One-Shot Shape-Based Amodal-to-Modal Instance Segmentation

Andrew Li[1], Michael Danielczuk[1], Ken Goldberg[1]

***Abstract*— Image instance segmentation plays an important role in mechanical search, a task where robots must search for a target object in a cluttered scene. Perception pipelines for this task often rely on target object color or depth information and require multiple networks to segment and identify the target object. However, creating large training datasets of real images for these networks can be time intensive and the networks may require retraining for novel objects. We propose OSSIS, a single-stage One-Shot Shape-based Instance Segmentation algorithm that produces the target object modal segmentation mask in a depth image of a scene based only on a binary shape mask of the target object. We train a fully-convolutional Siamese network with $800,000$ pairs of synthetic binary target object masks and scene depth images, then evaluate the network with real target objects never seen during training in densely-cluttered scenes with target object occlusions. OSSIS achieves a one-shot mean intersection-over-union (mIoU) of 0.38 on the real data, improving on filter matching and two-stage CNN baselines by 21% and 6%, respectively, while reducing computation time by 50 times as compared to the two-stage CNN due in part to the fact that OSSIS is one-stage and does not require pairwise segmentation mask comparisons.**

I. Introduction

Instance segmentation, the task of producing pixelwise masks of all objects within a scene image, can help a robot spatially and visually process its environment prior to decision-making. Combining an instance segmentation algorithm with a target recognition algorithm can further allow a robot to recognize a goal object among a cluttered scene. Manipulating objects in the scene to uncover and extract the target object, a problem called mechanical search, and robotic pick-and-place are two examples in which instance segmentation plays an important role [7, 20, 26]. In these tasks, segmenting the target object can be the first step to manipulating it. Instance segmentation masks can be further distinguished as *modal* (showing a view of the object within the scene, respecting occlusions) or *amodal* (showing the entire object unobstructed) [19]. *Amodal-to-modal* segmentation is the task of predicting the modal mask of an object given its amodal mask.

Image segmentation within the context of robotics has been approached with a variety of sensory data, including RGB, depth, RGB-D, tactile, and LiDAR data [8, 14, 22, 49]. To reduce data collection and processing costs, *sim-to-real transfer* and *one-shot* methods were developed. Sim-to-real methods train only or mostly on simulated images and thereby reduce dependence on large-scale real datasets. One-shot methods generalize beyond the training object classes, and can additionally prevent failure due to small deviations in an object's appearance.

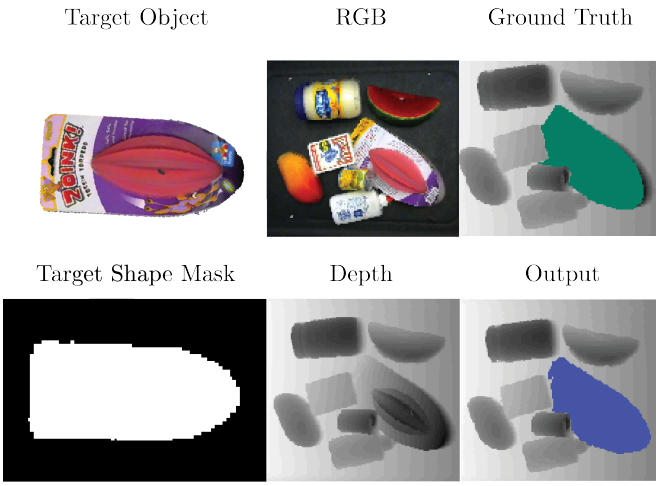

Fig. 1: Shape-based instance segmentation of a packaged dog toy (top-left) within a scene (top-middle). The target binary shape mask (bottom-left) and scene depth image (bottom-middle) are passed into the network, which outputs the blue segmentation mask (bottom-right). For comparison, the ground-truth segmentation mask is in green (top-right). Although the package has a different scale, 2D position, and rotation in the scene, as well as being heavily occluded by other objects, the network is able to segment it using its shape mask.

As compared to these methods, we additionally increase efficiency (i.e., reduce the number of network parameters) and train with a weaker form of supervision – binary shape masks without any texture or color information. We introduce *one-shot shape-based instance segmentation*, in which a network receives only a binary mask of a previously unseen target object shape and a depth image of a heap of objects as input and predicts a modal segmentation mask for the target object. Furthermore, to ease the burden of generating 3D rotated targets through a depth camera or simulator, the binary masks are generated from existing amodal segmentation masks mirroring the image.

We argue that training a network in this way minimizes data labeling cost and guides both domain and object class generalization. Additionally, stronger forms of supervision such as color or depth images from online retailers can be thresholded to create the binary mask input, using standard foreground-background segmentation techniques [11]. This enables its utilization in automation pipelines such as mechanical search [7].

