This is an extended version of the article that appeared at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019.

Please cite this paper as:
A. Eitel, N. Hauff, and W. Burgard, "Self-supervised Transfer Learning for Instance Segmentation through Physical Interaction," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 4020–4026.

@INPROCEEDINGS{eitel19iros,
author = {A. {Eitel} and N. {Hauff} and W. {Burgard}},
booktitle = {2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)},
title = {Self-supervised Transfer Learning for Instance Segmentation through Physical Interaction},
year = {2019},
volume = {},
number = {},
pages = {4020-4026},
}

Self-supervised Transfer Learning for Instance Segmentation through Physical Interaction

Andreas Eitel Nico Hauff Wolfram Burgard

***Abstract*— Instance segmentation of unknown objects from images is regarded as relevant for several robot skills including grasping, tracking and object sorting. Recent results in computer vision have shown that large hand-labeled datasets enable high segmentation performance. To overcome the time-consuming process of manually labeling data for new environments, we present a transfer learning approach for robots that learn to segment objects by interacting with their environment in a self-supervised manner. Our robot pushes unknown objects on a table and uses information from optical flow to create training labels in the form of object masks. To achieve this, we fine-tune an existing DeepMask network for instance segmentation on the self-labeled training data acquired by the robot. We evaluate our trained network (SelfDeepMask) on a set of real images showing challenging and cluttered scenes with novel objects. Here, SelfDeepMask outperforms the DeepMask network trained on the COCO dataset by 9.5% in average precision. Furthermore, we combine our approach with recent approaches for training with noisy labels in order to better cope with induced label noise.**

I. INTRODUCTION

The ability to segment object instances in a category-agnostic manner, i.e., to partition individual objects regardless of the class, is necessary to enhance the visual perception capabilities of a robot for various manipulation tasks, e.g., object instance grasping or object sorting. Additionally, it can be useful for identifying target objects and inferring their spatial relationships from the visual grounding of human language instructions [1], [2]. In the past, several methods have been proposed for object segmentation based on color, texture and 3D features. However, these are known to over-segment multi-colored objects and under-segment objects that are adjacent [3], [4].

Recent segmentation methods based on deep learning require precise hand-labeled segmentation annotations for a large number of objects as training data [5]. Existing large-scale datasets from the computer-vision community consist of RGB images that show natural scenes. As these scenes differ from the typical ones robots encounter (e.g., regarding clutter and the frontal camera viewpoint), a common procedure to learn object perception on a robot is to manually label a new dataset and to fine-tune a pre-trained model given the labeled data. To reduce the labeling effort, often bounding box detectors are used instead of pixel-wise object segmentation methods, because labeling boxes requires less effort compared to segmenting images.

All authors are with the laboratory for Autonomous Intelligent Systems, University of Freiburg, Germany. Wolfram Burgard is also with the Toyota Research Institute, Los Altos, USA.

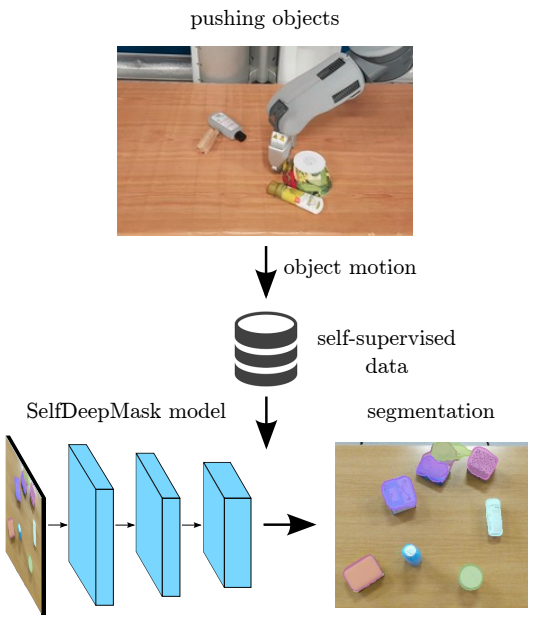

Fig. 1. Our robot collects training data by interaction with its environment (top). We train a ConvNet with the automatically labeled dataset (left). Instance segmentation result using our trained model (right). Implementation: `https://github.com/aeitel/self_deepmask`

Commonly, instance segmentation is considered as an offline process, where labeling is manually performed before training and at test time there is no mechanism to correct mistakes of the learned model. One of the main challenges for robots is to adapt their own perceptual capabilities to new environments. To address this challenge, we present a method that performs interactive data collection and uses a self-supervised labeling process for adapting the trained model to a novel scenario. Our robot interacts with its environment to collect and label its own data by using manipulation primitives and by observing the outcome of its own actions. The overall concept is depicted in Figure 1. It is inspired by research in interactive perception that aims to resolve perception ambiguities via interaction [6], [7] and combines it with the idea of self-supervised learning.

