Depth-aware Object Segmentation and Grasp Detection for Robotic Picking Tasks

Stefan Ainetter[1]
stefan.ainetter@icg.tugraz.at

Christoph Böhm[2]
christoph.boehm@ieee.org

Rohit Dhakate[2]
rohit.dhakate@ieee.org

Stephan Weiss[2]
stephan.weiss@ieee.org

Friedrich Fraundorfer[1]
fraundorfer@icg.tugraz.at

[1] Institute of Computer Graphics & Vision
Graz University of Technology
Graz, Austria

[2] Institute of Smart System Technologies
University of Klagenfurt
Klagenfurt, Austria

Abstract

In this paper, we present a novel deep neural network architecture for joint class-agnostic object segmentation and grasp detection for robotic picking tasks using a parallel-plate gripper. We introduce depth-aware Coordinate Convolution (CoordConv), a method to increase accuracy for point proposal based object instance segmentation in complex scenes without adding any additional network parameters or computation complexity. Depth-aware CoordConv uses depth data to extract prior information about the location of an object to achieve highly accurate object instance segmentation. These resulting segmentation masks, combined with predicted grasp candidates, lead to a complete scene description for grasping using a parallel-plate gripper. We evaluate the accuracy of grasp detection and instance segmentation on challenging robotic picking datasets, namely Siléane and OCID_grasp, and show the benefit of joint grasp detection and segmentation on a real-world robotic picking task.

1 Introduction

Automated grasping in challenging environments is one of the most active fields of research in robotics. In order to successfully grasp and move objects, accurate information about potential grasp candidates, which are areas where the gripper can grasp the object, as well as shape and location of these objects are highly important. Previous works used deep neural networks to predict highly accurate grasp candidates for parallel-plate grippers with one or multiple objects in the scene. Recent works focused on additional object detection [26, 36, 37] or semantic segmentation [1]. This additional information is used to identify and pick specific objects or a class of objects, and to understand the relation between objects in a cluttered scene.

Other researchers focused on class-agnostic object instance segmentation [15, 35], arguing that pixel-accurate instance segmentation of unknown objects is highly important for robotic picking, as it helps to understand each object's shape and location accurately and can be

used to avoid potential collisions during grasping (as shown in [25]). Although these methods provide highly accurate results for instance segmentation, they do not directly provide grasp candidates.

In this work, we address joint grasp detection for parallel-plate grippers as well as class-agnostic instance segmentation, by proposing an end-to-end trainable multi-task deep neural network. Our network provides high quality results for grasp detection and pixel-wise instance segmentation for multiple graspable objects in complex scenes, where objects highly overlap. We achieve this, by introducing our novel depth-aware CoordConv module. Depth-aware CoordConv enables us to generate additional feature maps which encode a location prior related to the searched object instance, which are used to predict accurate object segmentation masks. We evaluate our method on publicly available datasets for robotic manipulations with high complexity. Additionally, we demonstrate our method on a real-world robotic picking task, and show simple ways to use the predicted instance segmentation to enhance grasp accuracy. The main contributions of our paper are the following:

1. An end-to-end trainable deep neural network architecture for joint grasp detection and class-agnostic instance segmentation, which yields state-of-the-art performance for grasp detection and provides valuable information for scene understanding in robotic picking tasks.

2. Depth-aware CoordConv, a novel method to increase class-agnostic instance segmentation accuracy in highly challenging scenes with multiple overlapping objects. Notably, it does not introduce additional network parameters or computation complexity.

3. Extensive experiments including public grasp datasets and real-world experiments, to show the benefit of using joint grasp detection and instance segmentation in scenes with high clutter and occlusions. By leveraging the predicted segmentation masks we show how to optimize the picking and placing of target objects in a real environment.

2 Related Work

Grasping is a widely studied field in robotics [4, 5, 7]. A popular approach to solve it is Reinforcement Learning [16, 17, 29], which uses feedback of grasp attempts for learning in real or simulated environments. Regarding supervised learning approaches, we can distinguish between planar and spatial grasp detection. Methods for spatial grasp detection [9, 24, 25] usually predict 6-DOF grasp poses, whereas planar grasp detection represents grasp candidates as oriented rectangles or points in the image. In the following, we will focus our comparison on methods similar to ours using planar grasp detection.

Planar Grasp Detection. First approaches using deep neural networks and supervised learning to predict multiple grasp candidates for a single object are [19, 30]. Latest research focuses on detection of grasp candidates for multiple objects [3, 10, 18, 23, 27, 33]. The works of [36, 37] perform grasp detection with additional object detection to identify the relationship between multiple objects in complex scenes.

