Towards Generalization of One-Shot Amodal-To-Modal Instance Segmentation Using Shape Masks

Andrew Li

Towards Generalization of One-Shot Amodal-To-Modal Instance Segmentation Using Shape Masks

by

Andrew Li

A thesis submitted in partial satisfaction of the

requirements for the degree of

Fifth Year Master's

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Chair
Professor Ren Ng

Spring 2020

The thesis of Andrew Li, titled Towards Shape-Based Generalization of Instance Segmentation, is approved:

Chair ______ Digitally signed by Ken Goldberg
DN: cn=Ken Goldberg, o, ou,
email=goldberg@berkeley.edu,
c=US
Date: 2020.05.28 20:53:08 -08'00' ______ Date May 29, 2020

Yiren Ng Ren Ng ______ Date May 29, 2020

______ Date ______

University of California, Berkeley

Towards Generalization of One-Shot Amodal-To-Modal Instance Segmentation Using Shape Masks

Abstract

Towards Generalization of One-Shot Amodal-To-Modal Instance Segmentation Using Shape Masks

by

Andrew Li

Fifth Year Master's in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Ken Goldberg, Chair

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. In this thesis, we propose a single-stage One-Shot Shape-based Instance Segmentation algorithm (OSSIS) 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. The method 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 50x as compared to the two-stage CNN. This is achieved even though the real target masks are in color and the training scenes are in depth, due to the binarization of the shape target masks. Training and testing on multiple mediums of data has both the potential to shore up data deficiencies and mitigate retraining of networks.

To Yuanqian Li and Chun Feng, my ever-loving parents. To Josh - couldn't ask for a better brother.

Contents

List of Figures

List of Tables

Acknowledgments

I am very grateful for Professor Ken Goldberg and his ever-present support and guidance throughout my time at Cal. The opportunity to work with him has shaped my research skills and curiosity in a unique and valuable way. Michael Danielczuk has been an incredible mentor - insightful, personal, patient, and generous with his time. Thank you to Andrew Lee for being a great research partner in our earlier project and for getting me involved with the lab.

From day one my parents have been there for me, even though we are three thousand miles apart. Thank you, Mom and Dad, for everything. I love you very much. My brother has constantly reached across the distance and given me much hope for the future. I couldn't be more excited to see you next year!

There are many both at Cal and back home in Boston I owe my happiness to. Thank you, Andrew, for being the best roommate I could have asked for; a dual in many ways but a friend in one. Thank you, Jackson, for your unwavering loyalty; I know we'll keep in touch wherever we go. Thank you, Matthew and Gokul, for your constant support, love, and intuition on how I feel. Thank you, Dan, for forever inspiring me with your attitude and inner strength. Thank you, Celine, for forever keeping me grounded. Thank you, Kimmy; you made college much more bearable with your companionship. Thank you Henry and Angela, for our fantastic conversations - good luck next year! Cal Jazz Choir, I feel very lucky to have met you all in my final year, and wouldn't be here without you. The same goes for Sam - I've never met a gentler soul. Thank you to the friends I've made in my time at Cal, the golden ones at home, and my teachers throughout the years.

Thank you all.

Chapter 1

Introduction

Within the last decade, efforts in the field of robotics towards localizing and segmenting target objects within cluttered scenes have incorporated deep learning methods to great success in both academia and industry. For example, the impact of recent segmentation breakthroughs are seen in *mechanical search*, the task of manipulating objects in the scene to uncover and extract a target object. The specific variant of segmentation we discuss is *instance segmentation*, where the goal is to produce a segmentation mask of all target objects in a scene given both the object and the scene image. We refer to a singulated mask of the target object as an *amodal* mask and an in-scene mask as a *modal* mask. As such, modal masks take occlusions by other objects into account while amodal masks do not. Because the target object is often occluded by other objects (known as "distractors") in the scene, the desired modal mask can appear different from the given amodal target mask.

Additionally, the difficulty of translating the success of a learning method in simulated environments to the real world lies heavily in either hand-labeling a real world dataset or adapting a segmentation model trained on simulated data to a new, real set of objects. Because of the potential time and expense incurred with the former option, robotics and computer vision research have seen the rapid development of novel methods to perform the latter approach, known as *sim-to-real* transfer of deep networks [26, 47]. Many challenges arise in performing sim-to-real transfer for the task of instance segmentation, including reconciling different object poses and sets of objects. For instance, a target object may have a dramatically different scale and pose in a provided target image as compared to its scale and pose in the cluttered scene image. Additionally, for the task of *one-shot* instance segmentation, an object seen at test time by a segmentation network may not be a member of a training time class. In such settings, this prohibits using a standard pixel-wise classification to produce the segmentation mask.

We present a learning-based approach to instance segmentation in this thesis, addressing the issues of modal object occlusion, varying object pose, sim-to-real transfer, and the one-shot setting using shape masks. The chapter on one-shot shape-based instance segmentation is part of a joint effort between Andrew Li, Michael Danielczuk, and Professor Ken Goldberg, and was submitted to CASE 2020 with acceptance pending. My contributions and the contributions of everyone in the project can be found in Section 3.4.

This thesis contributes:

1. A formulation of the generalized instance segmentation problem and proposed solution.
2. A one-shot shape-based instance segmentation (OSSIS) method using a Siamese-U-Net for estimating the target modal segmentation mask in a scene of real objects.
3. Experiments comparing OSSIS to a filter matching baseline and two-stage MaskRCNN + Siamese matching baseline. 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.
4. Ablation studies exploring how the augmentations to both the dataset and algorithm affect quantitative performance and computational expense.