Our approach works as follows: The robot moves an object on the surface of a table in front of it using push actions and collects one RGB image before and one after each action. By detecting coherent motion of pixels in the two images, it creates a binary object mask that selects the moved object. We use this binary mask as a self-supervisory signal, so that our self-supervised method for transfer learning does not require any hand-labeled data. Following this approach, we collect a diverse set of training data consisting of various object

Fig. 2. We use motion between two consecutive images as primary self-supervisory signal to automatically generate object masks. The robot creates object motion by pushing an object. We filter and cluster the resulting optical flow field into segments (not depicted here) to generate a binary mask that we add to the training set.

instances to fine-tune DeepMask [5], a recent instance segmentation CNN pre-trained on the COCO dataset with 886K labeled object instances [8]. We show improved segmentation results using our transfer learning method (SelfDeepMask) compared to the pre-trained DeepMask network. We employ SelfDeepMask as part of an interactive object-separation experiment and show that the segmentation performance at test time increases with each action.

The contributions of this paper are: (1) a self-supervised method that generates labeled training data in form of object masks acquired from robot–object interactions, (2) a method that uses motion information from two consecutive images in combination with the end-effector position to generate a high-quality object mask, (3) real-world experiments evaluating the generalization abilities of our SelfDeepMask network to segment unseen objects in clutter.

II. RELATED WORK

This work builds on prior research in interactive segmentation and self-supervised learning for robotics and computer vision. The main motivation is to improve perception for robot manipulation, e.g., grasping and placing.

A. Interactive Segmentation

Previous work in interactive segmentation considers segmenting specific scenes in an interactive manner using image differencing techniques or visual feature tracking to update the segmentation after each action [9], [6], [10], [11], [12], [13]. Patten *et al.* [14] extend prior methods by enabling online segmentation also during the interaction. We use the perceptual signal from robot–object interactions to create a segmentation mask. Aforementioned online methods do not use learning to generalize to new object scenes, whereas our approach learns a visual model that improves segmentation for new object scenes using data gathered from over 2,300 robot–object interactions. Katz *et al.* [15] present an interactive segmentation algorithm based on a learned model for detecting favourable actions to remove object clutter. An interesting recent overview on interactive perception is presented by Bohg *et al.* [7], who summarize that perception is facilitated by interaction with the environment. Inspired by this line of research we use interaction to improve perception and provide a novel perspective by using interactive perception for self-supervised learning. To improve segmentation performance, we first collect training data through interaction that we use for self-supervised transfer learning. To further improve segmentation performance at test time, our robot uses its ability to interact with objects (i.e., we follow the usual scheme of interactive perception) together with the transferred network for instance segmentation. We will see this example in Fig. 4.

Closest to our work is the recent method by Pathak *et al.* [16] that uses grasping for self-supervised instance segmentation. As opposed to grasping objects, we show results for pushing, which allows to learn arbitrary objects and not only graspable ones. Furthermore, we use motion information to generate masks, while their method uses simple frame differencing, which is less robust to movement of multiple objects as we show in our experiments.

B. Self-supervised Robot Learning

Several recent works have used self-supervised learning for acquiring manipulation skills such as grasping [17], [18], [19], regrasping [20], pushing [21], combined pushing and grasping [22], pouring [23] and tool affordance understanding [24]. Pinto *et al.* [25] learn visual feature representations from manipulation interactions. Several works use multiple views as self-supervision signal for 6D pose estimation, which can be used complementary to our method [26], [27]. None of the mentioned approaches leverage self-supervised learning for instance segmentation. Pot *et al.* [28] learn a bounding box detector in a self-supervised manner by navigating in static environments and associating frames using Simultaneous Localization and Mapping. Wellhausen *et al.* [29] train a terrain segmentation network in a self-supervised manner by navigating with a legged robot.