Object segmentation for robotic picking tasks. The authors of [35] provide a method for unseen object instance segmentation for robotic environments by using an encoder/decoder CNN architecture with clustering based on Hough voting. The authors of [8] predict class-agnostic segmentation of objects from depth images using a variant of Mask R-CNN [12]. However, region proposal based segmentation methods like Mask R-CNN often fail to perform well in complex scenes where objects highly overlap, which is often the case for robotic picking. [15] proposed a method for point proposal based instance segmentation for robotic

picking tasks. To generate these point proposals, [15] added specific network branches, which introduce additional overhead in terms of network parameters and computation cost. Our method, which also relies on point proposals for instance segmentation, compensates this disadvantage by directly utilizing predicted grasp candidates as point proposals for instance segmentation.

Joint grasp detection and segmentation. The authors of [2] proposed a network for semantic segmentation, and grasp detection represented as points for a suction cup manipulator using multi-task learning with a single deep neural network. Other recent work [1] proposed a method for joint grasp detection and class-specific semantic segmentation. However, semantic segmentation is not sufficient to distinguish between multiple instances of the same object class.

3   Problem Statement

We propose an end-to-end trainable pipeline to detect reliable grasp candidates for robotic grasping, by combining grasp detection with semantic and class-agnostic instance segmentation. A brief introduction to the tasks of grasp detection and segmentation is given below:

Grasp detection. We use the five-dimensional bounding box representation [19] to define grasp candidates for robotic picking. A grasp candidate **g** is defined as

$$\mathbf{g} = (x, y, w, h, \theta), \tag{1}$$

where (x,y) define the center of the grasp candidate, w and h define the width and height of the bounding box, and θ defines the orientation of the grasp candidate.

Semantic segmentation. Semantic segmentation is the task of classifying each pixel of an input image into one of the predefined class labels. For grasp detection of unknown objects, we define the set of semantic class labels as $l \in \{1,...,N\}$, whereas $N = 2$ is the number of classes for foreground/background segmentation.

Class-agnostic instance segmentation. Class-agnostic (or category-agnostic) instance segmentation is the task of predicting pixel-wise segmentation masks for each object instance in the scene. Class-agnostic means that no information about object classes is available, for example the objects are either not seen or not known during training.

4   Architecture Overview

Our proposed architecture for joint grasp detection and segmentation consists of a shared backbone for feature extraction combined with task-specific branches for grasp detection, semantic, and instance segmentation. A ResNet-101 [11] with Feature Pyramid Network (FPN) [20] is used to extract feature maps from an RGB input image, which are then used as input for joint grasp detection and segmentation. Figure 1 shows an overview of our network architecture. Hereafter, we describe each component of our network in detail.

4.1   Grasp Detection Branch

The grasp detection branch performs the task of predicting grasp candidates for multiple graspable objects in the scene. To achieve this, we use a region proposal based grasp detection network, similar to [1]. The grasp detection head takes previously extracted feature maps as input and predicts grasp candidates in form of axis-aligned rectangular boxes and the rotation parameter θ. Additionally, it predicts a confidence score $s \in [0, 1]$, whereas higher
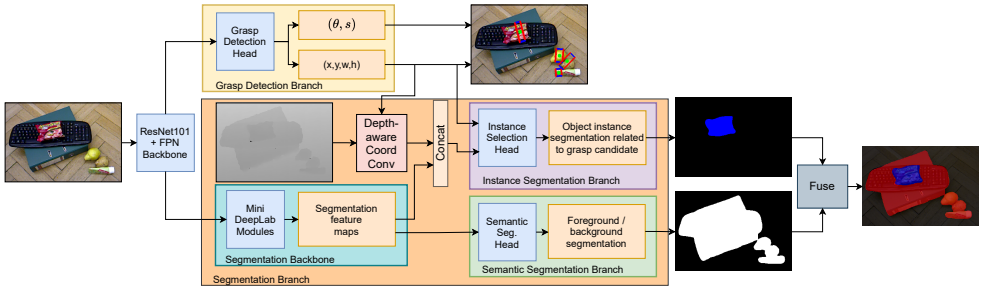
Figure 1: Architecture of our proposed method. Both branches for grasp detection and segmentation share the backbone network as feature extractor. The output of our network consists of grasp candidates for multiple graspable objects in the scene (green dot denotes center of grasp candidate, red lines denote opening width of gripper, blue lines denote parallel plates). Additionally, the network predicts accurate object instance seg. masks for one or multiple graspable objects and fore-/background segmentation. The final segmentation result shows the object instance mask related to the grasp candidate **g** with the highest confidence score *s* which is highlighted in blue, and all other objects in the scene (which can be seen as obstacles) are red.

values indicate the network is more confident that the grasp candidate is valid. The final result is a number of grasp candidates **g** (see Eq. 1) and corresponding confidence scores **s**.