Chapter 2

Related Work

2.1 Instance Segmentation Methods

Efforts in the field of computer vision towards image segmentation began with region and graph based methods [13, 25, 35, 14]. Commonly, these methods partition the image into subsets of similar intensity or features. These methods typically do not require large training datasets, and can be chosen based on the domain at hand [54].

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 [16, 37, 23], while others output a confidence map over all pixels in the image and threshold the results to produce the final masks [2, 33, 19]. 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.

Fully convolutional networks have been found to be effective for semantic segmentation [9]. While our task aims at segmenting individual objects, one similarity to semantic segmentation is that we have a known number of target instances and classes per image, our classes being “the object” and “not the object”. As opposed to standard instance segmentation, the method presented in this thesis instead targets singular objects. Convolutional Siamese neural networks provide a unique paired structure that quantifies similarity between two input images, namely the scene image and the target object mask [22, 5]. Furthermore, Ronneberger *et al.* provide a rich restructuring in the decoder stage through upsampling [39]. We employ the strengths of both approaches in our method, and additionally choose a deep learning baseline with a Siamese network.

2.2 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 [31, 4]. Yan *et al.* [53] introduce a loss based on consistency of silhouettes from different perspectives, and Gwak *et al.* [15] extend this result by adding an adversarial constraint. Tulsiani *et al.* [50] 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.* [11] and Pathak *et al.* [36] 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. Because real camera intrinsics and scenes are mimicked in simulation, object shape is a consistent signal that carries over from sim to real.

2.3 Segmentation Datasets and WISDOM

Several image datasets have been developed within the last two decades towards motivating and aiding the development of object segmentation methods. Martin *et al.* [30] and Nene *et al.* [34] created image datasets used in the aforementioned graph-based methods such as Felzenszwalb *et al.* [13]. The creation of larger segmentation datasets, in turn, fueled the success of more recent deep learning based methods such as those mentioned in Section 2.1. ImageNet and Microsoft's Common Objects in Context are two such widely used datasets that have helped standardize segmentation performance benchmarks [10, 27, 40]. For segmentation in industrial settings, the Warehouse Instance Segmentation Dataset for Object Manipulation (WISDOM) provides both simulated and real test scenes and objects [8]. Because we aim to perform sim-to-real instance segmentation in the context of mechanical search or bin-picking, for experiments we utilize WISDOM.

2.4 Sim-to-Real Transfer

Since collecting data for high-quality real world visual inference can often be expensive and time-consuming, training on datasets created in simulation and transferring to the real domain requires less manual labor and time [41, 17, 45, 42]. 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, 52]. Domain randomization randomly modifies lighting, pose, and textures in the simulated training dataset to bridge the sim-to-real gap [47, 48]. We choose inputs that have been shown to transfer easily from sim-to-real [8, 43, 29] and augment our dataset with target mask rotations, as binary masks are not affected by changes in lighting or texture.

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 [44, 51]. 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.* [20].

2.5 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 [18] as well as few-shot [12] or one-shot instance [32] or semantic [38] segmentation. However, in contrast to these methods, we do not leverage a large dataset of labeled real RGB images. Instead, we train only on synthetic data with a weaker form of supervision.

Chapter 3

One-Shot Depth Instance Segmentation

3.1 Problem Statement

Definitions

Let $\mathcal{O}$ be a set of objects. Let $\mathcal{S}$ be a set of scenes comprised of objects drawn from $\mathcal{O}$. Let $s_i \in \mathcal{S}$ be a scene consisting of m_i objects arranged in a pile, where we desire to locate and segment the target object $o_t \in \mathcal{O}$ among the $m_i - 1$ distractor objects $o_1, \dots o_{m_i-1} \in \mathcal{O}$. Let depth camera $\mathcal{C}$ be positioned with pose $\mathcal{T}_\mathcal{C}$, such that it observes the depth image $\mathcal{I}^{(s_i)} \in \mathbb{R}_+^{H_s \times W_s}$. Based on the camera's pose and how the distractor objects occlude the target object, within $\mathcal{I}_s$ there exists a set of pixels $\mathcal{M}_t \subset \mathcal{I}^{(s_i)}$ belonging to the visible portion of the target object. We refer to $\mathcal{M}_t$ as the "modal segmentation mask", or modal segmask. By our assumptions, this pixel set $\mathcal{M}_t$ is connected and unique; that is, o_t is the only target object and is in one continuous piece.

Furthermore, let each object o_t be associated with a set $\mathcal{A}_t$ of k_t amodal target masks $\mathcal{A}_t := \{A_j | j \in 1, \dots, k_t\}$, each representing a singulated instance of o_t in an arbitrary pose. We refer to any A_j as an "amodal target 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 amodal target mask.

Then, the objective for target object modal instance segmentation is to find a function that estimates the modal segmentation mask $\mathcal{M}_t$ for any given (s_i, o_t) and their respective images $(\mathcal{I}^{(s_i)}, A_j)$, such that the pixelwise distance between the function's output and ground-truth target mask $\mathcal{M}_t$ is minimized. We call this function

$$f : (\mathcal{I}^{(s_i)}, A_j) \to \hat{\mathcal{M}}_t \tag{3.1}$$

denoting the estimated segmentation mask as $\hat{\mathcal{M}}_t$.

