2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

GSMR-CNN: An End-to-End Trainable Architecture for Grasping Target Objects from Multi-Object Scenes

Valerija Holomjova[1], Andrew J. Starkey[1] and Pascal Meißner[1]

***Abstract*— We present an end-to-end trainable multi-task model that locates and retrieves target objects from multi-object scenes. The model is an extension of the Siamese Mask R-CNN, which combines the components of Siamese Neural Networks (SNNs) and Mask R-CNN for performing one-shot instance segmentation. The proposed network, called Grasping Siamese Mask R-CNN (GSMR-CNN), extends Siamese Mask R-CNN by adding an additional branch for grasp detection in parallel to the previous object detection head branches. This allows our model to identify a target object with a suitable grasp simultaneously, as opposed to other approaches that require the training of separate models to achieve the same task. The inherent SNN properties enable the proposed model to generalize and recognize new object categories that were not present during training, which is beyond the capabilities of standard object detectors. Moreover, an end-to-end solution uses shared features entailing less model parameters. The model achieves grasp accuracy scores of 92.1% and 90.4% on the OCID grasp dataset on image-wise and object-wise splits. Physical experiments show that the model achieves a grasp success rate of 76.4% when correctly identifying the object. Code and models are available at *https://github.com/valerija-h/grasping_siamese_mask_rcnn*.**

I. INTRODUCTION

Deep learning models have enabled robotic arms to grasp diverse sets of objects within unstructured environments without the need for an object model, permitting them to aid humans in various industrial and assistive tasks. In a subset of these tasks that involve post-grasp manipulations (e.g. cooking, assembling, organizing), the system would often require or benefit from recognizing the object it is grasping. This is typically achieved by using object detection models [1], [2], which have long training times and are limited to only identifying object categories they were trained on.

To address this issue, one-shot learning techniques that can classify objects when given a single or few examples, such as Siamese Neural Networks (SNNs) [3], [4], were developed. SNNs are composed of two identical sub-networks that share the same weights. By passing in different inputs into each sub-network, the extracted feature vectors can be used to predict a similarity score between the inputs. By combining the components of SNNs and an instance segmentation network (Mask R-CNN [2]), Michaelis *et al.* [5] create a network called Siamese Mask R-CNN (SMR-CNN) that shows potential in detecting and segmenting object categories that were not present during training from complex RGB scenes in the MS-COCO dataset [6].

[1]All authors are with the University of Aberdeen {v.holomjova.21, a.starkey, pascal.meissner}@abdn.ac.uk

This research is funded by a studentship awarded by the School of Engineering at the University of Aberdeen, Scotland UK.

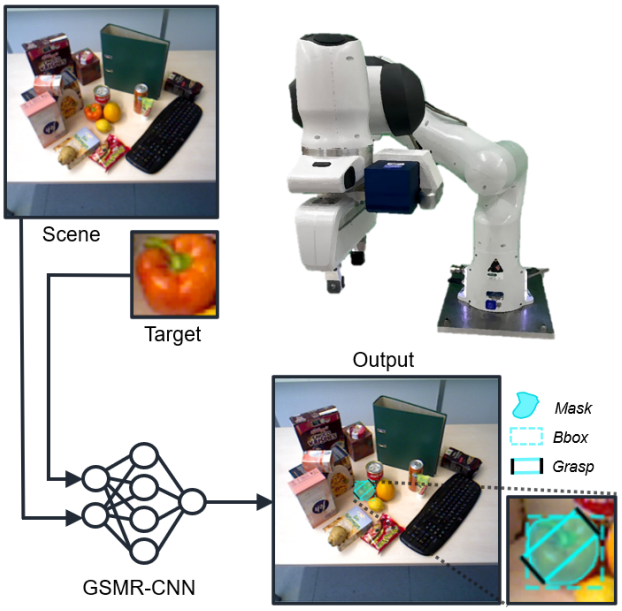

Fig. 1. The proposed system (GSMR-CNN) takes as input both an RGB image of a multi-object scene and an RGB image of the target object (e.g. pepper) to be located and grasped. GSMR-CNN then finds and outputs all instances of the target object in the scene with a corresponding bounding box, segmentation mask and a 2D antipodal grasp for each instance. Physical experiments of GSMR-CNN are carried out using a 7-DoF robotic arm by Franka Emika equipped with a RealSense RGB-D camera.