C. Self-supervised Visual Learning

Schmidt *et al.* [30] learn feature descriptors for dense correspondence. Ovsep *et al.* [31] discover objects in street scene videos using tracking. Pathak *et al.* [32] learn to segment objects by tracking their movement, but only consider single object videos that are passively observed. Milan *et al.* [33] present semi-automatic data labeling for semantic segmentation. Aforementioned methods are self-supervised but not interactive. More recently, Danielczuk *et al.* [34] train a depth-based network for instance segmentation with rendered data from a simulator and show successful transfer to the real world. As we use RGB data, training in simulation is not straightforward because it raises several domain-adaptation challenges, which are not the focus of this work.

III. SELF-SUPERVISED INSTANCE SEGMENTATION BY INTERACTION

In this section, we describe our approach for self-supervised instance segmentation. We require that all objects are movable and are placed on a flat surface. Based on this, we describe how to realize the interactive data collection, how our self-supervised mask generation works and how we perform transfer learning with the autonomously gathered data.

A. Interactive Data

To acquire the data necessary for adapting the model, we need a robot that is able to push objects on a table in front of it with its end effector. In our current approach, we sample push actions horizontally and parallel to the table plane, which we segment using depth information. The robot needs to be able to generate pushes that move objects into free space and keeps them within it. In comparison to random pushing this mitigates the problem of moving objects into each other or moving two objects at the same time. One approach for realizing this has been described in our previous work [35]. We employ this method in the approach described in this paper. Given a robust method for object pushing, human involvement can be rather limited and is only needed to exchange the objects on the table. In principle, this task could also be performed automatically using systems like Dex-Net [36]. To perform trajectory planning, one can use any existing approach like the LBKPIECE motion planning algorithm provided by the Open Motion Planning Library [37], which we also utilize in this paper. Please note that with our approach, depth information is only needed during data collection to extract the surface of the table. In principle the surface can also be extracted using tactile data. Independent of this, at execution time, our method only requires RGB image data.

In our current system, we use a PR2 robot equipped with a Kinect 2 head camera that provides RGB-D images with a resolution of 960×540 pixels. We use both robot arms to enable covering the whole workspace. The overall setup for learning is depicted in Figure 2.

B. Self-supervisory Signal

We use coherent motion of object pixels as the primary supervision signal, see Fig 2. The robot captures images $\mathbf{o}_t$ and $\mathbf{o}_{t+1}$ before and after each push action respectively $\mathbf{a}_t = (x_{push}, y_{push})$. We represent the push action as the pixel in the image where the push started and leverage the stored end-effector state during the interaction together with the known camera–robot extrinsic calibration. Before and after a push interacion, the robot arms are positioned such that they do not obstruct the view for capturing $\mathbf{o}_t$ and $\mathbf{o}_{t+1}$. The goal is to create a labeled training dataset $\mathcal{D} = \{(\mathbf{o}^1, \mathbf{s}^1), \ldots, (\mathbf{o}^N, \mathbf{s}^N)\}$ that consists of images and automatically-labeled segmentation masks $\mathbf{s}^n$. However, not every $\mathbf{o}_t$ is added to $\mathcal{D}$. During data collection, we perform the following steps in each time step t:

training objects test objects

Fig. 3. The set of objects used for training and testing.

We compute the optical flow $\mathbf{u}_t, \mathbf{v}_t = \mathit{flow}(\mathbf{o}_t, \mathbf{o}_{t+1})$ using the FlowNet2 network [38]. We filter the optical flow field by setting all flow vectors that point to the estimated table ground plane to zero. Second, we cluster the filtered flow $\mathbf{u}_t^*, \mathbf{v}_t^*$ using normalized graph cuts $\mathbf{S}_t = \mathit{clusters}(\mathbf{u}_t^*, \mathbf{v}_t^*)$ to obtain a set of segments $\mathbf{S}_t = \{\mathbf{s}_t^1, \ldots, \mathbf{s}_t^L\}$. To handle rotating objects we remove segments from $\mathbf{S}_t$ where the corresponding flow magnitudes on the segment exceed a standard deviation threshold (of 15.0). Next, we pick the segmentation mask $\mathbf{s}_t^l \in \mathbf{S}_t$ containing the push action pixel $\mathbf{a}_t$. Using the push action as a prior information enforces that only segments from $\mathbf{S}_t$ are used that overlap with the end-effector position at the beginning of the push. Finally, we add both the image $\mathbf{o}_t$ and the segmentation mask $\mathbf{s}_l$ to the training set $\mathcal{D}$. Note that using our approach we add at most one mask per time step t to the training set together with the associated image $\mathbf{o}_t$. We further discard complete interactions where the mean magnitude of all optical-flow vectors in the image exceeds a given threshold (of 7.0), to handle scenes with large motions.