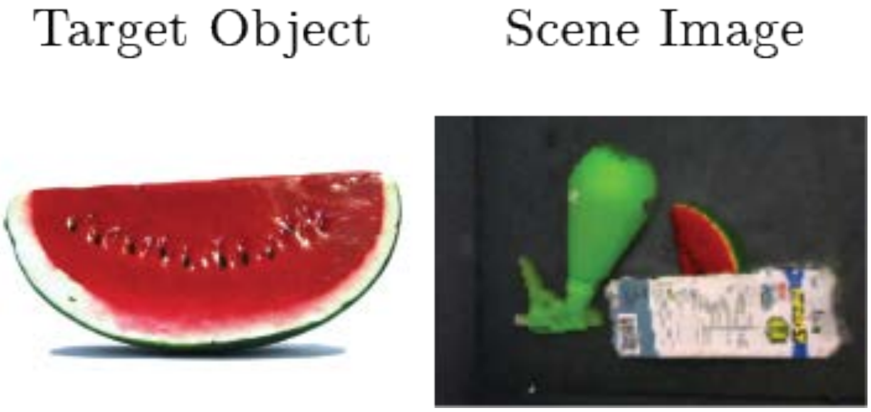

Figure 3.1: An example target object and scene from the WISDOM-Real dataset [7], in full color for visualization purposes. In the scene (right), the watermelon (left) is in a different pose and at a different scale compared to its target image, as well as being heavily occluded.

Metrics and Objective

To quantify the pixelwise distance between $\hat{\mathcal{M}}_t$ and $\mathcal{M}_t$, we employ the commonly used *intersection-over-union* metric (IoU). This is the ratio of correctly identified pixels to the total of correctly identified, extraneously identified pixels, and missed pixels of the modal target object in the scene. It is mathematically defined as

$$IoU(\hat{\mathcal{M}}_t, \mathcal{M}_t) = \frac{\left|\hat{\mathcal{M}}_t \bigcap \mathcal{M}_t\right|}{\left|\hat{\mathcal{M}}_t \bigcup \mathcal{M}_t\right|}. \tag{3.2}$$

A method f achieves a mean intersection-over-union (mIoU) over a validation or test dataset equal to the mean of its IoU on each pair $\mathcal{I}^{(s_i)}, A_j$. Equipped with this metric, we restate our earlier objective as

$$f = \arg\max_{g} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \sum_{j=1}^{m_i} \frac{1}{k_t} \sum_{k=1}^{k_t} \mathrm{IoU}(g(\mathcal{I}^{(s_i)}, A_k), \mathcal{M}_j) \tag{3.3}$$

with the right hand side representing the average IoU over all scenes, all objects, and all target masks of each object.

We additionally use an mIoU-like metric calculated over all scenes and all objects, which takes the maximum IoU out of the target masks associated with an object. This is equivalently defined as

$$f = \arg\max_{g} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{m_i} \sum_{j=1}^{m_i} \max_{k} \mathrm{IoU}(g(\mathcal{I}^{(s_i)}, A_k), \mathcal{M}_j) \tag{3.4}$$

This is meant to reflect the availability of all target masks during real-world evaluation, and we additionally report our evaluations of OSSIS and the baselines on this metric.

3.2 Methods

Dataset Generation

Training Dataset

We use the WISDOM-Sim to generate a simulated dataset to train the deep network. The training dataset consists of 250k depth image, amodal target mask, and ground-truth modal segmentation mask triples $(\mathcal{I}^{(s_i)}, \mathcal{A}_j, \mathcal{M}_t)$ from the WISDOM-Sim dataset [8]. The scene images are scaled and cropped such that $W_s = H_s = 384$ to include all objects in the bin and minimize downsampling. As hinted at by the mIoU objective in Section 3.1, for each scene we iterate through each object, treating it as the target object for that iteration. We scale, rotate and crop the amodal segmask of the object and use it as the target mask $\mathcal{A}_j$, such that the object is centered in the mask image and maintains a border of between 20 and 60 pixels in all directions (see Section 3.2). We additionally apply a rotation uniformly at random between 0 and 360 degrees in the 2D plane to each mask. This process generates m_i triples per image; one for each visible object in the scene, with all other $m_i - 1$ objects acting as distractors. The amodal target masks are scaled such that $H_t = W_t = 128$, which allows for faster computation. The modal segmentation masks, as they provide the ground-truth label for the target object pixels in the scene, have dimension (H_s, W_s).

It is important to note that we do not perform 3D transformations on the object pose in training; *however*, the testing datasets described in Section 3.2 fully evaluate the model's capacity to segment objects with 3D transformations. The significance of this is that our