In this paper, we propose an end-to-end trainable multi-task model, called Grasping Siamese Mask R-CNN (GSMR-CNN), that can detect and segment target objects, and predict a 2D antipodal grasp for each target object from multi-object RGB scenes in parallel (Fig. 1). GSMR-CNN extends SMR-CNN by adding a branch for grasp detection in parallel to the original object detection head branches. This allows GSMR-CNN to identify target objects with suitable grasps simultaneously, in contrast to other grasping systems that perform the same task using separate models [7], [8]. The inherent SNN properties of the model will improve its generalization capabilities towards new object categories without needing re-training, which is the case for grasping systems that use standard object detectors [9], [10]. Additionally, an end-to-end architecture results in fewer model parameters, which could lead to faster training and inference times, which is highly desired for robotic systems.

This paper aims to evaluate the performance of GSMR-CNN in retrieving target objects from multi-object scenes and explore the extent it can generalize to various household objects. The main **contributions** of this paper can be summarized as follows:

- We present an end-to-end network, GSMR-CNN, capable of performing one-shot instance segmentation and grasp detection simultaneously. Results show that GSMR-CNN achieves grasp accuracy scores of 92.1% and 90.4% on the OCID grasping dataset [9] on image-wise and object-wise splits when correctly recognizing the object.
- We propose an alternate target image generation strategy during training to reduce false positives that arise from confusing input target instances.
- Experiments with a 7-DoF robotic arm having an RGB-D camera are carried out to evaluate the system's ability to identify and grasp previously unseen household objects from multi-object scenes. The system achieves a 76.4% grasping success rate on objects correctly recognized.

II. BACKGROUND AND RELATED WORK

A. Instance Segmentation and One-Shot Learning

The task of localizing and segmenting instances of object classes from a visual scene is referred to as instance segmentation. The Mask R-CNN [2] is a state-of-the-art network used for instance segmentation and an extension of the Faster R-CNN [1] object detector. Both [1] and [2] are restricted to recognizing object categories they were trained on.

One-shot learning is the task of identifying object classes from a single or very few examples. Siamese Neural Networks (SNNs) [3] are a popular one-shot learning architecture commonly used for facial recognition [11] and signature verification tasks [12]. They are composed of two identical sub-networks that allow the model to compare two different inputs and predict their level of similarity. Koch *et al.* [4] construct an SNN architecture from Convolutional Neural Networks (CNNs) for tackling one-shot image recognition and demonstrate its superior generalization capabilities towards new objects. By combining the properties of SNNs and Mask R-CNN, Michaelis *et al.* [5] create a network called SMR-CNN that identifies and segments target objects from complex RGB scenes. Similar to [4], SMR-CNN shows potential in generalizing object categories beyond those found in the training set, as opposed to [1], [2].

B. Robotic Grasp Detection

Grasp detection is the task of finding a stable and achievable grasp configuration for any given object in a scene. Over the years, different deep learning solutions have been explored for solving grasp detection. These include simulated or synthetic-based approaches [13]–[16], self-supervised learning [17], [18] or generating grasps from sparse point cloud data [19], [20]. Two-stage detectors similar to standard object detectors (e.g. Faster R-CNN) are a simple yet effective approach for predicting 2D grasp poses from visual observations [9], [21], [22]. Over the years, several datasets containing single or multi-object scenes with annotated 2D grasps have been introduced for researchers to evaluate and compare their approaches [9], [21], [23], [24].

The 2017 Amazon Robotics Challenge tasked its participants with picking objects from a cluttered bin and categorizing them. The winning system [7] achieved a high object recognition accuracy by isolating each object from the clutter through grasps and then categorizing it to a product image in a database using a Siamese-like CNN. Araki *et al.* [10] built a model from a single-shot detector to carry out object detection, semantic segmentation and suction cup grasp detection in parallel. Ainetter *et al.* [9] proposed a network that leverages semantic segmentation maps to further refine grasp candidates. They later build upon their work by demonstrating the benefit of adding instance segmentation maps to improve detection and grasp accuracy within cluttered scenes [25].

Danielczuk *et al.* [8] defined the task *mechanical search* which involves retrieving a target object from a cluttered environment within minimal time. Their system used an individual Mask R-CNN and SNN to segment and recognize the target object from the scene, which is then fed into a search policy module to retrieve the object through a series of actions (e.g. push, grasp, suction). The drawback to current grasping systems that use object detection is that they either require the training of multiple models to generalize [7], [8], or are limited to recognizing object categories they were trained on [9], [10], [25], which is solved by GSMR-CNN.