C. Network Transfer Learning

We use a state-of-the-art method for category-agnostic instance segmentation known as DeepMask [5]. The core of DeepMask is a ConvNet, which jointly predicts a mask and an object score for an image patch. At test time, the network is fed with RGB image patches in a sliding-window manner using multiple scales. If the score network detects that the center pixel of the image patch belongs to an object it triggers the mask network to produce a corresponding mask. We use our interactive method to fine-tune the score network. The usage of DeepMask is complementary to our contribution of learning in a self-supervised manner from robot–object interactions, i.e., a different segmentation network could also be used.

Given $\mathbf{o}^n$ and $\mathbf{s}^n$ sampled from $\mathcal{D}$ we follow the default image preprocessing steps of DeepMask. The image is resized to different scales (from 2^2 to 2^1 with a step of $2^{1/2}$) and into an image size of 224×224. The score network of DeepMask is trained by sampling positive and negative image patches. DeepMask considers an image patch as positive if it contains an object in a canonical position in the middle of the patch. To account for noise, we jitter positive examples in translation (of ± 16 pixels) and scale deformation (of $2^{\pm 1/4}$). We label an image patch as a negative example if it is

TABLE I

QUANTITATIVE SEGMENTATION RESULTS

Method	Trained on	AP@0.5	AP@0.75	AP@0.5:0.95
DeepMask	COCO	59.3	52.0	40.0
SharpMask	COCO	59.4	48.9	37.8
DeepMask with NMS	COCO	71.4	58.1	45.9
SharpMask with NMS	COCO	71.6	54.5	43.6
DeepMask frame differencing	COCO & ROBOTPUSH	71.2 $\pm$ 6.7	45.9 $\pm$ 8.3	39.6 $\pm$ 5.7
Ours, SelfDeepMask, 1.3k interactions	COCO & ROBOTPUSH	76.8 $\pm$ 2.9	57.9 $\pm$ 3.5	47.1 $\pm$ 2.5
DeepMask, human-labeled	COCO & ROBOTPUSH	84.7 $\pm$ 0.9	66.0 $\pm$ 1.6	53.1 $\pm$ 0.7
Ours, SelfDeepMask, 2.3k interactions	COCO & ROBOTPUSH	80.0 $\pm$ 1.5	57.1 $\pm$ 1.3	47.5 $\pm$ 1.3
Ours, SelfDeepMask, 2.3k interactions, co-teaching [39]	COCO & ROBOTPUSH	80.9 $\pm$ 0.7	64.8 $\pm$ 0.4	51.9 $\pm$ 0.6

at least ± 32 pixels or 2^1 in scale away from a canonical positive example. We enhance the data augmentation pipeline of DeepMask, which by default consists of vertical image flipping ($p = 0.25$) with a $(0 - 360^\circ)$ rotation ($p = 0.25$) of both images and our automatically labeled segmentation masks, which increases robustness to rotations.

All scenes contain multiple objects but our method only labels one object per image. The training loss can be large if the pre-trained model assigns a high confidence to a true positive object in the image that is missing a label. To handle this issue we use bootstrapping, i.e., we use the predictions of the pre-trained network to relabel these potentially false labels [40]. In practice we set the gradient to zero for image patches, for which the pre-trained model assigns a confidence greater than 0.5 for class "object" while the label denotes class "background".

We fine-tune our SelfDeepMask network for ten epochs with a learning rate of 0.001, using stochastic gradient descent with momentum of 0.9 and weight decay of 0.0005. We train each model with five different random seeds to report an uncertainty measure of the final performance and to take into account that we are training with noisy training data. Furthermore, we add non-maximum suppression (NMS) at test time to both DeepMask and our SelfDeepMask to remove false overlapping detections.

IV. EXPERIMENTS

In this section, we evaluate the instance segmentation performance of our approach on the ROBOTPUSH dataset. ROBOTPUSH contains 2,300 training, 50 validation and 190 test scenes (images) of diverse real-world objects. The training set contains over 50 different objects and the test set 16 novel objects to examine generalization performance, see Fig 3. The test scenes contain between six and eight objects, with various levels of clutter. We manually annotated masks in the validation and test scenes for evaluation. We compare against two state-of-the-art category-agnostic methods for instance segmentation called DeepMask [5] and SharpMask [41], both trained with over 886K labeled object instances from the COCO dataset. To improve the performance of the two baselines we add NMS and filter out large masks that cannot correspond to objects in the ROBOTPUSH dataset. We also implement a frame-differencing baseline similar to Pathak *et al.* [16], in which we replaced the optical-flow-based mask generation, while keeping the rest of our method fixed. We train two variants of our SelfDeepMask with different amounts of training data: one trained with 800 training images gathered from 1.3k interactions and one trained with 1.5k images gathered from 2.3k interactions. Finally, we report results in which a human labels 300 training images in a pixel-wise manner.