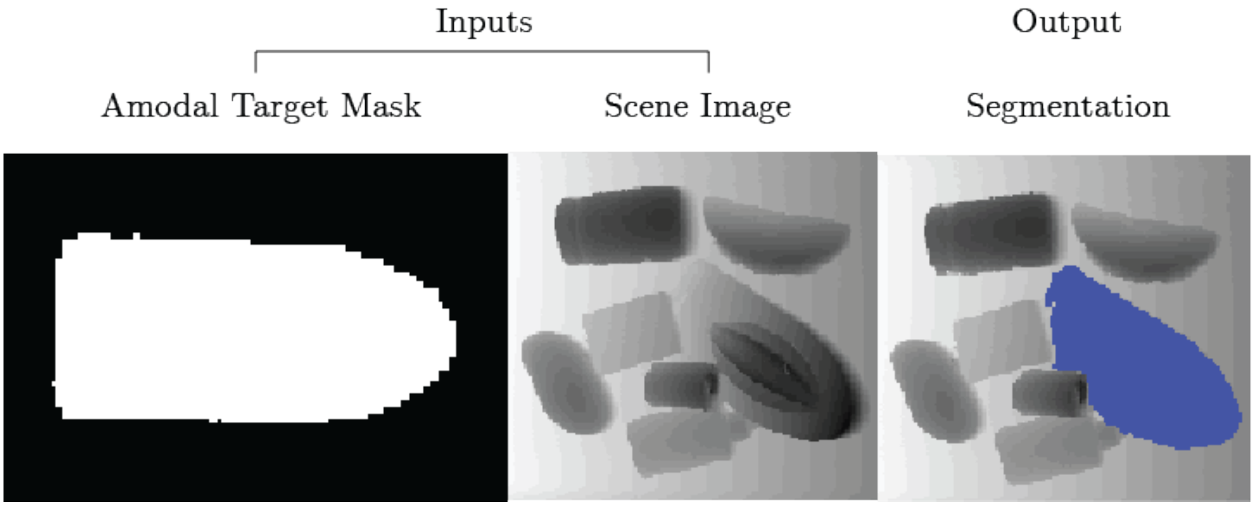

Figure 3.2: Inputs and output for a network f. We choose the amodal target mask to be binary, notably, as opposed to using a depth mask. This allows it to be derived from many common image mediums such as color or depth. The scene image remains encoded in the original image medium.

training method only requires existing 2D images and masks, while still being able to perform well on real objects in varying 3D poses. This means that the

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. 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.

Algorithm 1 Training Dataset Generation Procedure

```
for $i = 1, 2, \ldots, n$ do
    $\mathcal{I}^{(s_i)} \leftarrow$ read_from_file($s_i$)
    Separate existing amodal segmask associated with $s_i$ into $\hat{\mathcal{A}}_1, \ldots, \hat{\mathcal{A}}_{m_i}$
    for $j = 1, 2, \ldots, m_i$ do
        if $|\mathcal{A}_j| > 0$ then
            $\mathcal{M}_t \leftarrow$ read_from_file($\mathcal{M}_j$)
            $\hat{\mathcal{A}}_j \leftarrow$ crop($\hat{\mathcal{A}}_j$)
            $\mathcal{A}_j \leftarrow$ rotate($\hat{\mathcal{A}}_j, 0, 360$)
            save_as_triplet($\mathcal{I}^{(s_i)}, \mathcal{A}_j, \mathcal{M}_t$)
        end if
    end for
end for
```

Figure 3.3: We use the original amodal ground-truth segmask to create the target mask. In accordance with our assumptions (Section 3.2), we omit examples with no visible target pixels, because the modal segmask set must be nonempty. A scene image is saved as a part of several triplets; one for every object in the scene, with that object acting as the target.

One-Shot Datasets

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

To evaluate model performance on the one-shot task without sim-to-real transfer, 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. The generation of this dataset exactly follows the algorithm for the training set using the new scenes and objects.

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 similarly 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.

Algorithm 2 Real One-Shot Dataset Generation Procedure

```
for $i = 1, 2, \ldots, n$ do
    $\mathcal{I}^{(s_i)} \leftarrow$ read_from_file($s_i$)
    for $j = 1, 2, \ldots, m_i$ do
        $\mathcal{M}_t \leftarrow$ read_from_file($\mathcal{M}_j$)
        for $k = 1, 2, \ldots, k_j$ do
            $\hat{\mathcal{A}}_k \leftarrow$ read_from_file($A_k$)    ▷ read separate amodal image from file
            $\hat{\mathcal{A}}_k \leftarrow$ binarize($\hat{\mathcal{A}}_k$)▷ map non-zero pixel channels to 1 and zero to 0 to create shape mask
            $\hat{\mathcal{A}}_k \leftarrow$ crop($\hat{\mathcal{A}}_k$)
            $\mathcal{A}_k \leftarrow$ rotate($\hat{\mathcal{A}}_k, 0, 360$)
            save_as_triplet($\mathcal{I}^{(s_i)}, \mathcal{A}_k, \mathcal{M}_t$)
        end for
    end for
end for
```

Figure 3.4: As opposed to the train set generation, we iterate through the amodal masks found in separate files. In accordance with our assumptions (Section 3.2), we omit examples with no visible target pixels, because the modal segmask set must be nonempty. For each scene, and for each object in a scene, we utilize every amodal image of the object as the target shape mask by binarizing it. In the case of the WISDOM-Real dataset, we have 5 color amodal images per object. Even though our data mediums are mismatched, we are able to evaluate OSSIS on this dataset in the same way we evaluate OSSIS on the validation dataset.