III. PROBLEM STATEMENT

Given an RGB image of a target object and an RGB image of a multi-object scene containing a set of objects O with n instances of the target object, the objective is to localize and retrieve all instances of the target object $T = \{t_1, t_2, .., t_n\}$ from the scene. Each target object t is localized in the scene by predicting a bounding box $b = (x^b, y^b, w^b, h^b)$ and segmentation map s. The bounding box is a rectangle that encapsulates the object s.t. the parameters $(x^b, y^b), w^b, h^b$ represent the top-left corner co-ordinates, width and height of the bounding box respectively. The segmentation map s is a binary mask of the RGB image where each pixel is labelled as an 'object' (i.e. 1) or 'background' (i.e. 0) based on if it is part of the target object t or not.

Given a parallel plate gripper, a target object t can be retrieved by predicting a 2D grasp pose g that can be parameterized as an oriented bounding box $g = (x^g, y^g, w^g, h^g, \theta^g)$ [23], [26]. These parameters denote center co-ordinates (x^g, y^g), gripper opening distance w^g, gripper size w^h and an orientation θ^g w.r.t the horizontal axis (Fig. 2). Given the symmetry of the grippers, θ^g lies in the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$.

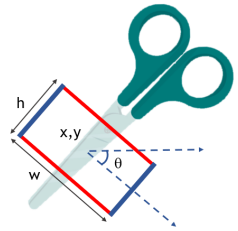

Fig. 2. An example of a 2D grasp pose $g = (x^g, y^g, w^g, h^g, \theta^g)$ on an object, with center co-ordinates (x^g, y^g), gripper opening w^g, gripper size h^g and θ^g rotation.

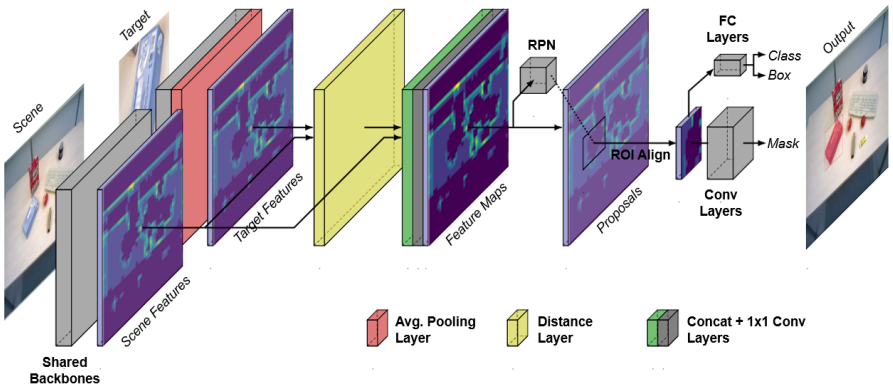

Fig. 3. Architecture of Siamese Mask R-CNN [5] for one-shot instance segmentation.

IV. GRASPING SIAMESE MASK R-CNN

GSMR-CNN is an extension of [4], which was designed to perform one-shot instance segmentation and showed potential in generalizing to new object categories on the MS-COCO dataset [6]. We extend [4] by adding an additional branch for grasp detection. The motivation behind this approach is to inherit generalization properties from SNNs and create an end-to-end architecture resulting in fewer model parameters, which could lead to reduced inference and training times. Considering a ResNet-50 Feature Pyramid Network (FPN) backbone, GSMR-CNN has a total of 52.4M parameters. However, a multi-network approach using an SMR-CNN model (42.5M) for object recognition followed by a standard object detector for grasp detection [22] (41.4M) would yield a cumulative total of 83.9M parameters.

Our aim is to train the network to find and retrieve detailed and specific categories of household objects (e.g. sanitizer, screwdriver) as opposed to the broad categories found in MS-COCO (e.g. car, door). Next, we introduce and describe the underlying architectures of GSMR-CNN, followed by implementation details of the novel grasp detection branch.

A. Mask R-CNN

When given an image, Mask R-CNN uses a CNN-based backbone (e.g. ResNet-50 [27]) to encode image features, which are then passed into a Region Proposal Network (RPN) that generates object proposals (Fig. 4). Features are extracted from each proposal and resized using the RoI align layer and then passed into two separate branches; one that predicts an object class and bounding box, and another that outputs a segmentation map for each proposal.

