus, D. Learning Object Grasping for Soft Robot Hands. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2370–2377. [CrossRef]
25. Ficuciello, F.; Migiozzi, A.; Laudante, G.; Falco, P.; Siciliano, B. Vision-Basedgrasp Learning of an Anthropomorphic Hand-Arm System in a Synergy-Based Control Framework. *Sci. Robot.* **2019**, *4*, eaao4900. [CrossRef]
26. Qb Robotics C++ Library. Available online: https://qbrobotics.com/ (accessed on 15 December 2020).
27. Krasin, I.; Duerig, T.; Alldrin, N.; Ferrari, V.; Abu-El-Haija, S.; Kuznetsova, A.; Rom, H.; Uijlings, J.; Popov, S.; Kamali, S.; et al. OpenImages: A Public Dataset for Large-Scale Multi-Label and Multi-Class Image Classification. Available online: https://storage.googleapis.com/openimages/web/index.html (accessed on 15 December 2020).
28. Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Duerig, T.; et al. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *Int. J. Comput. Vis.* **2020**, *128*, 1956–1981. [CrossRef]
29. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S.; et al. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3296–3297.
30. Kappler, D.; Bohg, J.; Schaal, S. Leveraging big data for grasp planning. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 4304–4311.
31. Binvox, a 3D Mesh Voxelizer. Available online: https://www.patrickmin.com/binvox/ (accessed on 15 December 2020).
32. Nooruddin, F.S.; Turk, G. Simplification and Repair of Polygonal Models Using Volumetric Techniques. *IEEE Trans. Vis. Comput. Graph.* **2003**, *9*, 191–205. [CrossRef]
33. Varley, J.; Dechant, C.; Richardson, A.; Ruales, J.; Allen, P. Shape completion enabled robotic grasping. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2442–2447.
34. Koenig, N.; Howard, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004; pp. 2149–2154.
35. Ji, S.; Zhang, C.; Xu, A.; Shi, Y.; Duan, Y. 3D Convolutional Neural Networks for Crop Classification with Multi-Temporal Remote Sensing Images. *Remote Sens.* **2018**, *10*, 75. [CrossRef]

36. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
37. Brahmbhatt, S.; Handa, A.; Hays, J.; Fox, D. ContactGrasp: Functional Multi-finger Grasp Synthesis from Contact. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019.
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
39. Heravi, E.J.; Aghdam, H.H.; Puig, D. An Optimized Convolutional Neural Network with Bottleneck and Spatial Pyramid pooling layers for Classification of Foods. *Pattern Recognit. Lett.* **2018**, *105*, 50–58. [CrossRef]
40. Intel RealSense Python Library. Available online: https://github.com/IntelRealSense/librealsense (accessed on 15 December 2020).
41. URX Python Library. Available online: https://github.com/SintefManufacturing/python-urx (accessed on 15 December 2020).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.