4.2   Segmentation Branch

The segmentation branch consists of a segmentation backbone, and sub-branches for semantic and instance segmentation. We will now describe all parts in detail.

4.2.1   Segmentation Backbone

The segmentation backbone takes as input a subset of the extracted feature maps from the ResNet101 + FPN backbone network, which correspond to the first four scales of the FPN. Mini-DeepLab modules [28] are then applied independently to each input. The output of each module consists of 128 feature maps which are down-sampled by the factor of x4 compared to the input image size. The feature maps of all modules are concatenated and the resulting final 512 segmentation feature maps are used as input for both the semantic and instance segmentation branch.

4.2.2   Semantic Segmentation Branch

The semantic segmentation branch assigns a class label $l \in \{1, ..., N\}$ to each pixel of the input image, whereas $N = 2$ is the number of classes for foreground/background segmentation. Note that if class information is available, the set of semantic classes can be adjusted accordingly. The semantic segmentation head takes as input the feature maps from the segmentation backbone, which are fed to a 1x1 convolution layer with N output channels. The resulting logits are used as input for a final softmax layer which provides class probabilities for each pixel of the input image. At the end, the probability maps are bilinearly upsampled to the size of the input image which results in the final foreground/background segmentation.

4.2.3   Instance Segmentation Branch

Our instance segmentation branch is based on two main components, depth-aware Coord-Conv and the instance selection head. Both components take a point proposal $p(x,y)$ as input, which is used to select a specific object for instance segmentation. Our full pipeline

(shown in Figure 1) utilizes the center of a predicted grasp candidate **g** as point proposal. Note that utilizing information about our predicted grasp candidates provides a clear advantage to other point proposal based instance segmentation methods, e.g. [15, 32], which need an additional module (e.g. an additional neural network branch) to predict point proposals.

Depth-aware CoordConv calculates feature maps to encode a location prior for a specific object based on $p(x,y)$. The instance selection head performs the task of predicting an instance segmentation mask related to the point proposal $p(x,y)$. A detailed description about both parts follows below:

Depth-aware CoordConv. It is based on the idea of coordinate convolution [21], which is a mechanism to make a convolutional neural network aware of input coordinates. This counteracts the fact that these networks have the property of translation invariance, producing similar outputs for similar objects at different locations.

Depth-aware CoordConv generates several feature maps that encode a positional prior for a specific object instance. This helps to distinguish multiple object instances in highly stacked scenes, which is often the case for robotic picking tasks. First, we calculate $\mathbf{X}_{rel}$ and $\mathbf{Y}_{rel}$, which encode the relative coordinates (rel. CoordConv) for the point proposal $p(x,y)$ in x and y direction, respectively. Both $\mathbf{X}_{rel}$ and $\mathbf{Y}_{rel}$ are then divided by a hyperparameter R, which roughly sets the maximum object size. Similar methods have already been used in the literature, e.g. in [32]. Additionally, we define the depth-distance map $\mathbf{D}_{dist} = \alpha \cdot (\mathbf{D} - \mathbf{D}(p))$ with $\mathbf{D}_{dist} \in [-\alpha, \alpha]$, whereas $\mathbf{D} \in [0,1]$ is the normalized depth input and α is a hyperparameter used as scaling factor. At last, we calculate a 2.5D distance map using $\mathbf{F}_{2.5D} = \sqrt{\mathbf{X}_{rel}^2 + \mathbf{Y}_{rel}^2 + \mathbf{D}_{dist}^2}$. All calculated feature maps from depth-aware Coord-Conv are then clamped to $(-1,1)$. Figure 2 shows an example of these final feature maps. The output of depth-aware CoordConv is concatenated with the feature maps produced by the segmentation backbone, and together they are passed to the instance selection head. Note that this usage of depth information does not add any network parameters or computational complexity (compared to other methods that use multiple input modalities to improve performance, e.g. [22, 34]), and is therefore a simple and effective way to utilize depth information.
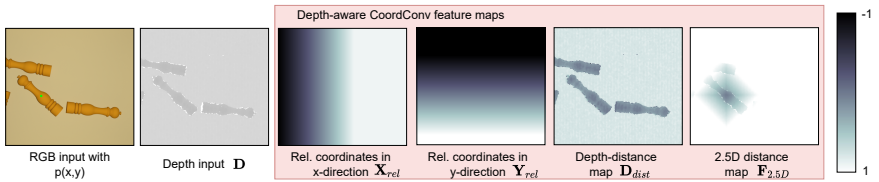

Figure 2: Visualizations of feature maps calculated using depth-aware CoordConv. The two left images show an RGB input with the point proposal $p(x,y)$ marked as green dot, and the depth input D, respectively. Additionally, the corresponding feature maps generated using depth-aware CoordConv are shown. One can see that these feature maps encode a location prior for the selected object.