The AUTOLAB at UC Berkeley (automation.berkeley.edu)
{andrewyli, mdanielczuk, goldberg}@berkeley.edu

This paper makes two contributions:

1) One-Shot Shape Instance Segmentor (OSSIS), an algorithm using a Siamese-U-Net [31] trained on 800k synthetic depth images and amodal target shape masks to estimate the target modal segmentation mask in a scene of real objects.
2) Experiments comparing OSSIS to a filter matching baseline and two-stage MaskRCNN + Siamese matching baseline, as well as ablation studies exploring how the augmentations to both the dataset and algorithm affect quantitative performance and computational expense. The algorithm outperforms the filter matching baseline by 21% and the two-stage CNN by 6% in mean intersection-over-union on 6000 images derived from the WISDOM-Real test dataset.

II. Related Work

The approach taken in this paper builds on previous work in the fields of convolutional encoder-decoder neural networks for instance segmentation, image dataset augmentation, and sim-to-real training.

A. Instance Segmentation Methods

Efforts in the field of computer vision towards image segmentation began with region and graph based methods [11, 12, 25, 33]. Commonly, these methods partition the image into subsets of similar intensity or features. Convolutional encoder-decoder neural networks have been more recently found to be effective in localizing and segmenting objects for applications such as autonomous driving and robot grasping [2, 6, 24]. Some networks rely first on a bounding box generator before performing segmentation [14, 23, 35], while others output a confidence map over all pixels in the image and threshold the results to produce the final masks [2, 17, 32]. We leverage the computational advantage of the latter approach, where only one forward pass through a network is required to both localize and segment a target object.

B. Binary Masks as Weak Supervision

Binary masks have been commonly used as a form of weak supervision for 3D reconstruction from single or multiple object views [4, 30]. Yan *et al.* [50] introduce a loss based on consistency of silhouettes from different perspectives, and Gwak *et al.* [13] extend this result by adding an adversarial constraint. Tulsiani *et al.* [46] directly use binary masks or noisy depth images as training inputs, learning a network that can reconstruct 3D objects from these single-view inputs based on a ray consistency loss across multiple views during training. For instance segmentation, Eitel *et al.* [9] and Pathak *et al.* [34] use binary masks as part of a self-supervised pipeline that leverages push and grasp actions to generate training data and improve segmentation across actions. In contrast to these works, we aim to generate our labels entirely in simulation without interaction and focus on segmenting unseen target objects.

C. Sim-to-Real Transfer

Since collecting data for high-quality real world visual inference can often be expensive and time-consuming [27], training on datasets created in simulation and transferring to the real domain requires less manual labor and time [15, 37, 38, 41]. Several approaches have been taken to both decrease the generalization gap between sim and real performance when training on a simulated dataset. In the process of sim-to-real fine-tuning, a network is first trained on a large simulated dataset and then additionally trained on a small real dataset [1, 48]. Domain randomization randomly modifies lighting, pose, and textures in the simulated training dataset to bridge the sim-to-real gap [43, 44]. We choose inputs that have been shown to transfer easily from sim-to-real [8, 29, 39] and augment our dataset with target mask rotations, as binary masks are not affected by changes in lighting or texture.

D. One-Shot Object Detection and Segmentation

In a similar vein, generalizing to previously unseen object classes for detection or segmentation can be useful when data is limited. One-shot methods learn from training datasets that may not contain all the object classes in the evaluation set. Recently, there has been significant interest in both one-shot object detection [16] as well as few-shot [10] or one-shot instance [31, 32] or semantic [36] segmentation. However, in contrast to these methods, we do not leverage a large dataset of labeled RGB images. Instead, we train only on synthetic data with a weaker form of supervision.

Fine-tuning can also be effective for this problem, but many one-shot methods in segmentation omit this for the sake of efficiency and reduced training iterations [40, 47]. We mitigate the need for fine-tuning by using binary shape masks as targets, and depth images to represent scenes as in Mahler *et al.* [28] and Johns *et al.* [18].

III. Problem Statement

Given a binary image of a target object and a depth image containing a single instance of that object, our goal is to find the set of all pixels in the depth image that correspond to a subset of the target object binary mask. Note that the set of pixels in the depth image may be scaled, rotated, and translated as compared to the corresponding pixels in the target object mask.

A. Definitions

- *Target Object*: The object o_t to be segmented, specified as a binary image $\mathcal{I}_t \in \{0,1\}^{H_t \times W_t}$ of the singulated target object.
- *Scene*: A simulated or physical heap of m objects, containing the target object o_t and $m-1$ distractor objects $o_1, \ldots o_{m-1}$.
- *Observation*: A depth image $\mathcal{I}_s \in \mathbb{R}_+^{H_s \times W_s}$ taken from a camera $\mathcal{C}$ at pose $\mathcal{T}_\mathcal{C}$.
- *Target Mask*: The set of pixels $\mathcal{M}_t$ belonging to the target object in the observation.