All data in ROBOTPUSH is collected autonomously by the robot, which uses a learning-based method for object separation (from own prior work) that effectively isolates cluttered objects using push actions [35]. In a second experiment, we perform a fine-grained evaluation of segmentation performance with respect to each push interaction, which provides insights in segmentation performance for various degrees of clutter. We use the same 190 images for evaluation. The NMS threshold is optimized on the validation set. We found a value of 0.5 for SharpMask and DeepMask to give best results. For our SelfDeepMask we set the NMS threshold to 0.4.

A. Quantitative Comparisons

We compare the performance of the methods using the standard COCO instance segmentation benchmark metric. The metric that we report is average precision (AP) over different IoU (intersection over union) thresholds (AP from 0.5 to 0.95, AP at 0.5 and AP at 0.75). Higher AP indicates better performance and higher IoU thresholds penalize localization errors of the methods. The results are shown in Table I. Our SelfDeepMask outperforms SharpMask with NMS and also improves the AP performance for two of the three IoU thresholds with respect to DeepMask with NMS. The results indicate that our self-supervised transfer learning approach is able to further improve the performance of a system that is already trained on large amounts of labeled data. Our results further show that using motion information from optical flow outperforms generating training masks based on frame differencing. Moreover, almost doubling the amount of data results in a moderate improvement in performance and reduces the variance. In addition, combining our approach with Co-teaching [39], a recent method for coping with noisy labels, further improves the performance.

TABLE II

ABLATION STUDIES

Method	AP@0.5	AP@0.75	AP@0.5:0.95
SelfDeepMask, 1.3k interactions	76.8 ± 2.9	57.9 ± 3.5	47.1 ± 2.5
SelfDeepMask without action as prior	73.7 ± 2.7	54.9 ± 3.4	45.5 ± 2.3
SelfDeepMask without bootstrapping	74.6 ± 1.3	54.7 ± 2.0	45.1 ± 1.4
SelfDeepMask without bootstrapping, fine-tune mask head	71.9 ± 1.5	45.1 ± 1.8	38.9 ± 0.5
SelfDeepMask without table filtering	66.5 ± 3.3	29.2 ± 5.8	31.8 ± 2.8

TABLE III

EXPERIMENTS WITH METHODS FOR COMBATING LABEL NOISE AND TRAINING DATA FROM 2.3K INTERACTIONS.

Method	AP@0.5	AP@0.75	AP@0.5:0.95
SelfDeepMask, 2.3k interactions, no noisy-label filtering	72.9 ± 1.5	50.0 ± 2.0	42.6 ± 1.4
SelfDeepMask, reed-hard [42]	76.3 ± 0.8	53.5 ± 0.8	45.3 ± 1.0
SelfDeepMask, bootstrapping heuristic	80.0 ± 1.5	57.1 ± 1.3	47.5 ± 1.3
SelfDeepMask, self-paced [43]	80.4 ± 0.1	64.7 ± 0.2	51.0 ± 0.1
SelfDeepMask, small-loss sampling [39]	80.7 ± 0.7	64.4 ± 0.7	51.5 ± 0.4
SelfDeepMask, co-teaching [39]	80.9 ± 0.7	64.8 ± 0.4	51.9 ± 0.6

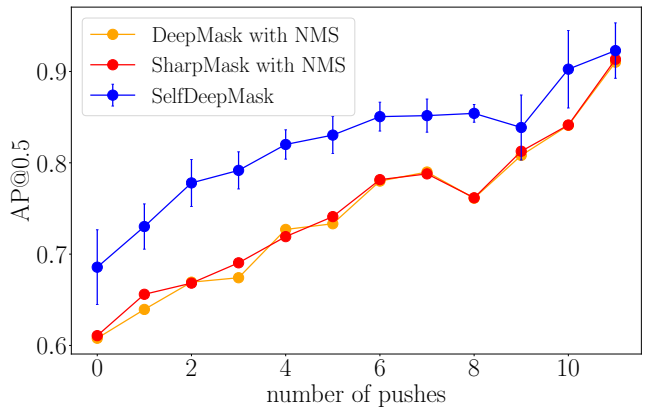

Fig. 4. Interactive segmentation experiment. The robot separates cluttered objects using pushing, which increases the segmentation performance of our SelfDeepMask network at test time after each push. SelfDeepMask achieves higher performance in cluttered scenes, showing higher average precision when then robot has performed only few or no pushes. Results are averaged over 5 models, trained with different random seeds. Error bars denote the standard deviation of the average precision. We use the model trained with 2.3k interactions.