Instance selection head. The instance selection head is based on [32] and takes as input the segmentation and depth-aware feature maps, as well as the point proposal $p(x,y)$. First, the number of seg. feature maps is reduced by the factor of 2 using a 1x1 convolutional layer. Then, this feature maps are passed to three convolutional layers, followed by an AdaIN layer [13] and again two convolutional layers whereas the last one performs 1x1 convolution. Note that the AdaIN layer is used to guide the instance selection head to segment the specific object selected using the point proposal $p(x,y)$. This is done by extracting a descriptive feature

vector from the segmentation features at the location $p(x, y)$ using a few fully connected layers, which is then passed to the AdaIN layer. Each convolutional layer in the instance selection head is followed by ReLU activation and Batch Normalization [14], except the 1x1 convolutional layers. The resulting logits are bilinearly upsampled to the size of the input image and then used as input for a softmax function which provides probabilities that a pixel belongs to the selected object instance.

4.3   Loss Function

The loss function for joint grasp detection, semantic and instance segmentation, is defined as $\mathcal{L} = \lambda_{grasp}\mathcal{L}_{grasp} + \lambda_{sem}\mathcal{L}_{sem} + \lambda_{inst}\mathcal{L}_{inst}$, with the grasp detection loss $\mathcal{L}_{grasp}$, the semantic segmentation loss $\mathcal{L}_{sem}$, and the instance selection loss $\mathcal{L}_{inst}$. The grasp detection loss $\mathcal{L}_{grasp}$, based on [1], is a combination of a regression loss for predicting axis-aligned bounding box parameters (x, y, w, h), and a classification loss for predicting the orientation parameter θ. For $\mathcal{L}_{sem}$ we used a weighted per-pixel log loss similar to [28]. As instance selection loss $\mathcal{L}_{inst}$ we used a Normalized Focal Loss as described in [32]. The specific hyperparameters λ are used for loss balancing. Note that the grasp detection branch and segmentation branch are supported with distinct losses, and depth-aware CoordConv influences the grasp detection only indirectly through the shared backbone. We refer to the supplementary material for detailed information about the loss function.

5   Experiments and Results

To assess the benefits of our proposed method, we divided the evaluation into three parts: 1) evaluation of our instance segmentation method focusing on the benefit of depth-aware CoordConv, 2) evaluation of joint grasp detection and object segmentation on the challenging OCID_grasp dataset [1], and 3) real-world robotic grasp experiments with the focus on using instance segmentation masks to optimize picking and placing of objects. Note that for all experiments, we initialize the backbone network with pre-trained ImageNet [31] weights and freeze the parameters of the first two backbone network modules.

Data Augmentation. Throughout all training runs, we use the following method for data augmentation. For each RGB-D input image, we apply random rotation between $0°$ and $360°$, and randomly translate the input data in x and y direction independently up to $50px$.

Evaluation metrics. The grasp accuracy is measured using the Jaccard index, with the following criteria needed to be true for a valid grasp candidate:

1. The angle difference between predicted grasp candidate $\mathbf{g}_p$ and ground truth grasp candidate $\mathbf{g}_{gt}$ is within $30°$ and

2. the Intersection over Union (IoU) of them is greater than 0.25.

For instance and semantic segmentation, we calculate the mean IoU between predicted and ground truth segmentation.

5.1   Depth-aware Instance Segmentation in Complex Scenes

To evaluate the benefit of depth-aware CoordConv for object instance segmentation, we used the synthetic part of the Siléane dataset [6] (excluding images with no objects), which consists of 1594 RGB-D images depicting various numbers of object instances in bulk, with pixel-wise annotated segmentation masks. Figure 3 shows examples for RGB and depth data.

Dataset preparation. First, we center crop each image to a size of $400 \times 400px$. During

training, we randomly sampled point proposals $p(x,y)$ on random object instances, and predicted the corresponding instance segmentation. During testing, we used ground truth point proposals which were sampled in advance, to ensure fair comparison throughout our experiments. We performed an 80:20 image-wise split into training and test set, using 80% of data for training and 20% for testing. The image-wise split means that images are separated randomly without considering which objects are in the scene.