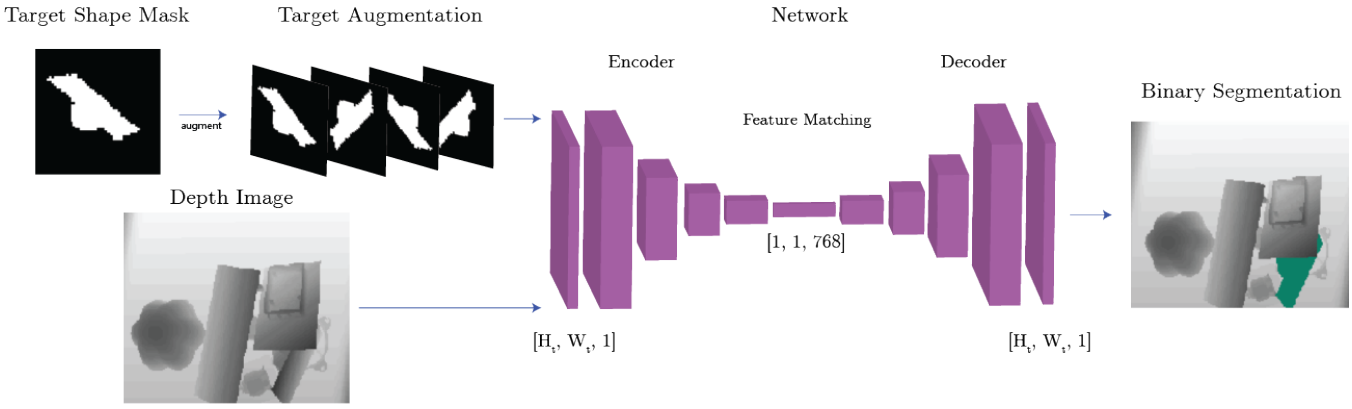

Fig. 2: **Algorithm Overview:** The network takes in a binary shape mask of the target object and a depth image, and produces the modal segmentation mask of the target object in the depth image. We augment the target shape mask before training by rotating the mask randomly between 0 and 360 degrees, and treat each rotated mask as an individual training point.

B. Assumptions

- The set of pixels in the depth image that belong to the target image is not empty (i.e., some part of the target object is visible in the scene).
- There is only one instance of the target object in the scene.

Then, the objective for target object modal instance segmentation is to find a function that estimates the segmentation mask from the scene image and target mask, $f : (\mathcal{I}_s, \mathcal{I}_t) \rightarrow \hat{\mathcal{M}}_t$, such that the pixelwise distance between the estimated segmentation mask $\hat{\mathcal{M}}_t$ and ground-truth segmentation mask $\mathcal{M}_t$ is minimized. Specifically, we employ the commonly used *intersection-over-union* metric (IoU) to quantify pixelwise distance between $\hat{\mathcal{M}}_t$ and $\mathcal{M}_t$, defined as:

$$IoU(\hat{\mathcal{M}}_t, \mathcal{M}_t) = \frac{\left|\hat{\mathcal{M}}_t \bigcap \mathcal{M}_t\right|}{\left|\hat{\mathcal{M}}_t \bigcup \mathcal{M}_t\right|} \tag{1}$$

IV. Methods

A. Dataset Generation

We use the WISDOM-Sim and WISDOM-Real datasets to generate a simulated training dataset, a simulated one-shot test dataset, and real one-shot test dataset [8]. The one-shot datasets contain only scenes and objects that have not been seen during training.

The training dataset consists of 250k depth image, amodal target mask, and ground-truth modal segmentation mask triples $(\mathcal{I}_s, \mathcal{I}_t, \mathcal{M}_t)$ from the WISDOM-Sim dataset [8]. As stated by Danielczuk *et al.*, there are an average of 6.5 object instances per scene image, which yields approximately 325,000 total instances across the 50,000 image dataset. Each object is a member of the Thingiverse object set. We then remove instances where the target object is completely occluded in the scene. Triplets are randomly assigned to the training and validation splits in an 80:20 ratio, leaving us 200k images for training.

The scene images are scaled and cropped such that $W_s = H_s = 384$ to include all objects in the bin and minimize downsampling. For each scene image, we use the modal segmentation masks as ground truth labels. To generate the target shape masks, we collect the amodal segmentation masks for each visible object and apply a rotation between 0 and 360 degrees in the 2D plane uniformly at random to each mask. This process generates m triples per image; one for each visible object in the scene, with all other $m-1$ objects acting as distractors. The amodal target mask is scaled such that $H_t = W_t = 128$, which allows for faster computation.

To evaluate model performance, we create a simulated test dataset comprised of 12.5k similarly-generated triplets containing only scenes and objects that have not been seen during training. This dataset allows evaluation of the model's one-shot performance. We also create a real test dataset comprised of 2.4k real scenes and objects that are also unseen during training. However, as amodal masks cannot be easily determined even by humans from a scene image when there are heavy occlusions, we use RGB images of the objects in the scene singulated on a black background in one of their stable poses. We then binarize the images to create the input target mask for our algorithm. Note that this distinction results in the real image one-shot task being much more difficult than in simulation, as the modal segmentation mask of the target object in the scene may not be a true subset of the amodal target mask given as input (i.e., the target may have an additional *3D rotation* out of the image plane from the input target mask). The real test dataset allows evaluation both of the model's one-shot performance, sim-to-real transfer ability, and capacity to segment novel 3D poses.