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,

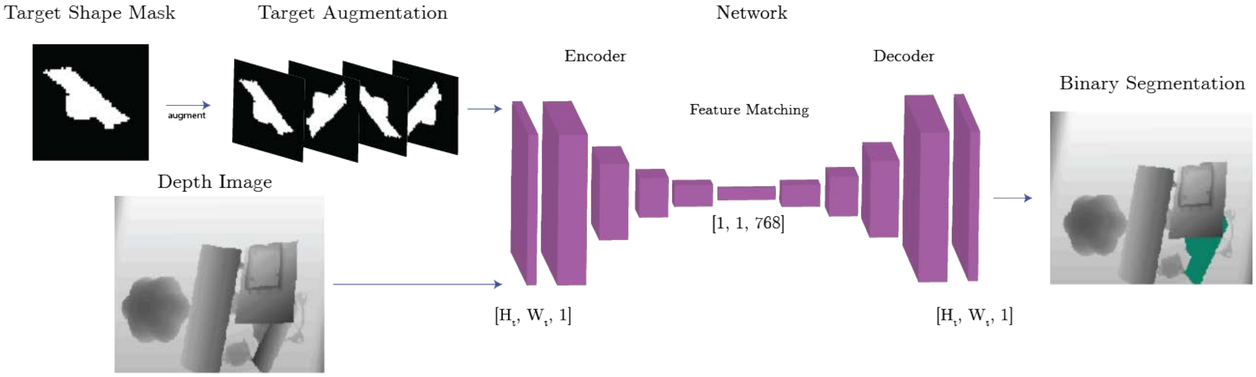

Figure 3.5: **Algorithm Overview:** The network takes in a binary shape mask of the target object and a depth image, and produces the modal segmentation mask of the target object in the depth image. We augment the target shape mask before training by rotating the mask randomly between 0 and 360 degrees, and treat each rotated mask as an individual training point.

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.

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.* [32], 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 the validation set over a range of 0.1

to 0.5 with a step size of 0.05.

For the loss function, we use a weighted cross entropy loss based on the average number of positive pixels in a sample of modal segmask labels. Specifically, we draw a sample of 100 triplets *i.i.d.* from the training dataset and sum their modal segmasks, dividing by 100 times the image dimensions. We use the reciprocal of this to weight the positive class term in the cross entropy expansion. This improves results at convergence and removes a significant portion of training time near the beginning where the network is stuck at a local optimum of predicting no positive pixels.

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).

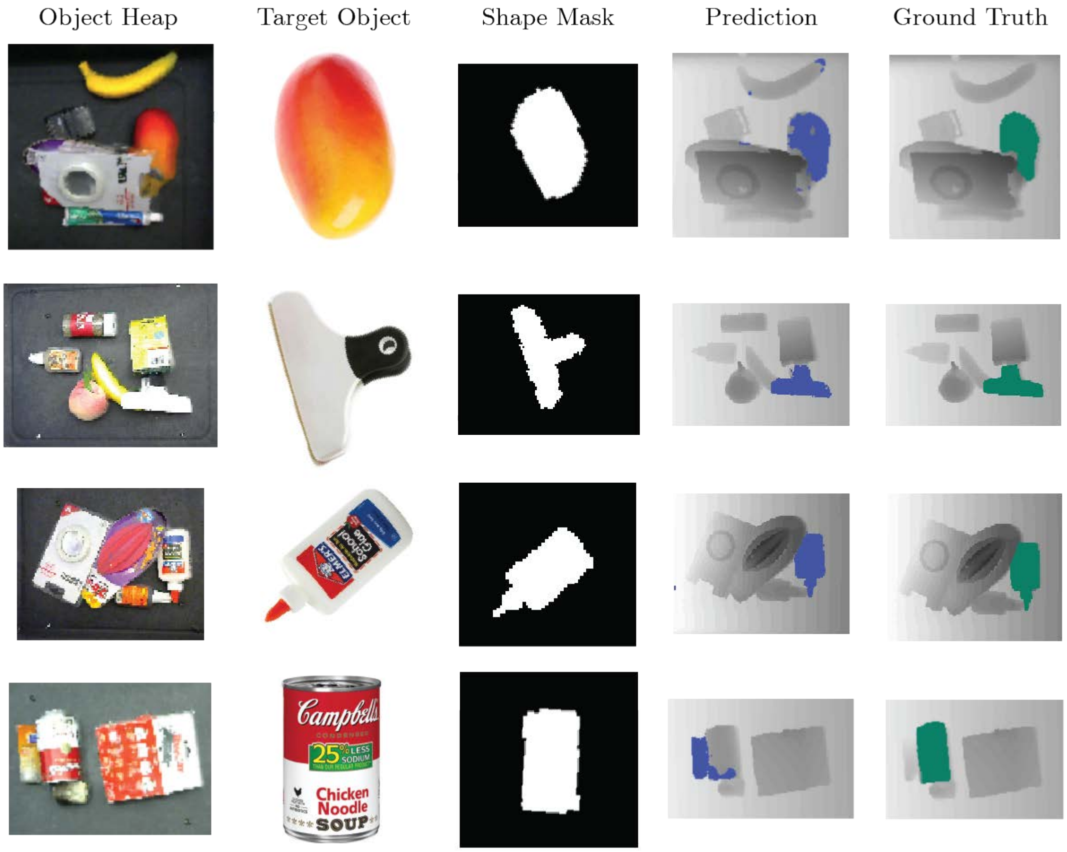

Figure 3.6: Qualitative results from applying the algorithm on the test real depth image dataset. We include the color image as well for visual clarity. The first three rows show the ability of the network to segment partially occluded and rotated target objects at different scales. The final row displays a failure mode inherent to the shape-based approach of confounding two similar shapes.

3.3 Experiments

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 3.1 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.

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.

The filter matching algorithm localizes the target object within the image by measuring cosine similarity scoring for each convolution [3, 46, 49]. 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 [16], 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. 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 $k_t = 5$ 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. For OSSIS, we report mIoU both when taking the mean and maximum IoU across the 5 images.