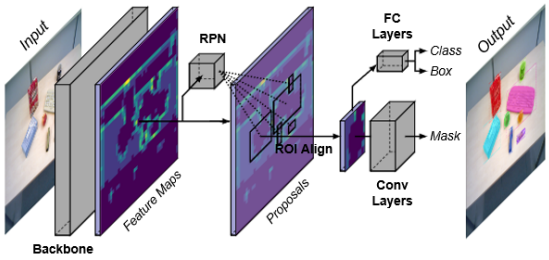

Fig. 4. Architecture of Mask R-CNN [2] for instance segmentation.

B. Siamese Mask R-CNN

Siamese Mask R-CNN uses the same architecture as Mask R-CNN with a modified feature extractor component that allows the model to encode both the content of the scene and its similarity to the target image (Fig. 3). This is achieved by splitting the input stream of the original Mask R-CNN into two, which both use the same ResNet-50 backbone model with shared weights to extract features from the RGB scene image and the RGB target object image. The target object features are then average-pooled into an embedding vector to compute the difference (i.e. L1 distance) between the target object and scene embedding at each pixel position. The differences are concatenated with the original scene features and passed into a 1x1 convolution to reduce dimensionality, whose output is passed to the remaining components of the original Mask R-CNN network like the original backbone features. Although the remaining architecture of the model is the same, the head object classes predicted by Siamese Mask R-CNN are binary (i.e. whether they match the target object or not).

C. Grasp Detection

As depicted in Figure 5, grasp detection is achieved by extending the heads of Siamese Mask R-CNN, which are identical to that of Mask R-CNN. We add an extra branch parallel to the original object detection branch and use the same 7x7 RoI features. The grasp detection branch consists of two 1024 unit fully connected layers and an output layer with k×5 dimensional output for each RoI, where k=2 represents the possible classes the model predicts (i.e. 'target object' or 'background') and 5 denotes the grasp pose parameters $(x^g, y^g, w^g, h^g, \theta^g)$.

With the added grasp detection branch, we now define a multi-task loss function L for each RoI to train the model;

$$L = \lambda_c L_{class} + \lambda_b L_{bbox} + \lambda_m L_{mask} + \lambda_g L_{grasp} \quad (1)$$

where the classification loss L_{class}, bounding box loss L_{bbox} and mask loss L_{mask} are the same as defined in [1], [2] with hyperparameter weights λ assigned. Since there are multiple ground-truth grasp annotations in each scene and a single grasp prediction, the grasp loss L_{grasp} retrieves the closest

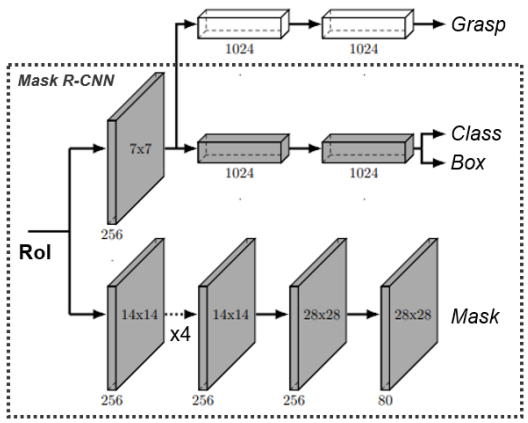

Fig. 5. Head architecture of GSMR-CNN with the added grasp detection branch (white) and original object detection and mask branches from Mask R-CNN (grey).

ground-truth grasp (Eq. 4) to the predicted grasp for loss computation and then applies smooth L1 loss.

Given a set of ground-truth grasps $\mathcal{GT} = \{gt_0, gt_1, ..\}$, the closest-ground truth grasp gt_c to the predicted grasp gp is retrieved by finding the gt with the smallest center coordinate distance c_{diff} and rotation difference θ_{diff};

$$c_{\text{diff}}(gt, gp) = |x^{gt} - x^{gp}| + |y^{gt} - y^{gp}| \tag{2}$$

$$\theta_{\text{diff}}(gt, gp) = |((\theta^{gt} - \theta^{gp}) + \tfrac{\pi}{2})\ mod\ \pi - \tfrac{\pi}{2}| \tag{3}$$

$$gt_c = \underset{gt \in \mathcal{GT}}{\text{argmin}}\ (c_{\text{diff}}(gt, gp) + \theta_{\text{diff}}(gt, gp)) \tag{4}$$

We choose to retrieve the closest grasp instead of a random grasp to avoid penalizing correctly predicted grasps. Similar to L_{bbox} and L_{mask}, only positive (i.e. non-background) classes contribute to the L_{grasp}.