B. Dataset Augmentation

To improve the performance of the network on the simulated and real image test datasets, we augment our base training dataset by rotating the amodal mask inputs. We create R-rotated datasets for $R = 1, 2, 4$, where $R = 1$ denotes the base dataset. For our augmented datasets, we form R data points per existing scene image, amodal mask and modal segmask triplet by rotating the target object amodal mask between 0 and 360 degrees uniformly at random R

times. For each rotated amodal mask, we store an unchanged copy of the scene image and segmask as a new triplet. This process results in two augmented datasets totaling 400k and 800k images, respectively. These augmentations expose the model to a wider variety of amodal target poses in the 2D plane. A key observation here is that rotations are the most readily available augmentations to binary shape masks, since techniques such as domain randomization would not affect a texture-less and depth-less mask.

C. Training

We use a convolutional encoder-decoder which takes as input a scene image and a target shape mask, and outputs a modal target object segmentation mask. This is advantageous because it preserves both the high and low level features of input images. We employ a modified Siamese U-Net architecture used by Michaelis *et al.* [31], which was originally introduced by Bromley and LeCun [5]. To better process our larger scene images, we increase the number of layers in the encoder by 1 to 6 and double the number of feature maps to 784. We also insert a dropout layer with factor 0.1 after the last convolutional layer of the encoder to increase amodal robustness. The fully convolutional encoder allows for parallel computation of the low-level feature tensors from the input images. As described by Michaelis *et al*, the final output of the network is produced from feeding the inner and outer products of these tensors into the decoder, which is aided by skip connections to corresponding decoder layers. This network produces a heatmap of predicted confidences in the interval $[0, 1]$ that each pixel belongs to the mask. To produce the final binary segmentation we use a threshold of 0.3 on the heatmap, having optimized for mIoU on thresholds over a range of 0.1 to 0.5 with a step size of 0.05.

The model is trained with the Adam stochastic optimization method with default parameters and initial learning rate of 0.0005 for 10 epochs on a standard 80-20 train-val split of our simulated dataset [21]. On the base simulated dataset, the network converges in approximately 12 hours with batch size 10 on an NVIDIA Titan X GPU. Each forward pass of the network takes 45 ms for a single real scene image and target image pair (averaged over 1000 steps).

V. Experiments

A. Metrics

We measure the effectiveness of each segmentation quantitatively using the mean-intersection-over-union (mIoU) metric, which is the IoU metric defined in Section III averaged across predictions. We define *one-shot sim mIoU* and *one-shot real mIoU* to be the network's performance on the simulated and real test sets, respectively. The first measures the model's generalization to unseen objects and the second measures generalization to unseen objects and ability to transfer from the sim to the real domain.

B. Baselines

We use both classical filter matching and two-stage CNN-based methods as comparisons to evaluate the performance and runtime of our model. These methods are chosen to illustrate the difference in performance and evaluation speed between non-CNN methods, two-stage CNN methods, and our method.

Method	OS Sim	OS Real	Runtime
Filter Matching	0.186	0.171	180 ms
Two-Stage CNN	N/A	0.316	2.5 s
OSSIS (R=1, Mean)	0.357	0.250	45 ms
OSSIS (R=1, Max)	0.357	0.250	45 ms
OSSIS (R=4, Mean)	0.591	0.299	45 ms
OSSIS (R=4, Max)	***0.591***	***0.381***	45 ms

TABLE I: We compare OSSIS trained on datasets with $R = 1$ and $R = 4$ rotations, as well as using the mean and maximum IoUs across the 5 target images, to filter-matching and two-stage CNN baselines using *one-shot mean intersection-over-union* (mIoU) on both the simulated test set and the real test set, both of which are entirely made of objects unseen in training. The baselines have access to depth and color target masks. In comparison, OSSIS only makes use of target object shape information. OSSIS is better able to compensate for target scale, rotation, and translation. Additionally, OSSIS runs 4 times faster compared to filter matching and over 50 times faster than the two-stage CNN.

The filter matching algorithm localizes the target object within the image by measuring cosine similarity scoring for each convolution [3, 42, 45]. The target mask is chosen from the set of masks rotated by angles in $[0, 360]$ with increments of 10 degrees, such that it maximizes mIoU.