Method	One-Shot Sim	One-Shot Real
Filter Matching	0.186	0.171
Two-Stage CNN	N/A	0.316
OSSIS (R=1, Mean)	0.357	0.250
OSSIS (R=1, Max)	0.357	0.250
OSSIS (R=4, Mean)	0.591	0.299
OSSIS (R=4, Max)	***0.591***	***0.381***

Table 3.1: 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.

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 3.1. 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. This is in large part due to the single stage nature of OSSIS.

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 3.2, 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

Method	**Runtime**	**Training Time**
Filter Matching	180 ms	N/A
Two-Stage CNN	2.5 s	24 hrs
OSSIS (R=1)	45 ms	18 hrs
OSSIS (R=4)	45 ms	38 hrs

Table 3.2: We compare the forward pass runtime and training times of OSSIS trained on datasets with $R = 1$ and $R = 4$ random rotations with the two baselines. OSSIS runs 4 times faster compared to filter matching and over 50x faster than the two-stage CNN. A major reason for the two-stage CNN taking significantly longer is that segmenting the entire image and cross-comparing each resultant mask against the given object can be expensive with many objects in the scene. OSSIS, by comparison, directly produces the target mask in a single-stage. While OSSIS takes less time to train in on the base training dataset, it takes significantly longer on the optimal rotated dataset.

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 both the sim and real datasets.

A visual study of segmentation successes and failures is shown in Figure 3.6. 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.

Ablations

We characterize both the effect of augmenting the dataset with rotations and the effect of dataset size on network performance. Table 3.7 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.

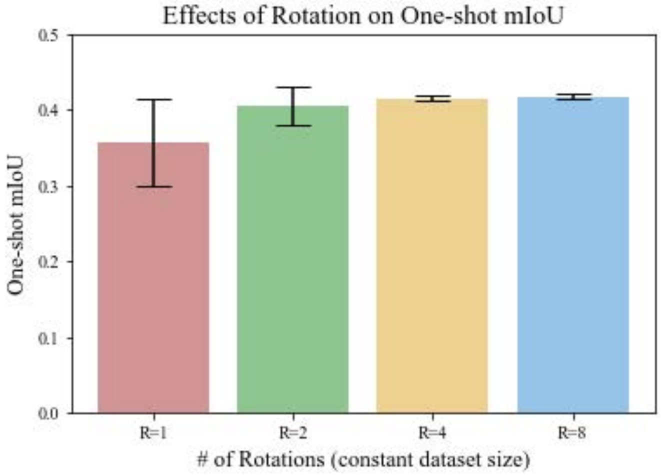

Figure 3.7: We measure the effects of rotating target masks on network performance. We control for dataset size and number of unique scene images, and train a model on each resulting dataset. Using four random rotations per target mask ($R = 4$) is the most effective in increasing mIoU and reducing variance while also having a low dataset generation cost. We expect that using more than four rotations potentially yields diminishing returns because with target shape masks there is no texture or color to break the symmetry when an object with a high degree of symmetry in its shape is used as the target.

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 3.7 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. While we find L2 regularization 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.

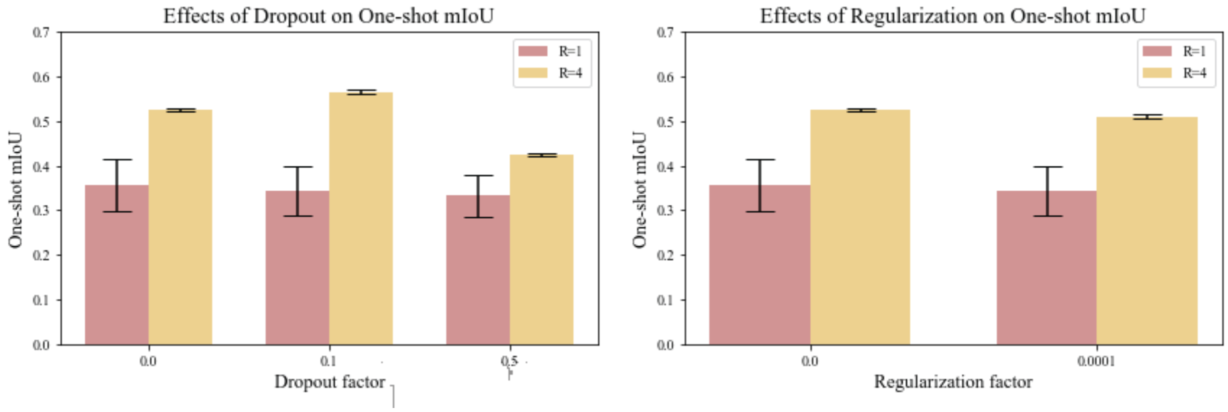

Figure 3.8: We measure the effects of fine-tuning dropout and regularization factor on network performance on network performance. Applying L2 penalty regularization does not improve the output of the network. We find that slight dropout applied to the last layer is effective in increasing the one-shot sim mIoU for $R = 4$, but has no effect when $R = 1$.

3.4 Individual Contributions

This work on One-Shot Amodal-to-Modal Instance Segmentation was a joint effort between Andrew Li, Michael Danielczuk, and Professor Ken Goldberg. Our paper was submitted to IEEE CASE 2020 and is under review. I was responsible for the design, implementation, and evaluation of OSSIS and the classical baseline. Throughout the research process I ran over 100 experiments optimizing network architecture and testing different data sources. The code for this project can be found at `https://github.com/andrewyli/one-shot-segmentation`. Michael Danielczuk was responsible for implementing the one-shot setting of the CNN baseline, helped with the design of OSSIS, and aided in the writing of the paper. Professor Goldberg contributed valuable feedback and guidance throughout the project. This project was completed during the COVID-19 pandemic.