V. EXPERIMENTS AND EVALUATION

The proposed network is built upon the implementation of [5] and Matterport's Mask R-CNN library [28] in TensorFlow [29] with Python 3.7. We assess the capabilities of GSMR-CNN by training and evaluating it on the OCID grasping dataset [9], as well as conducting physical grasping experiments using a 7-DoF robotic arm. Further details on the training process, evaluation procedure and physical experiments are provided below.

A. OCID Grasping Dataset

The OCID grasping dataset is publicly available and contains 1,763 RGB images of multi-object scenes with 31 different object classes, where each object is annotated with a class, segmentation map, and multiple hand-annotated grasp candidates. The dataset is split into training and testing sets consisting of 1,411 and 352 images respectively. Since certain object classes had a large number of missing grasp annotations, only 25 classes were used in training. A majority of these classes include large objects (e.g. keyboards, binders) which are difficult to grasp, which could create a bias in the dataset towards small and medium objects.

B. Target Image Generation

At each iteration during training, the model takes a scene image from the training set and selects a random object from it as the target object, and feeds both into the model. The solution by [5] generates a target image by choosing a random scene image in the training set in which the target's object class is present, and then cropping and resizing the target object from the image. Since our objective is to retrieve more specific instances of objects than the diverse categories found in the MS-COCO dataset, we extract target objects from the scene images directly and add the mask of the object to obtain a color mask. Applying the mask allows us to isolate the object from the scene, as they are very cluttered resulting in multiple objects being present in the target image otherwise. Using the scene image as opposed to random images also prevents certain confusing target instances due to object classes in the dataset having similar shapes to each other (e.g. lemon, lime, apple, orange) but diverse colors between their instances (e.g. ball), making it difficult for the network to differentiate colors and shapes. To maintain variability, we apply data augmentations to our target images consisting of random flips and rotations.

C. Training Details

Prior to training, a ResNet-50 FPN model with weights pre-trained on ImageNet [30] is used as the backbone. The weights of the backbone are initially frozen and the network is trained for 25 epochs, where the first epoch consists of training the head branches solely and the remaining epochs are used to train the rest of the network weights. The network is trained using a batch size of 4 on a system equipped with an NVIDIA GeForce RTX 3070 with CUDA 11.3 and an Intel Core i9 processor. A Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.02 was used and reduced to 0.002 for the last five epochs.

D. Evaluation Procedure

The model is evaluated when trained on two separate data splits; image-wise and object-wise. An image-wise split entails all object classes are present during training and testing, whereas an object-wise split entails splitting the object categories into training and testing categories such that the model is only evaluated on testing categories.

We adopt a similar evaluation procedure to [5] where for each data split, we evaluate GSMR-CNN by iterating through each image in the test set and retrieving a target image (Section V-B) for each of the test categories present in the scene image. The target and scene image are then passed to the model and the resulting prediction is assigned to its respective test category. Each prediction is evaluated in terms of object recognition (i.e. detection and segmentation) and in grasp detection (Section V-E).

E. Evaluation Metrics

Object recognition results are calculated using standard MS-COCO metrics for both bounding boxes (i.e. detection) and masks (i.e. segmentation). These metrics consists of