Training procedure. The network for this experiment consists of a ResNet-101 + FPN backbone and the segmentation branch (without the grasp detection branch, see Figure 1). We used SGD as optimizer, with a learning rate of 0.02 and a weight decay of 0.0001. We used a batch size of 4 and randomly sampled 9 point proposals for each image. For loss balancing, we chose weighting factors $\lambda_{sem} = 1.0$ and $\lambda_{inst} = 1.0$.

Quantitative evaluation. Table 1 shows the influence of feature maps generated by depth-aware CoordConv. As previously shown in [52], we can confirm that using rel. Coord-Conv feature maps as location prior increase segmentation accuracy. Furthermore, the results show that our depth-aware CoordConv significantly improves instance segmentation accuracy for complex scenes. We used the same network architecture, hyperparameters and training schedule for all experiments. Therefore, the improvement for instance segmentation is directly related to depth-aware CoordConv. Additional experiments with different variants of depth-aware CoordConv are provided in the supplementary material.

Network Architecture	Rel. CoordConv [52]	Depth-dist. Map (ours)	2.5D Dist. Map (ours)	Inst. Seg. IOU (%)
Backbone + Seg. Branch	✗	✗	✗	83.01
Backbone + Seg. Branch	✓	✗	✗	85.63
Backbone + Seg. Branch	✓	✓	✓	**91.27**

Table 1: Comparison of different configurations of CoordConv for instance segmentation. The results show that using all feature maps generated by our depth-aware CoordConv module significantly improves segmentation accuracy in complex scenes.

Qualitative evaluation. Figure 3 shows qualitative results for object instance segmentation with and without depth-aware CoordConv. The results show that using our method significantly improves instance segmentation in scenes where multiple objects highly overlap.

5.2 Joint Grasp Detection and Object Segmentation

The recently proposed OCID_grasp [1], a dataset extension for OCID [53], contains RGB-D images of diverse settings of objects and backgrounds, with hand annotated grasp candidates and object instance segmentation masks as ground truth. Figure 4 a) shows sample RGB images of the dataset. To highlight the effect of depth-aware CoordConv, we evaluated the full network architecture for joint grasp detection and segmentation (as proposed in Figure 1) with and without depth-aware CoordConv.

Training schedule. The network was trained end-to-end using a learning rate of 0.02 with weight decay of 0.0001, a momentum factor of 0.9 with enabled nesterov momentum and SGD as optimizer. We used a batch size of 3 during training and weighting factors of $\lambda_{grasp} = 1.0$, $\lambda_{sem} = 0.7$ and $\lambda_{inst} = 0.7$ for loss balancing. We performed an 80:20 image-wise split into training and test set, using 80% of data for training and 20% for testing. During training, we randomly sampled point proposals for object instance segmentation, during evaluation, we used the center of the grasp candidates **g** as point proposals $p(x,y)$.

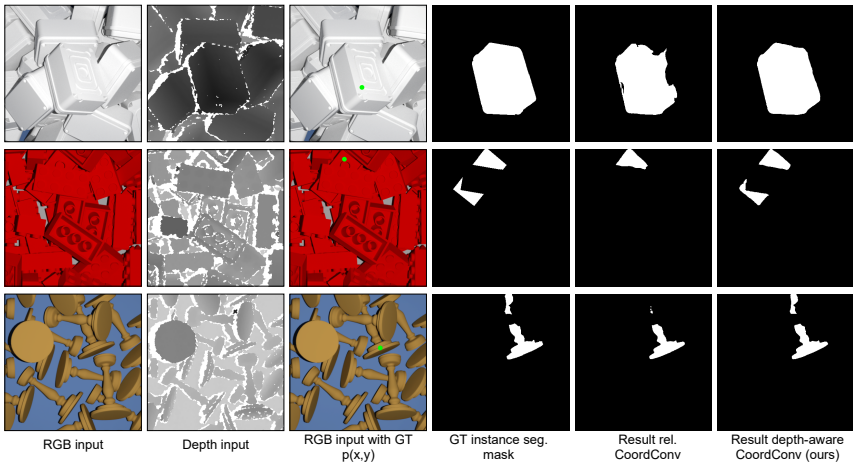| RGB input | Depth input | RGB input with GT p(x,y) | GT instance seg. mask | Result rel. CoordConv | Result depth-aware CoordConv (ours) |

Figure 3: Qualitative results for instance segmentation on Siléane dataset. Note that the point proposals $p(x,y)$ in this experiment are given as ground truth. The resulting instance segmentation masks are only shown for one point proposal to ensure clarity. By comparing the results of rel. CoordConv [32] and depth-aware CoordConv (ours), one can see that our method increases segmentation accuracy especially for highly occluded objects.