B. Ablation Studies

We conduct several experiments in which we remove one step of our method, see Table II. We observe that the final performance is lower if the action information is not used as a prior to select the mask. Similarly, turning off bootstrapping reduces performance because our method does not account for true positives generated by the pre-trained network in combination with missing object annotations in our training set. Furthermore, we find that fine-tuning the mask head in addition to the score head of DeepMask deteriorates performance, which shows that the mask head is more sensitive to training with noisy masks. Finally, skipping the step of filtering out motion of pixels that map to the table also reduces performance.

C. Learning with Noisy Labels

Real-world data, annotated in a self-supervised manner can be noisy. This is especially the case for a robot that collects and labels its own data as in our case. In this section we test various existing methods that we combine with our SelfDeepMask network to combat label noise. In most existing methods, sample reweighting or removing high-loss samples are commonly used strategies to cope with noisy labels. We implemented five recent methods that modify the classification loss in the score network and compare them in Table III.

Hard bootstrap [42]: The loss term consists of a convex weighted combination of predicted and original labels. We set the weighting factor β to 0.7.

Self-paced learning [43]: Self-paced training uses a pre-defined curriculum, which skips certain data points that are considered to be yet too hard (measured by the per-example loss). It uses a threshold λ to distinguish between easy/hard examples that is increased every epoch with a growing factor (of 1.2). We start with $\lambda = 0.002$, which corresponds to the average loss in the first epoch (a value we took from prior experiments).

Small-loss sampling [39]: This approach is similar to self-paced learning but removes a fixed amount of samples in each batch. We found that removing 10% of samples sorted based on high loss values yields best results.

Co-teaching [39]: It uses the same small-loss sampling scheme but adds a second network. In each mini-batch of data, each network views its small-loss instances and selects the useful instances for its peer network to update the parameters.

Bootstrapping heuristic: Our initial bootstrapping approach that we described in section III-C.

For all methods we tested the last and the best (using early-stopping) model snapshot and report the higher numbers only.

Fig. 5. Visualization of predicted instance segmentation masks generated by DeepMask (left), DeepMask with NMS (middle) and our SelfDeepMask trained with 1.3k interactions (right). The top three rows show examples where our method predicts accurate masks. The fourth row shows that all methods produce failures for very cluttered scenes. The last row shows that DeepMask produces more false positives at the borders of the workspace.

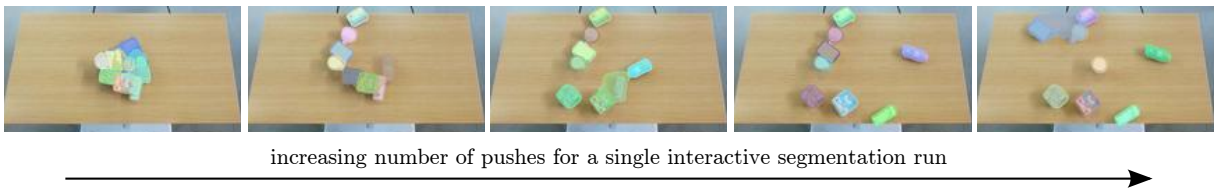

Fig. 6. Exemplary trial of the interactive segmentation experiment. Segmentation accuracy improves with each interaction due to minimization of clutter, see video at `https://bit.ly/38WXKlx`.

D. Interactive Instance Segmentation with Object Separation

In this second experiment we show that combining the paradigm of interactive segmentation with object separation improves overall perception performance at test time. Figure 4 shows that the segmentation performance improves with each interaction up to 23.7 average precision points compared to a passive segmentation where the initial scene is not changed by the robot. In total the robot interactively segmented 20 scenes consisting of unknown objects. Figure 6 qualitatively shows the improved instance segmentation after each push interaction. The push actions were chosen based on a pre-trained CNN for object separation from own prior work [35]. The results suggest that following an interactive perception strategy substantially improves the segmentation performance at test time and in addition provides the self-supervised data for training a more accurate network.