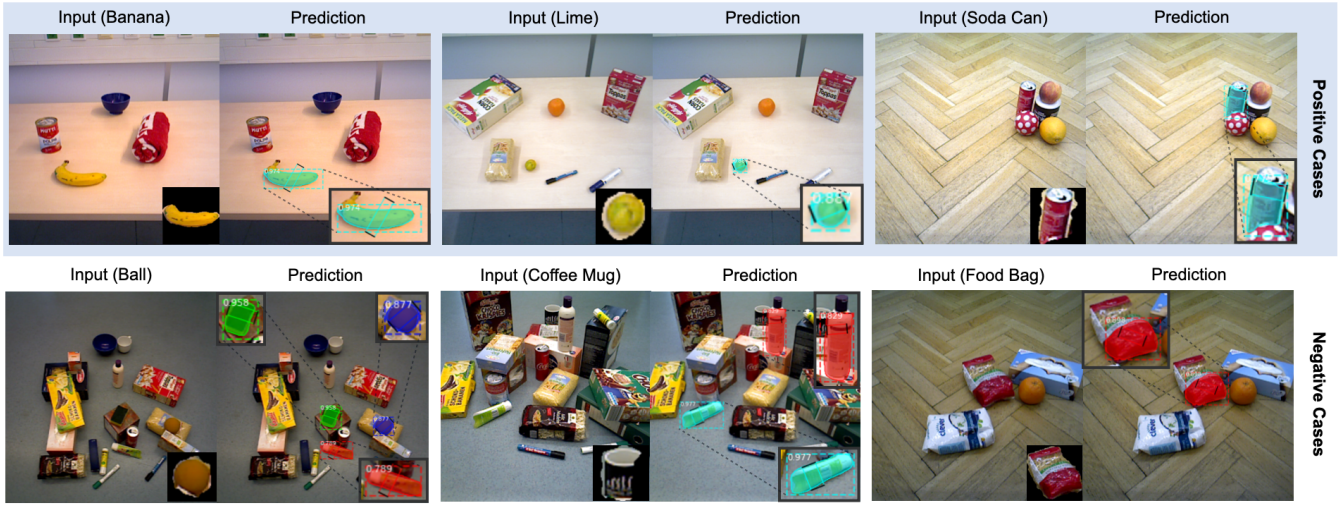

Fig. 6. Example of predictions made by GSMR-CNN on novel object categories that were not present during training (i.e. object-wise split) from the OCID grasping dataset. The top row shows positive cases where the model manages to solely detect and predict a grasp on the target object in the scene. The bottom row shows negative cases consisting of the model either generating false positives (left), missing the target object (center) or not fully masking it resulting in a failed grasp (right).

Average Precision (AP) and Average Recall (AR) scores at different Intersection over Union (IoU) thresholds (averaged, 0.50, 0.75), object scales (small, medium, large) and maximum amount of detections (1, 10, 100).

To evaluate grasp detection, we calculate grasp accuracy (GA) similar to previous literature [9], [21], [22]. This metric classifies a predicted grasp rectangle gp as a success when evaluated against a ground truth grasp rectangle gt if both of these conditions are met:

- The angle difference between gp and gt is within $30°$.
- The IoU score between gp and gt is greater than 25%:

$$\text{IoU}(gp, gt) = \frac{|gp \cap gt|}{|gp \cup gt|} > 0.25 \tag{5}$$

Given that a ground-truth grasp is required, this metric is only calculated if an object is correctly recognized. Grasp accuracy is evaluated for detection and segmentation tasks, which use the bounding box and mask respectively in IoU calculation to determine true positives. This helps determine whether grasps are more accurate when bounding boxes or masks are correctly identified.

Fig. 7. Random household objects used for physical grasping experiments. These include: highlighter, screwdriver, screw, clementine, scissors, sanitizer, measuring tape, soup can, soup box, masking tape.

F. Physical Experiments

The generalization capabilities of GSMR-CNN is evaluated through physical experiments using a 7-DoF robotic arm by Franka Emika equipped with a D415 Intel RealSense camera that captures RGB-D images. The trained image-wise split model is used for inference, and the Frankx library [31] is used for motion planning. Experiments are carried out on 10 random household objects (Fig. 7) of varying colours and shapes to build various multi-object scenes with up to three target objects present. The full-width of the gripper is used in experiments as opposed to the predicted gripper width w^g, since the grasp pose widths from OCID form tightly around object boundaries being suboptimal for planar grasping.

VI. RESULTS

A. Object Recognition

Table I summarizes the results of GSMR-CNN on object-wise and image-wise data splits for both detection and segmentation tasks. Overall, GSMR-CNN performs well on image-wise split and achieves a mean AP score of 52.6% and 55.9% for detection and segmentation respectively, surpassing [5] that had scores of 21.8% and 19.3%. However, GSMR-CNN performed worse on the object-wise split and had a mean AP score of 3.0% and 2.9%, whereas [5] acquired scores of 9.1% and 7.7%.

Given that test classes from the object-wise split are omitted from target image generation, but can still be present in the image scene, the poor performance could be attributed to the model learning to ignore the test object features within the image scene during training. This is similar to the model learning to ignore the scene background. Thus, the same object features that were ignored during training are then ignored during evaluation leading to low detection scores on the test classes from the object-wise split. One possible solution to solve this is to create a dataset with scene images that only contain object classes of their respective