Quantitative & qualitative results. Table 2 shows quantitative results of our method for grasp accuracy and segmentation. The grasp accuracy is calculated using the Jaccard index, the instance seg. accuracy is reported using the mean IOU, both is done for all detected graspable objects in the scene. Foreground / background segmentation is evaluated by calculating the mean IOU of predicted and ground truth segmentation. To evaluate the effect of depth-aware CoordConv, we trained our method with and without using these additional feature maps. By comparing the results, one can see that using depth-aware CoordConv leads to an improved overall performance, especially for the task of instance segmentation. Using depth-aware CoordConv also has a positive effect on grasp accuracy (as the grasp accuracy is indirectly influenced by depth-aware CoordConv through the shared backbone).

Although not directly comparable, we report the results of our quantitative evaluation together with the results of [1], to emphasize the state-of-the-art accuracy of our method. The main difference is that [1] uses class information for their prediction, whereas our method has the advantage that it is class-agnostic and does not need additional information. Furthermore, as [1] only predicts sem. segmentation, they are not able to distinguish between instances of the same object class. Additional visualizations to highlight this difference are shown in the supplementary material. Figure 4 b) - d) show grasp candidates for multiple graspable objects in the scene, and the corresponding instance segmentation and fore-/background segmentation. Figure 4 e) - f) show grasp candidate and corresponding segmentation for one graspable object. This leads to a pixel-wise classification into background, obstacle, and graspable object providing all necessary information for robotic picking.

5.3   Real-World Robotic Picking

Model description and setup. To test the accuracy of our proposed architecture for picking objects in practice, we used a hydraulic actuated crane model build for picking logs in the wood sector. This crane is a 1:5 scaled version of a real forestry crane, and the hydraulics are specifically designed to match this scale. Figure 5 a) and b) show the 1:5 scaled crane model and gripper, respectively. A ceiling mounted ZED camera captures RGB-D images.

Method	Class-specific		Class-agnostic		
	Grasp Accuracy	Sem. Seg. IOU	Grasp Accuracy	Fore-/Background Seg. IOU	Inst. Seg. IOU
Grasp Detection & Sem. Seg. [□]	89.02	94.05	-	-	-
(ours) without depth-aware CC	-	-	95.95	96.65	92.37
(ours) with depth-aware CC	-	-	**97.09**	**97.16**	**94.55**

Table 2: Results for grasp accuracy, semantic segmentation and instance segmentation for OCID_grasp in (%). Note that the method from [□] and (ours) are not comparable, as [□] uses class-specific grasp detection and segmentation. Experiments show that using depth-aware CoordConv (CC) improves overall performance of our network, especially for inst. segmentation.
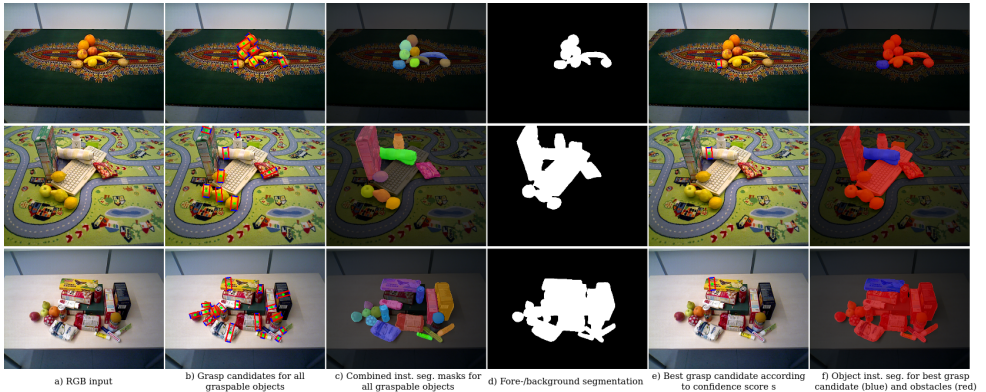

| a) RGB input | b) Grasp candidates for all graspable objects | c) Combined inst. seg. masks for all graspable objects | d) Fore-/background segmentation | e) Best grasp candidate according to confidence score s | f) Object inst. seg. for best grasp candidate (blue) and obstacles (red) |

Figure 4: Qualitative results for joint grasp detection and segmentation on OCID_grasp (using *(ours) with depth-aware CC*). Results in b) - d) are used for quantitative evaluation in Table 2. Note that our method is able to accurately detect grasp candidates and corresponding inst. seg. masks for multiple graspable objects in the scene. Each color in c) represents a independent object instance. As one can see in e) and f), for real-world robotic grasping of unknown objects, we are primarily interested in one valid grasp candidate, and a segmentation which defines the graspable object and all other objects in the scene (which are in this case obstacles).