The CNN-based approach uses an implementation of SD Mask-RCNN [8] to segment all objects in a given depth scene and a Siamese matching network to select the mask corresponding to the target object [7]. SD Mask-RCNN closely follows the architecture of Mask-RCNN [14], but is adapted for depth images and uses a lighter ResNet-35 backbone. As described by Danielczuk *et al*, the Siamese network combines a fixed ResNet-50 head trained on ImageNet with two dense layers and outputs a probability that two input objects are the same. The Siamese network performs pairwise comparisons between masks output by SD Mask-RCNN to select the most similar target mask. To ensure a fair one-shot comparison between methods, we split the 50 objects in the WISDOM-Real dataset randomly into 10 groups. Then, we train 10 instances of the Siamese network, where for each we choose one of the 10 groups to be a test group and the other 9 groups to be the training groups, resulting in 45 train objects and 5 test objects for each network. When testing, we evaluate each of the networks on each instance of its corresponding 5 test objects in the network and average IoU across all test instances from all networks.

For the one-shot real test dataset, we use 5 images of the target object from different views. For the two-stage CNN, we report the IoU for the mask with the highest match probability across all 5 views. For the filter-matching baseline, we report the maximum IoU across the 5 images.

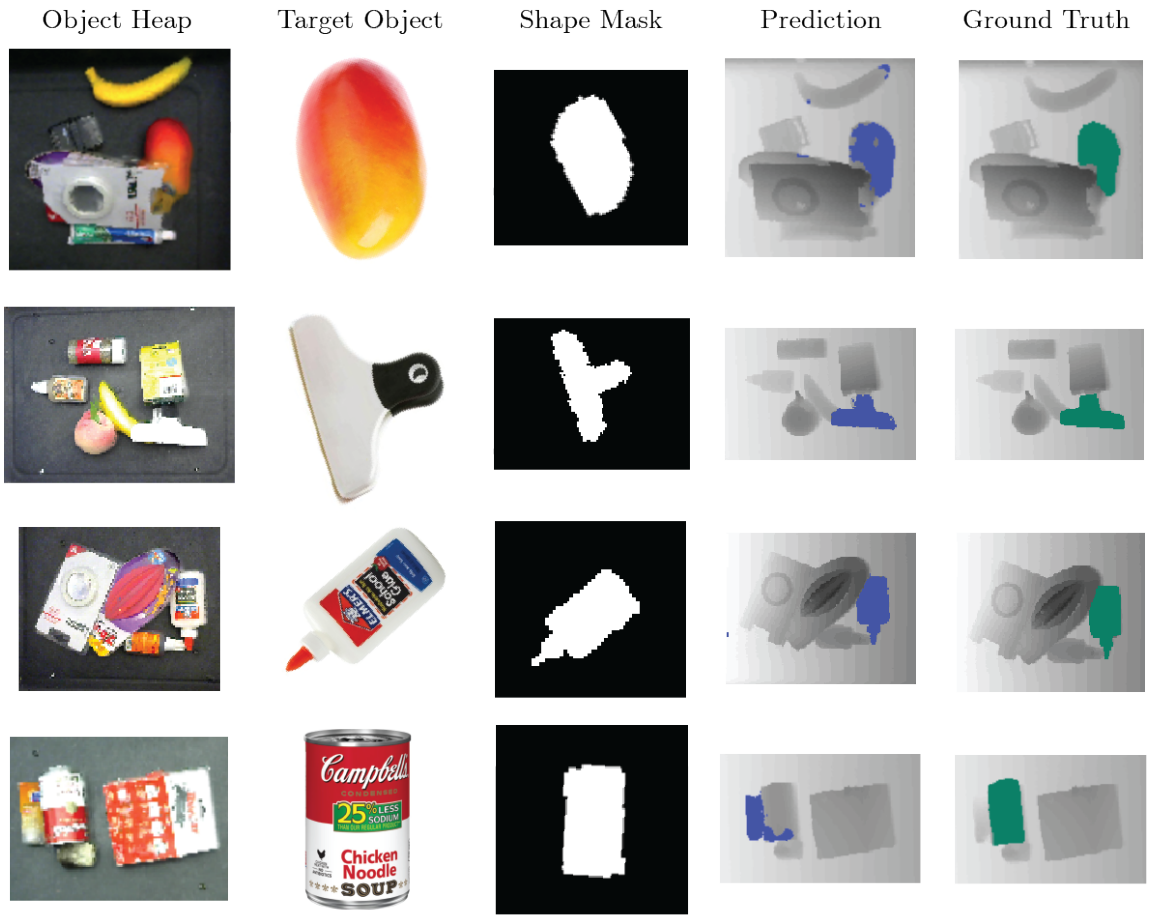

Fig. 3: Qualitative results from applying the algorithm on the test real depth image dataset. We include the color image as well for visual clarity. The first three rows show the ability of the network to segment partially occluded and rotated target objects at different scales. The final row displays a failure mode inherent to the shape-based approach of confounding two similar shapes.

For OSSIS, we report mIoU both when taking the mean and maximum IoU across the 5 images.