E. Qualitative Results

Figure 5 shows the outputs generated by the different methods. Our method generalizes and segments novel objects effectively. Qualitatively, our method performs similarly to DeepMask. All methods produce failures for highly cluttered scenes but our method is less prone to clutter.

V. CONCLUSIONS

In this work, we presented a self-supervised transfer learning approach for instance segmentation that leverages physical robot interaction with its environment to automatically generate a training dataset for adapting pre-trained networks to the current environment. Instead of labeling object masks in an expensive manual procedure, our robot learns to generate object masks by observing the outcome of its own interaction with objects. As the main supervision

signal we use motion information from pushing objects. Our results suggest that fine-tuning a pre-trained model with the automatically labeled data substantially improves the segmentation performance. The more high-level take-home message from this is that robots can in fact improve their perception performance, achieved by manually labeled large-scale datasets, by physically interacting with their environment. We also showed that we can further improve the performance of our method if we leverage recent algorithms for training with noisy labels. In future work, we will explore how to fine-tune our model in a new environment without forgetting the previously learned knowledge, particularly when the adaptation is carried out for longer periods of time.

ACKNOWLEDGMENTS

This work was partly funded by the German Research Foundation (DFG) under the priority program Autonomous Learning SPP 1527, under grant number EXC1086 as well as by the Federal Ministry of Education and Research (BMBF) under Deep PTL and OML. We thank Oier Mees for his help. We also thank the anonymous reviewers for suggestions.

REFERENCES

[1] P. Jund, A. Eitel, N. Abdo, and W. Burgard, "Optimization Beyond the Convolution: Generalizing Spatial Relations with End-to-End Metric Learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 4510–4516, iSSN: 2577-087X.

[2] M. Shridhar and D. Hsu, "Interactive Visual Grounding of Referring Expressions for Human-Robot Interaction," in *Robotics: Science and Systems XIV*, vol. 14, Jun. 2018. [Online]. Available: http://www.roboticsproceedings.org/rss14/p28.html

[3] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[4] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze, "Segmentation of unknown objects in indoor environments," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 4791–4796, iSSN: 2153-0858.

[5] P. O. O. Pinheiro, R. Collobert, and P. Dollar, "Learning to Segment Object Candidates," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 1990–1998. [Online]. Available: http://papers.nips.cc/paper/5852-learning-to-segment-object-candidates.pdf

[6] J. Kenney, T. Buckley, and O. Brock, "Interactive segmentation for manipulation in unstructured environments," in *2009 IEEE International Conference on Robotics and Automation*, May 2009, pp. 1377–1382, iSSN: 1050-4729.

[7] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive Perception: Leveraging Action in Perception and Perception in Action," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, Dec. 2017.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[9] P. Fitzpatrick, "First contact: an active vision approach to segmentation," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 3, Oct. 2003, pp. 2161–2166 vol.3.

[10] M. Björkman and D. Kragic, "Active 3d scene segmentation and detection of unknown objects," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 3114–3120, iSSN: 1050-4729.

[11] D. Schiebener, A. Ude, J. Morimoto, T. Asfour, and R. Dillmann, "Segmentation and learning of unknown objects through physical interaction," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, Oct. 2011, pp. 500–506, iSSN: 2164-0572.

[12] K. Hausman, F. Balint-Benczedi, D. Pangercic, Z.-C. Marton, R. Ueda, K. Okada, and M. Beetz, "Tracking-based interactive segmentation of textureless objects," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 1122–1129, iSSN: 1050-4729.

[13] H. van Hoof, O. Kroemer, and J. Peters, "Probabilistic Segmentation and Targeted Exploration of Objects in Cluttered Environments," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1198–1209, Oct. 2014.

[14] T. Patten, M. Zillich, and M. Vincze, "Action Selection for Interactive Object Segmentation in Clutter," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 6297–6304, iSSN: 2153-0858.

[15] D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, and A. Stentz, "Perceiving, learning, and exploiting object affordances for autonomous pile manipulation," *Autonomous Robots*, vol. 37, no. 4, pp. 369–382, Dec. 2014. [Online]. Available: https://doi.org/10.1007/s10514-014-9407-y

[16] D. Pathak, Y. Shentu, D. Chen, P. Agrawal, T. Darrell, S. Levine, and J. Malik, "Learning Instance Segmentation by Interaction," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2018, pp. 2123–21 233, iSSN: 2160-7508.

[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)*, May 2016, pp. 3406–3413.