Data preparation. We used scaled versions of real tree logs as objects in our experiments. Figure 5 c) shows an example of the RGB input data. Note that although the objects have no complex shape (unlike the objects in the previous experiments), this experiment serves as demonstration of how to use our method to achieve high grasp success rate.

Training details. We manually annotated 266 images (similar to the one in Figure 5 c)) with ground truth grasp candidates and segmentation masks. Extensive data augmentation ensured that no overfitting during training occurred. The network was trained end-to-end using a learning rate of 0.005 with weight decay of 0.0001, a momentum factor of 0.9 with enabled nesterov momentum and SGD as optimizer. We used a batch size of 2 and weighting factors of $\lambda_{grasp} = 1.0$, $\lambda_{sem} = 0.8$ and $\lambda_{inst} = 0.8$ for loss balancing. During training, we randomly sampled point proposals for object instance segmentation, during test runs, we

used the center of the grasp candidates **g** as point proposals $p(x,y)$.

Using segmentation results to increase grasp success rate for picking and placing logs.
This robotic picking experiment is used to demonstrate simple ways to utilize the predicted
segmentation to achieve a high grasp success rate. This is done by implementing three
segmentation-based sanity checks: 1) checking if the inst. seg. mask is continuous (if not, it
indicates that the object may be occluded, and we skip the corresponding grasp candidate),
2) optimizing the opening width of the gripper by expanding the height of the predicted grasp
candidate towards the edges of the inst. segmentation mask, and 3) calculating the centroid
of the graspable object using the inst. segmentation mask, as the distance from the centroid
to the center of the grasp candidate is important for correctly placing the object in the storage
container. Although very simple, these checks help to achieve a high success rate for picking
and placing objects. Another way to leverage segmentation results is to focus on obstacle
awareness and collision avoidance, similar as [25] proposed for 6-DOF grasping.

Grasping experiments and results. Multiple logs were randomly placed in the scene, with
different locations and orientations, often resulting in highly cluttered scenes. Our system
automatically predicts a grasp candidate for a graspable log and starts the picking process.
This procedure continues until no logs are detected. The grasping experiments collected
data from 12 test runs with up to 7 logs each. Figure 5 d) - e) show an example for grasp
detection and object/obstacle segmentation, respectively. Each grasp attempt is evaluated
for its success and the overall result is presented in Table 3 for completeness. Additional
information about our robotic system can be found in the supplementary material.

a) Crane model   b) Gripper   c) RGB input image   d) Predicted grasp candidate and centroid of object   e) Segmentation of graspable object / obstacle objects
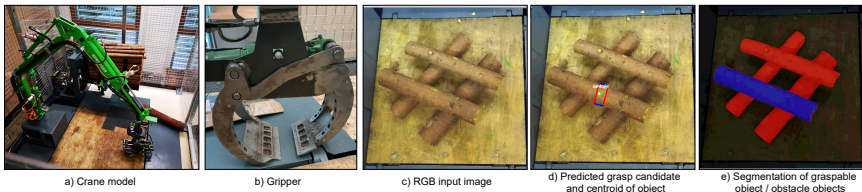
Figure 5: Overview of the experimental setup and visualization of results. a) shows the 1:5
scaled version of a forestry crane, b) shows the used gripper. c) shows an example RGB
input image, with the corresponding results shown in d) - e).

Total attempts	Inaccurate grasp detection	Physical failure	Grasp success rate (%)
58	2	1	94.82

Table 3: Results for picking real objects in highly cluttered scenes. High grasp success rate
was achieved by leveraging segmentation results to optimize picking and placing of objects.

6   Conclusion

In this work, we introduced a novel CNN architecture for joint grasp detection and highly
accurate class-agnostic instance segmentation for robotic picking tasks. We showed that our
proposed depth-aware CoordConv module is a simple yet effective way to enhance point
proposal based object instance segmentation, especially in scenes where objects highly over-
lap. Using our proposed network architecture, we showed highly accurate results for the
challenging Siléane and OCID_grasp datasets, and implemented a practical experiment to
highlight the benefits of joint grasp detection and segmentation for picking objects. In the
future, we plan to use the segmentation results for collision avoidance during path planning
and picking of objects.

Acknowledgments

References

[1] Stefan Ainetter and Friedrich Fraundorfer. End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 13452–13458, 2021.

[2] Ryosuke Araki, Takeshi Onishi, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Mt-dssd: Deconvolutional single shot detector using multi task learning for object detection, segmentation, and grasping detection. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 10487–10493, 2020.