TABLE I

OBJECT RECOGNITION RESULTS ON OCID

Split	Task	AP	AP^{50}	AP^{75}	AP^{S}	AP^{M}	AP^{L}	AR^{1}	AR^{10}	AR^{100}	AR^{S}	AR^{M}	AR^{L}
image	detection	52.6	70.7	67.3	37.1	53.5	54.0	50.1	67.5	67.5	37.4	69.1	76.3
	segmentation	55.9	70.3	67.5	36.2	56.8	63.0	52.4	70.1	70.1	36.3	71.8	86.3
object	detection	3.0	5.5	3.1	5.5	2.7	5.6	6.4	14.3	14.3	5.9	15.1	11.6
	segmentation	2.9	4.9	3.2	5.2	2.6	3.3	6.2	13.6	13.6	5.4	14.5	6.0

TABLE II

GRASP DETECTION RESULTS ON OCID

Split	Task	GA	GA^{50}	GA^{75}	GA^{S}	GA^{M}	GA^{L}	GA^{1}	GA^{10}	GA^{100}
image	detection	91.9	91.3	91.9	87.1	92.5	70.6	91.5	91.9	91.9
	segmentation	92.2	91.4	91.5	93.1	92.9	64.0	91.5	92.2	92.2
object	detection	92.0	88.2	95.0	100.0	91.3	84.7	94.6	92.0	92.0
	segmentation	88.7	87.8	90.0	100.0	87.8	95.7	92.6	88.7	88.7

splits. Nonetheless, Figure 6 still shows potential in GSMR-CNN recognizing novel object categories. As similar work performing one-shot image recognition [5], [8] also reported failures in their object recognition components, further work needs to be carried out to enhance the underlying Siamese properties of the network. Besides creating a new dataset as previously suggested, this improvement could be in the form of modifying the internal SNN architecture or exploring different data generation techniques (e.g. using depth, adding more data augmentation).

B. Grasp Detection

The grasp detection results of GSMR-CNN on the OCID grasping dataset for each MS-COCO metric parameter are shown in Table II. The table shows that the system obtains a high grasp accuracy score in both data splits and tasks, attaining an average accuracy score of 92.1% and 90.4% over both tasks on image-wise and object-wise splits respectively. The similarity between the object-wise and image-wise scores suggests that the model was able to generalize predicted grasps to novel object categories successfully.

It is observed that the model mostly had lower grasp accuracy scores when dealing with larger object areas (GA^{L}), which could be due to the dataset having fewer grasp annotations for larger object categories. We believe that grasp accuracy could further be improved by modifying the grasp detection component to perform multi-grasp detection.

C. Physical Experiments

Table III summarizes the results of the physical grasping experiments, depicting the number of times the system successfully detected and grasped each target object instance from various multi-object scenes. Since it is difficult to discern whether a predicted grasp is stable and correct in a real-life setting, we only consider grasp success based on physical success and not the prediction itself. The table shows that the model was able to identify 55.0% of the target instances successfully. GSMR-CNN had difficulty segmenting objects with more complex geometries (e.g. scissors and masking tape) than those present in the training dataset. Moreover, the model had issues identifying larger objects (e.g. soup can and soup box), which could be due to the difference in camera perspective from the training dataset. Overall, the model achieves a grasp success rate of 76.4% when correctly identifying the target object. Grasp failures were mostly attributed to unsuccessfully detecting the entire object resulting in unstable grasps, accidental faults such as collisions with nearby objects, or objects slipping from the grippers. Grasp success rates may be further improved through multi-grasp detection and use of pre-grasp manipulations (e.g. pushing).

TABLE III

RESULTS FROM PHYSICAL EXPERIMENTS

Object	Object Detection	Physical Grasp Success	False Positives
clementine	10/10	8/10	1
highlighter	7/10	7/7	2
screw	8/10	7/8	1
screwdriver	8/10	7/8	3
measuring tape	9/10	6/9	3
sanitizer	7/10	6/7	8
scissors	5/10	1/5	3
masking tape	0/10	0/0	7
soup can	1/10	0/1	0
soup box	0/10	0/0	0
Overall	55/100	42/55 (76.4%)	28

VII. CONCLUSIONS

This paper presents an end-to-end trainable multi-task model, called GSMR-CNN, that is able to simultaneously detect, segment, and predict 2D antipodal grasps of target objects from multi-object RGB scenes. We extend SMR-CNN by adding an additional branch for grasp detection in parallel to the object detection head branches. Results show that GSMR-CNN obtains grasp accuracy scores of 92.1% and 90.4% on image-wise and object-wise data splits, as well as grasp success rates of 76.4% on physical experiments with random household objects. The performance of the model is currently limited by the object recognition component and potential areas of improvement have been identified.