Chapter 4

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 multiple datasets. Experiments suggest that OSSIS outperforms a filter-matching baseline method by 21% in mIoU.

We observe that our experiments are centered around the WISDOM-Sim and WISDOM-Real datasets, and in future work can generalize the shape-based method to other datasets. Although we believe the shape-based method fits the industrial application setting, there are potentially uses for it in more general segmentation applications as well. Furthermore, when applying our methods towards different segmentation problems, we would like to perform more experiments comparing established state-of-the-art results with the new shape-based results to measure any dropoff in performance.

In future work, we will also continue to address the disparity between one-shot sim and real images. We expect that incorporating all available amodal target masks as a batch will be helpful in recognizing objects out of the training distribution. Additionally, we aim to move in this direction because it more accurately reflects the challenges of a distribution center or warehouse sorting through objects.

With access to the physical objects of WISDOM-Real, or lookalikes, inserting OSSIS into a mechanical search pipeline would be informative and provide further ways to optimize the method for practical use. In such a pipeline, having multiple cameras positioned at various angles might showcase an instance where single-stage methods are desirable.

Bibliography

[1] Jeroen van Baar et al. “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] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495. ISSN: 1939-3539. DOI: 10.1109/tpami.2016.2644615. URL: http://dx.doi.org/10.1109/TPAMI.2016.2644615.

[3] Kai Briechle and Uwe D Hanebeck. “Template matching using fast normalized cross correlation”. In: *Optical Pattern Recognition XII*. Vol. 4387. International Society for Optics and Photonics. 2001, pp. 95–102.

[4] Adrian Broadhurst, Tom W Drummond, and Roberto Cipolla. “A probabilistic framework for space carving”. In: *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*. Vol. 1. IEEE. 2001, pp. 388–393.

[5] Jane Bromley et al. “Signature verification using a siamese time delay neural network.” In: *Int. Journal of Pattern Recognition and Artificial Intelligence* (1993).

[6] Liang-Chieh Chen et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *Lecture Notes in Computer Science* (2018), pp. 833–851. ISSN: 1611-3349. DOI: 10.1007/978-3-030-01234-2_49. URL: http://dx.doi.org/10.1007/978-3-030-01234-2_49.

[7] Michael Danielczuk et al. “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] Michael Danielczuk et al. “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] JLaESaT Darrell, J Long, and E Shelhamer. “Fully Convolutional Networks for Semantic Segmentation”. In: *IEEE T PATTERN ANAL* 39.4 (2014).

[10] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition* (June 2009). DOI: `10.1109/cvpr.2009.5206848`. URL: `http://dx.doi.org/10.1109/CVPR.2009.5206848`.

[11] Andreas Eitel, Nico Hauff, and Wolfram 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.

[12] Zhibo Fan et al. "FGN: Fully Guided Network for Few-Shot Instance Segmentation". In: *arXiv preprint arXiv:2003.13954* (2020).

[13] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. "Efficient Graph-Based Image Segmentation". In: *International Journal of Computer Vision* 59.2 (2004), pp. 167–181. ISSN: 0920-5691. DOI: `10.1023/b:visi.0000022288.19776.77`. URL: `http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77`.

[14] Sanja Fidler et al. "Bottom-Up Segmentation for Top-Down Detection". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2013). DOI: `10.1109/cvpr.2013.423`. URL: `http://dx.doi.org/10.1109/CVPR.2013.423`.

[15] JunYoung Gwak et al. "Weakly supervised 3d reconstruction with adversarial constraint". In: *Int. Conf. on 3D Vision (3DV)*. IEEE. 2017, pp. 263–272.

[16] Kaiming He et al. "Mask R-CNN". In: *Proc. IEEE Int. Conf. on Computer Vision (ICCV)* (2017). DOI: `10.1109/iccv.2017.322`. URL: `http://dx.doi.org/10.1109/ICCV.2017.322`.

[17] Zhang-Wei Hong et al. "Virtual-to-real: Learning to control in visual semantic segmentation". In: *arXiv preprint arXiv:1802.00285* (2018).

[18] Ting-I Hsieh et al. "One-Shot Object Detection with Co-Attention and Co-Excitation". In: *Proc. Advances in Neural Information Processing Systems*. 2019, pp. 2721–2730.

[19] Vladimir Iglovikov et al. "TernausNetV2: Fully Convolutional Network for Instance Segmentation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2018). DOI: `10.1109/cvprw.2018.00042`. URL: `http://dx.doi.org/10.1109/CVPRW.2018.00042`.

[20] Edward Johns, Stefan Leutenegger, and Andrew 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.

[21] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[22] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition". In: *ICML deep learning workshop*. Vol. 2. Lille. 2015.

[23] Victor Lempitsky et al. "Image Segmentation with A Bounding Box Prior". In: *Proc. IEEE Int. Conf. on Computer Vision (ICCV)* (2009). DOI: `10.1109/ICCV.2009.5459262`.

[24] Ian Lenz, Honglak Lee, and Ashutosh Saxena. "Deep learning for detecting robotic grasps". In: *Int. Journal of Robotics Research (IJRR)* 34.4-5 (2015), pp. 705–724. ISSN: 1741-3176. DOI: 10.1177/0278364914549607. URL: http://dx.doi.org/10.1177/0278364914549607.