[3] Umar Asif, Jianbin Tang, and Stefan Harrer. Densely supervised grasp detector (dsgd). In *Proceedings of the AAAI Conf. on Artificial Intelligence*, volume 33, pages 8085–8093, 2019.

[4] Antonio Bicchi and Vijay Kumar. Robotic grasping and contact: A review. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, volume 1, pages 348–353, 2000.

[5] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, pages 289–309, 2013.

[6] Romain Brégier, Frédéric Devernay, Laetitia Leyrit, and James L. Crowley. Symmetry aware evaluation of 3d object detection and pose estimation in scenes of many parts in bulk. In *Proceedings of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2017.

[7] Shehan Caldera, Alexander Rassau, and Douglas Chai. Review of deep learning methods in robotic grasp detection. *Multimodal Technologies and Interaction*, 2(3):57, 2018.

[8] Michael Danielczuk, Matthew Matl, Saurabh Gupta, Andrew Li, Andrew Lee, Jeffrey Mahler, and Ken Goldberg. Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 7283–7290, 2019.

[9] Marcus Gualtieri and Robert Platt. Learning 6-dof grasping and pick-place using attention focus. In *Conf. on Robot Learning*, pages 477–486. PMLR, 2018.

[10] Di Guo, Fuchun Sun, Huaping Liu, Tao Kong, Bin Fang, and Ning Xi. A hybrid deep architecture for robotic grasp detection. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1609–1614, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conf. on Computer Vision (ECCV)*, pages 630–645, 2016.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2961–2969, 2017.

[13] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 1501–1510, 2017.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Int. Conf. on Machine Learning (ICML)*, pages 448–456. PMLR, 2015.

[15] Satoshi Ito and Susumu Kubota. Point proposal based instance segmentation with rectangular masks for robot picking task. In *Proceedings of the Asian Conf. on Computer Vision (ACCV)*, 2020.

[16] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 12627–12637, 2019.

[17] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conf. on Robot Learning*, pages 651–673. PMLR, 2018.

[18] Sulabh Kumra and Christopher Kanan. Robotic grasp detection using deep convolutional neural networks. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 769–776, 2017.

[19] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *The Int. Journal of Robotics Research*, 34(4-5):705–724, 2015.

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[21] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, 2018.

[22] Mohammad Reza Loghmani, Mirco Planamente, Barbara Caputo, and Markus Vincze. Recurrent convolutional fusion for rgb-d object recognition. *IEEE Robotics and Automation Letters*, 4(3):2878–2885, 2019.

[23] Douglas Morrison, Peter Corke, and Jürgen Leitner. Learning robust, real-time, reactive robotic grasping. *The Int. Journal of Robotics Research*, 39(2-3):183–201, 2020.

[24] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 2901–2910, 2019.

[25] Adithyavairavan Murali, Arsalan Mousavian, Clemens Eppner, Chris Paxton, and Dieter Fox. 6-dof grasping for target-driven object manipulation in clutter. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 6232–6238, 2020.

[26] D. Park, Y. Seo, D. Shin, J. Choi, and S. Y. Chun. A single multi-task deep neural network with post-processing for object detection with reasoning and robotic grasp detection. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 7300–7306, 2020.

[27] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3406–3413, 2016.

[28] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8277–8286, 2019.

[29] Deirdre Quillen, Eric Jang, Ofir Nachum, Chelsea Finn, Julian Ibarz, and Sergey Levine. Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods. In *2018 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 6284–6291, 2018.

[30] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1316–1322, 2015.

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. Journal of Computer Vision*, 115(3):211–252, 2015.

[32] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *Proceedings of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 7355–7363, 2019.

[33] Markus Suchi, Timothy Patten, David Fischinger, and Markus Vincze. Easylabel: A semi-automatic pixel-wise object annotation tool for creating robotic rgb-d datasets. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 6678–6684, 2019.

[34] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 12695–12705, 2020.

[35] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. Unseen object instance segmentation for robotic environments. *IEEE Transactions on Robotics*, 2021.

[36] Chenjie Yang, Xuguang Lan, Hanbo Zhang, and Nanning Zheng. Task-oriented grasping in object stacking scenes with crf-based semantic model. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 6427–6434, 2019.

[37] H. Zhang, X. Lan, S. Bai, L. Wan, C. Yang, and N. Zheng. A multi-task convolutional neural network for autonomous robotic grasping in object stacking scenes. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 6435–6442, 2019.

[38] Xinwen Zhou, Xuguang Lan, Hanbo Zhang, Zhiqiang Tian, Yang Zhang, and Narming Zheng. Fully convolutional grasp detection network with oriented anchor box. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 7223–7230, 2018.