C. Results

We evaluate the filter-matching baseline and OSSIS trained on simulated datasets with different numbers of target mask rotations and report one-shot mIoU on both the simulated and real test datasets in Table I. We also report one-shot mIoU for the two-stage CNN baseline on the real test dataset. OSSIS achieves a 21% improvement over the filter-matching baseline and outperforms the two-stage CNN by 6% on the real test dataset. There is little difference in the filter-matching performance on sim and real images, because there is no generalization gap for the filter matching algorithm to bridge. OSSIS also successfully adapts to previously unseen objects in the sim test dataset, with a low one-shot generalization gap of under 4%. Additionally, OSSIS is 4 and 50 times faster than the filter matching and two-stage CNN baselines, respectively, showing a large improvement in efficiency during testing. The efficiency advantage of OSSIS over the two-stage baseline is that OSSIS is one-stage, and does not require pairwise comparisons to extract the target object from the rest of the segmentation masks.

Despite the two-stage CNN baseline having access to color information in addition to shape information, OSSIS is still able to outperform it in the one-shot setting. This result suggests that while the Siamese network may perform very well on objects within its training distribution, it can struggle to generalize to novel objects. Indeed, when we train the Siamese network on all of the objects (albeit only seen in their stable poses), removing the one-shot aspect, it performs very well, achieving 0.69 mIoU.

We find that the combined one-shot and sim-to-real generalization gap for OSSIS is 21%. One reason for this disparity is that the real target images are taken with each object in a stable pose, as mentioned in Section IV, which may be dramatically different from the pose that the object is in when lying on top of or underneath other objects. On real images, oversegmentation tends to occur more frequently, especially with similarly smooth or rectangular objects. Additionally, we find that the network may confuse two objects with very similar, regular shapes (such as a rectangular prism or sphere), especially if there are multiple distractor objects with this shape in the same scene as the target object. The network shows robustness to change in pose and scale on

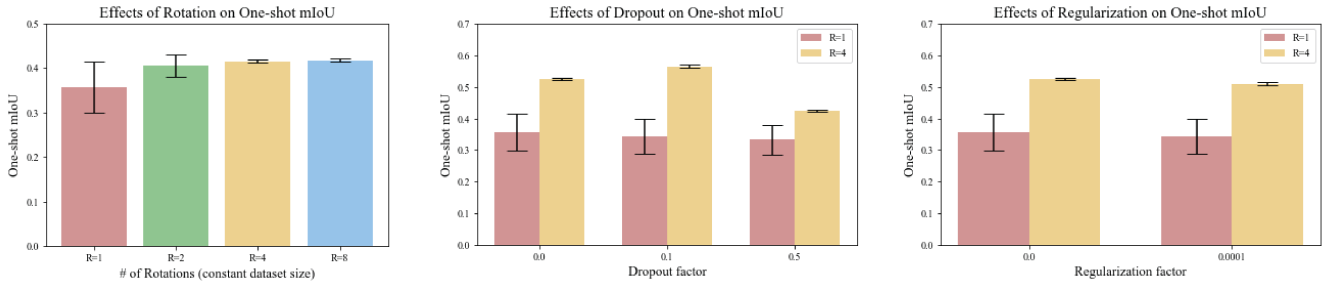

Fig. 4: We measure the effects of rotating target masks, dropout, and regularization on network performance. We control for dataset size and number of unique scene images, and train a model on each resulting dataset. Using four rotations per target mask ($R = 4$) is the most effective in increasing mIoU and reducing variance while also having a low dataset generation cost. Applying L2 penalty regularization does not improve the output of the network. We find that slight dropout applied to the last layer is effective in increasing the one-shot sim mIoU.

both the sim and real datasets.

A visual study of segmentation successes and failures is shown in Figure 3. In the first row, we see the partially occluded mango successfully segmented amongst several distractor objects. The large bag clip in the second row is also successfully segmented, and is dramatically rotated and scaled in the scene compared to its target pose. The final row shows an inherent failure mode: the cylindrical nature of the Campbell soup can is not represented by the target shape mask and the network mistakes the partially occluded lotion for the can, as both shapes are rectangular.

D. Ablations

We characterize both the effect of augmenting the dataset with rotations and the effect of dataset size on network performance. Figure 4 suggests that as the total dataset size increases by adding rotations, so does both the validation and one-shot mIoU. Rotating twice (R=2) improves one-shot mIoU significantly but still has high variance, indicating good performance on some images but failing to segment others almost entirely. Even though the R=2 dataset does not present new scene data to the model, it shows significant improvement over the base dataset by improve both mean and variance of mIoU. At R=8, the improvement is marginal. Because the cost of generating the R=8 dataset is double that of the R=4 dataset for the *training* datasets, which are not restricted in size, we use four rotations to generate the training dataset used in the final results.

We see relatively little difference $< 1\%$ between validation and one-shot results, likely because the generic nature of the target mask lends itself to being applicable to objects in the test split. It is important to note that there is an increase in the gap between one-shot and validation mIoUs as the number of rotations increases. This may potentially be attributable to slight overfitting to the scenes in the training set; having additional rotated shape masks does not preclude overfitting given that the additional scenes still contain objects only from the train split.