[18] A. Murali, L. Pinto, D. Gandhi, and A. Gupta, "CASSL: Curriculum Accelerated Self-Supervised Learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 6453–6460, iSSN: 2577-087X.

[19] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, Apr. 2018. [Online]. Available: https://doi.org/10.1177/0278364917710318

[20] Y. Chebotar, K. Hausman, Z. Su, G. S. Sukhatme, and S. Schaal, "Self-supervised regrasping using spatio-temporal tactile features and reinforcement learning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2016, pp. 1960–1966, iSSN: 2153-0866.

[21] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to Poke by Poking: Experiential Learning of Intuitive Physics," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 5074–5082.

[22] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning Synergies Between Pushing and Grasping with Self-Supervised Deep Reinforcement Learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 4238–4245, iSSN: 2153-0858.

[23] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, S. Levine, and G. Brain, "Time-Contrastive Networks: Self-Supervised Learning from Video," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1134–1141, iSSN: 2577-087X.

[24] T. Mar, V. Tikhanoff, G. Metta, and L. Natale, "Self-supervised learning of tool affordances from 3d tool representation through parallel SOM mapping," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 894–901.

[25] L. Pinto, D. Gandhi, Y. Han, Y.-L. Park, and A. Gupta, "The Curious Robot: Learning Visual Representations via Physical Interactions," in *Computer Vision – ECCV 2016*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 3–18.

[26] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the Amazon Picking Challenge," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1386–1383.

[27] C. Mitash, K. E. Bekris, and A. Boularias, "A self-supervised learning system for object detection using physics simulation and multi-view pose estimation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 545–551, iSSN: 2153-0866.

[28] E. Pot, A. Toshev, and J. Kosecka, "Self-supervisory Signals for

Object Discovery and Detection," *CoRR*, vol. abs/1806.03370, 2018. [Online]. Available: http://arxiv.org/abs/1806.03370

[29] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where Should I Walk? Predicting Terrain Properties From Images Via Self-Supervised Learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1509–1516, Apr. 2019.

[30] T. Schmidt, R. Newcombe, and D. Fox, "Self-Supervised Visual Descriptor Learning for Dense Correspondence," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 420–427, Apr. 2017.

[31] A. Osep, P. Voigtlaender, J. Luiten, S. Breuers, and B. Leibe, "Large-Scale Object Discovery and Detector Adaptation from Unlabeled Video," *CoRR*, vol. abs/1712.08832, 2017. [Online]. Available: http://arxiv.org/abs/1712.08832

[32] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan, "Learning Features by Watching Objects Move," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6024–6033, iSSN: 1063-6919.

[33] A. Milan, T. Pham, K. Vijay, D. Morrison, A. Tow, L. Liu, J. Erskine, R. Grinover, A. Gurman, T. Hunn, N. Kelly-Boxall, D. Lee, M. McTaggart, G. Rallos, A. Razjigaev, T. Rowntree, T. Shen, R. Smith, S. Wade-McCue, Z. Zhuang, C. Lehnert, G. Lin, I. Reid, P. Corke, and J. Leitner, "Semantic Segmentation from Limited Training Data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1908–1915, iSSN: 2577-087X.

[34] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting Unknown 3d Objects from Real Depth Images using Mask R-CNN Trained on Synthetic Data," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 7283–7290, iSSN: 1050-4729.

[35] A. Eitel, N. Hauff, and W. Burgard, "Learning to Singulate Objects Using a Push Proposal Network," in *Robotics Research*, N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, Eds. Cham: Springer International Publishing, 2020, pp. 405–419.

[36] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Aparicio, and K. Goldberg, "Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics," in *Robotics: Science and Systems XIII*, vol. 13, Jul. 2017. [Online]. Available: http://www.roboticsproceedings.org/rss13/p58.html

[37] I. A. Sucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics Automation Magazine*, vol. 19, no. 4, pp. 72–82, Dec. 2012.

[38] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1647–1655, iSSN: 1063-6919.

[39] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 8527–8537.

[40] C. Szegedy, S. E. Reed, D. Erhan, and D. Anguelov, "Scalable, High-Quality Object Detection," *CoRR*, vol. abs/1412.1441, 2014. [Online]. Available: http://arxiv.org/abs/1412.1441

[41] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to Refine Object Segments," in *Computer Vision - ECCV 2016*, ser. Lecture Notes in Computer Science, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 75–91.

[42] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6596

[43] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 1189–1197. [Online]. Available: http://papers.nips.cc/paper/3923-self-paced-learning-for-latent-variable-models.pdf