[25] Anat Levin and Yair Weiss. "Learning to Combine Bottom-Up and Top-Down Segmentation". In: *Lecture Notes in Computer Science* (2006), pp. 581–594. ISSN: 1611-3349. DOI: 10.1007/11744085_45. URL: http://dx.doi.org/10.1007/11744085_45.

[26] Sergey Levine et al. "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection". In: *Int. Journal of Robotics Research (IJRR)* 37.4-5 (2017), pp. 421–436. ISSN: 1741-3176. DOI: 10.1177/0278364917710318. URL: http://dx.doi.org/10.1177/0278364917710318.

[27] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *Lecture Notes in Computer Science* (2014), pp. 740–755. ISSN: 1611-3349. DOI: 10.1007/978-3-319-10602-1_48. URL: http://dx.doi.org/10.1007/978-3-319-10602-1_48.

[28] Jeffrey Mahler et al. "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] Jeffrey Mahler et al. "Learning ambidextrous robot grasping policies". In: *Science Robotics* 4.26 (2019), eaau4984.

[30] D. Martin et al. "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics". In: *Proc. 8th Int'l Conf. Computer Vision*. Vol. 2. July 2001, pp. 416–423.

[31] Wojciech Matusik et al. "Image-based visual hulls". In: *Conf. on Computer graphics and interactive techniques*. 2000, pp. 369–374.

[32] Claudio Michaelis, Matthias Bethge, and Alexander S Ecker. "One-shot segmentation in clutter". In: *arXiv preprint arXiv:1803.09597* (2018).

[33] Claudio Michaelis et al. "One-shot instance segmentation". In: *arXiv preprint arXiv:1811.11507* (2018).

[34] S. A. Nene, S. K. Nayar, and H. Murase. "Columbia Object Image Library (COIL-20)". In: *Technical Report CUCS-005-96*. Feb. 1996.

[35] Richard Nock and Frank Nielsen. "Statistical region merging". In: *IEEE Trans. Pattern Analysis and Machine Intelligence* 26.11 (2004), pp. 1452–1458.

[36] Deepak Pathak et al. "Learning instance segmentation by interaction". In: *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 2042–2045.

[37] Martin Rajchl et al. "DeepCut: Object Segmentation From Bounding Box Annotations Using Convolutional Neural Networks". In: *IEEE Transactions on Medical Imaging* 36.2 (2017), pp. 674–683. ISSN: 1558-254X. DOI: 10.1109/tmi.2016.2621185. URL: http://dx.doi.org/10.1109/TMI.2016.2621185.

[38] Hasnain Raza et al. “Weakly Supervised One Shot Segmentation”. In: *IEEE Int. Conf. on Computer Vision Workshops*. 2019.

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[40] Olga Russakovsky et al. “Imagenet large scale visual recognition challenge, 2014”. In: *arXiv preprint arXiv:1409.0575* (2014).

[41] Olga Russakovsky et al. “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision* 115.3 (2015), pp. 211–252. ISSN: 1573-1405. DOI: `10.1007/s11263-015-0816-y`. URL: `http://dx.doi.org/10.1007/s11263-015-0816-y`.

[42] Andrei A. Rusu et al. “Sim-to-Real Robot Learning from Pixels with Progressive Nets”. In: *Conf. on Robot Learning (CoRL)*. 2017.

[43] Daniel Seita et al. “Deep transfer learning of pick points on fabric for robot bed-making”. In: *arXiv preprint arXiv:1809.09810* (2018).

[44] Amirreza Shaban et al. “One-Shot Learning for Semantic Segmentation”. In: *Proc. British Machine Vision Conference (BMVC)* (2017). DOI: `10.5244/c.31.167`. URL: `http://dx.doi.org/10.5244/c.31.167`.

[45] Ashish Shrivastava et al. “Learning from Simulated and Unsupervised Images through Adversarial Training”. In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2017). DOI: `10.1109/cvpr.2017.241`. URL: `http://dx.doi.org/10.1109/CVPR.2017.241`.

[46] Arnold WM Smeulders et al. “Visual tracking: An experimental survey”. In: *IEEE transactions on pattern analysis and machine intelligence* 36.7 (2013), pp. 1442–1468.

[47] Josh Tobin et al. “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.

[48] Jonathan Tremblay et al. “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.

[49] Du-Ming Tsai and Chien-Ta Lin. “Fast normalized cross correlation for defect detection”. In: *Pattern Recognition Letters* 24.15 (2003), pp. 2625–2631.

[50] Shubham Tulsiani et al. “Multi-view Supervision for Single-view Reconstruction via Differentiable Ray Consistency”. In: *IEEE Trans. Pattern Analysis and Machine Intelligence* (2019).

[51] Oriol Vinyals et al. “Matching networks for one shot learning”. In: *Proc. Advances in Neural Information Processing Systems*. 2016, pp. 3630–3638.

[52] Guotai Wang et al. "Interactive Medical Image Segmentation Using Deep Learning With Image-Specific Fine Tuning". In: *IEEE Transactions on Medical Imaging* 37.7 (2018), pp. 1562–1573. ISSN: 1558-254X. DOI: `10.1109/tmi.2018.2791721`. URL: `http://dx.doi.org/10.1109/TMI.2018.2791721`.

[53] Xinchen Yan et al. "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision". In: *Proc. Advances in Neural Information Processing Systems*. 2016, pp. 1696–1704.

[54] Nida M Zaitoun and Musbah J Aqel. "Survey on image segmentation techniques". In: *Procedia Computer Science* 65 (2015), pp. 797–806.