To demonstrate the effect of 2D rotation augmentations beyond the increased dataset size, we compared model performance across datasets with constant size (i.e., same number of training triplets) that contained different numbers of unique scenes and target rotations. For example, the dataset with 4 rotations of the target object contained 4x fewer images per scene than the original dataset with a single target object rotation. Figure 4 shows the results. Under this setup, we found that augmenting by rotating four times yielded the best performance, suggesting that diversity in target object rotations for a given scene was more important in training than more views of a scene (e.g., different camera poses for the same arrangement of objects).

We additionally perform experiments to reduce network generalization error when evaluating on either one-shot dataset. The architecture changes from the original Michaelis network, which amount to deepening the filters and adding two additional encoder layers, improve mIoU performance by 0.045 on the real dataset. While we find L2 penalty to have no positive effect on performance, dropout at the last convolutional layer improves one-shot sim mIoU. This is potentially due to the amodality/modality disparity between the scene and target inputs; dropout at low feature map levels can allow for robustness against large occlusion of the object in the scene. We note that too high of a dropout factor also leads to severe mIoU loss, because the network begins to be unable to correctly segment even simple shapes. Using these ablations, we determine effective hyperparameters for optimizing mIoU performance in our final results.

VI. Conclusions and Future Work

We present OSSIS, an algorithm trained entirely on simulated binary target masks and depth images that predicts modal masks for novel target objects in real images, even in the presence of rotations, scale differences, and occlusions. We intend these results to be a first step for this difficult problem of one-shot shape-based instance segmentation, and show that using binary target masks can allow for sim-to-real transfer and can be easily generated from stronger forms of supervision across datasets. Experiments suggest that OSSIS outperforms a state-of-the-art convolutional method by 6% in mIoU.

In future work, we will continue to address the disparity between one-shot sim and real images and further explore

the impact of target depth information as compared to the binary mask supervision we use here. lthough our focus in this paper was an application of one-shot shape-based segmentation to the problem of mechanical search, investigating its impact to other problems in industrial robotics will require some reformulation of the problem statement. For instance, instead of looking for a single target object as is appropriate for mechanical search, training on multi-class, multi-object labels could yield a more effective method for segmenting street scenes for autonomous driving.

Another natural extension of this work is to incorporate multiple target views simultaneously to improve segmentation results. While our current formulation assumes a single physical camera to capture the target object mask, in a sophisticated pipeline multiple camera angles could provide more target masks simultaneously. These target masks could be batched together as input to the network, as opposed to just passing in one mask. Our initial experiments attempting to batch target masks with the encoder-decoder network yielded inferior results, but an alternative method of aggregating target mask information may improve segmentation results.

ACKNOWLEDGMENTS

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab. The authors were supported in part by donations from Google. This material is based on work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1752814. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Sponsors. We thank our colleagues and collaborators who provided helpful feedback, code, and suggestions, especially Andrew Lee, Gokul Swamy, Thomas Liao, Henry Zhu.

REFERENCES

[1] J. van Baar, A. Sullivan, R. Cordorel, D. Jha, D. Romeres, and D. Nikovski, "Sim-to-real transfer learning using robustified controllers in robotic tasks involving complex dynamics," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 6001–6007.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[3] K. Briechle and U. D. Hanebeck, "Template matching using fast normalized cross correlation," in *Optical Pattern Recognition XII*, International Society for Optics and Photonics, vol. 4387, 2001, pp. 95–102.

[4] A. Broadhurst, T. W. Drummond, and R. Cipolla, "A probabilistic framework for space carving," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, IEEE, vol. 1, 2001, pp. 388–393.

[5] J. Bromley, J. W, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a siamese time delay neural network.," *Int. Journal of Pattern Recognition and Artificial Intelligence*, 1993.

[6] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *Lecture Notes in Computer Science*, pp. 833–851, 2018.

[7] M. Danielczuk, A. Kurenkov, A. Balakrishna, M. Matl, D. Wang, R. Martın-Martın, A. Garg, S. Savarese, and K. Goldberg, "Mechanical search: Multi-step retrieval of a target object occluded by clutter," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2019, pp. 1614–1621.

[8] 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 *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2019, pp. 7283–7290.

[9] A. Eitel, N. Hauff, and W. Burgard, "Self-supervised transfer learning for instance segmentation through physical interaction," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 4020–4026.

[10] Z. Fan, J.-G. Yu, Z. Liang, J. Ou, C. Gao, G.-S. Xia, and Y. Li, "Fgn: Fully guided network for few-shot instance segmentation," *arXiv preprint arXiv:2003.13954*, 2020.

[11] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[12] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun, "Bottom-up segmentation for top-down detection," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[13] J. Gwak, C. B. Choy, M. Chandraker, A. Garg, and S. Savarese, "Weakly supervised 3d reconstruction with adversarial constraint," in *Int. Conf. on 3D Vision (3DV)*, IEEE, 2017, pp. 263–272.

[14] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2017.