Currently, the system is limited to predicting a single grasp for each target object instance in the scene. Future work consists of extending the system to multi-grasp detection and improving the object recognition component.

REFERENCES

[1] S. Ren *et al.*, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, C. Cortes *et al.*, Eds., vol. 28, 2015, pp. 91–99.

[2] K. He *et al.*, "Mask R-CNN," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[3] J. Bromley *et al.*, "Signature Verification using a "Siamese" Time Delay Neural Network," *Advances in Neural Information Processing Systems*, vol. 6, pp. 737–744, 1993.

[4] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, "Siamese Neural Networks for One-shot Image Recognition," in *ICML Deep Learning Workshop*, vol. 2, Lille, 2015.

[5] C. Michaelis *et al.*, "One-shot Instance Segmentation," *arXiv preprint arXiv:1811.11507*, 2018.

[6] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," in *European Conference on Computer Vision (ECCV)*, Springer, 2014, pp. 740–755.

[7] A. Zeng *et al.*, "Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 3750–3757.

[8] M. Danielczuk *et al.*, "Mechanical Search: Multi-step Retrieval of a Target Object Occluded by Clutter," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 1614–1621.

[9] S. Ainetter and F. Fraundorfer, "End-to-end Trainable Deep Neural Network for Robotic Grasp Detection and Semantic Segmentation from RGB," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 13 452–13 458.

[10] R. Araki *et al.*, "MT-DSSD: Deconvolutional Single Shot Detector using Multi Task Learning for Object Detection, Segmentation, and Grasping Detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 10 487–10 493.

[11] R. Chatterjee, S. Roy, and S. Roy, "A Siamese Neural Network-Based Face Recognition from Masked Faces," in *International Conference on Advanced Network Technologies and Intelligent Computing*, Springer, 2021, pp. 517–529.

[12] A. B. Jagtap *et al.*, "Verification of genuine and forged offline signatures using Siamese Neural Network (SNN)," *Multimedia Tools and Applications*, vol. 79, no. 47, pp. 35 109–35 123, 2020.

[13] B. León *et al.*, "OpenGRASP: A Toolkit for Robot Grasping Simulation," in *International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, Springer, 2010, pp. 109–120.

[14] A. Zeng *et al.*, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 4238–4245.

[15] K. Bousmalis *et al.*, "Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 4243–4250.

[16] Y. Lin *et al.*, "Using Synthetic Data and Deep Networks to Recognize Primitive Shapes for Object Grasping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 10 494–10 501.

[17] L. Pinto and A. Gupta, "Supersizing Self-supervision: Learning to Grasp from 50K Tries and 700 Robot Hours," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 3406–3413.

[18] L. Berscheid, P. Meißner, and T. Kröger, "Self-supervised Learning for Precise Pick-and-place without Object Model," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4828–4835, 2020.

[19] P. Ni *et al.*, "PointNet++ Grasping: Learning An End-to-end Spatial Grasp Generation Algorithm from Sparse Point Clouds," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 3619–3625.

[20] A. Murali *et al.*, "6-DOF Grasping for Target-driven Object Manipulation in Clutter," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 6232–6238.

[21] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world Multi-object, Multi-grasp Detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[22] V. Holomjova and P. Meißner, "Exploring Rotated Object Detection Models for Antipodal Robotic Grasping," in *UKRAS22 Conference "Robotics for Unconstrained Environments" Proceedings*, 2022, pp. 62–63.

[23] I. Lenz, H. Lee, and A. Saxena, "Deep Learning for Detecting Robotic Grasps," *The International Journal of Robotics Research (IJRR)*, vol. 34, no. 4-5, pp. 705–724, 2015.

[24] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A Large Scale Dataset for Robotic Grasp Detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3511–3516.

[25] S. Ainetter *et al.*, "Depth-aware Object Segmentation and Grasp Detection for Robotic Picking Tasks," in *British Machine Vision Conference (BMVC)*, BMVA Press, 2021, p. 376.

[26] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD Images: Learning using a new rectangle representation," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2011, pp. 3304–3311.

[27] K. He *et al.*, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[28] W. Abdulla, *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow*, `https://github.com/matterport/Mask_RCNN`, 2017.

[29] Martín Abadi *et al.*, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015. [Online]. Available: `https://www.tensorflow.org/`.

[30] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015.

[31] L. Berscheid, *Frankx: High-Level Motion Library for the Franka Emika Robot*, `https://github.com/pantor/frankx`, 2013.