[15] Z.-W. Hong, C. Yu-Ming, S.-Y. Su, T.-Y. Shann, Y.-H. Chang, H.-K. Yang, B. H.-L. Ho, C.-C. Tu, Y.-C. Chang, T.-C. Hsiao, *et al.*, "Virtual-to-real: Learning to control in visual semantic segmentation," *arXiv preprint arXiv:1802.00285*, 2018.

[16] T.-I. Hsieh, Y.-C. Lo, H.-T. Chen, and T.-L. Liu, "One-shot object detection with co-attention and co-excitation," in *Proc. Advances in Neural Information Processing Systems*, 2019, pp. 2721–2730.

[17] V. Iglovikov, S. Seferbekov, A. Buslaev, and A. Shvets, "Ternausnetv2: Fully convolutional network for instance segmentation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[18] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 4461–4468.

[19] G. Kanizsa, *Organization in vision: Essays on Gestalt perception*. Praeger Publishers, 1979.

[20] M. U. Khalid, J. M. Hager, W. Kraus, M. F. Huber, and M. Toussaint, "Deep workpiece region segmentation for bin picking," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, IEEE, 2019, pp. 1138–1144.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] W. Kuo, A. Angelova, J. Malik, and T.-Y. Lin, "Shapemask: Learning to segment novel objects by refining shape priors," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 9207–9216.

[23] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp, "Image segmentation with a bounding box prior," *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2009.

[24] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. Journal of Robotics Research (IJRR)*, vol. 34, no. 4-5, pp. 705–724, 2015.

[25] A. Levin and Y. Weiss, "Learning to combine bottom-up and top-down segmentation," *Lecture Notes in Computer Science*, pp. 581–594, 2006.

[26] 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," *Int. Journal of Robotics Research (IJRR)*, vol. 37, no. 4-5, pp. 421–436, 2017.

[27] 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," *Lecture Notes in Computer Science*, pp. 740–755, 2014.

[28] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," in *Proc. Robotics: Science and Systems (RSS)*, 2017.

[29] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, eaau4984, 2019.

[30] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan, "Image-based visual hulls," in *Conf. on Computer graphics and interactive techniques*, 2000, pp. 369–374.

[31] C. Michaelis, M. Bethge, and A. S. Ecker, "One-shot segmentation in clutter," *arXiv preprint arXiv:1803.09597*, 2018.

[32] C. Michaelis, I. Ustyuzhaninov, M. Bethge, and A. S. Ecker, "One-shot instance segmentation," *arXiv preprint arXiv:1811.11507*, 2018.

[33] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1452–1458, 2004.

[34] D. Pathak, Y. Shentu, D. Chen, P. Agrawal, T. Darrell, S. Levine, and J. Malik, "Learning instance segmentation by interaction," in *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2042–2045.

[35] M. Rajchl, M. C. H. Lee, O. Oktay, K. Kamnitsas, J. Passerat-Palmbach, W. Bai, M. Damodaram, M. A. Rutherford, J. V. Hajnal, B. Kainz, and et al., "Deepcut: Object segmentation from bounding box annotations using convolutional neural networks," *IEEE Transactions on Medical Imaging*, vol. 36, no. 2, pp. 674–683, 2017.

[36] H. Raza, M. Ravanbakhsh, T. Klein, and M. Nabi, "Weakly supervised one shot segmentation," in *IEEE Int. Conf. on Computer Vision Workshops*, 2019.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[38] A. A. Rusu, M. Vecerik, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," in *Conf. on Robot Learning (CoRL)*, 2017.

[39] D. Seita, N. Jamali, M. Laskey, A. K. Tanwani, R. Berenstein, P. Baskaran, S. Iba, J. Canny, and K. Goldberg, "Deep transfer learning of pick points on fabric for robot bed-making," *arXiv preprint arXiv:1809.09810*, 2018.

[40] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," *Proc. British Machine Vision Conference (BMVC)*, 2017.

[41] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[42] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1442–1468, 2013.

[43] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2017, pp. 23–30.

[44] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 969–977.

[45] D.-M. Tsai and C.-T. Lin, "Fast normalized cross correlation for defect detection," *Pattern Recognition Letters*, vol. 24, no. 15, pp. 2625–2631, 2003.

[46] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2019.

[47] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, "Matching networks for one shot learning," in *Proc. Advances in Neural Information Processing Systems*, 2016, pp. 3630–3638.

[48] G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, and et al., "Interactive medical image segmentation using deep learning with image-specific fine tuning," *IEEE Transactions on Medical Imaging*, vol. 37, no. 7, pp. 1562–1573, 2018.

[49] Y. Wang, T. Shi, P. Yun, L. Tai, and M. Liu, "Pointseg: Real-time semantic segmentation based on 3d lidar point cloud," *arXiv preprint arXiv:1807.06288*, 2018.

[50] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision," in *Proc. Advances in Neural Information Processing Systems*, 2016, pp. 1696–1704.