MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Few-shot Transparent Instance Segmentation for Bin Picking

Cherian, Anoop; Jain, Siddarth; Marks, Tim K.

TR2024-127 September 13, 2024

Abstract

In this paper, we consider the problem of segmenting multiple instances of a transparent object from RGB or gray scale camera images in a robotic bin picking setting. Prior methods for solving this task are usually built on the Mask-RCNN framework, but they require large annotated datasets for fine-tuning. Instead, we consider the task in a few- shot setting and present TrInSeg, a data-efficient and robust instance segmentation method for transparent objects based on Mask-RCNN. Our key innovations in TrInSeg are twofold: i) a novel method, dubbed TransMixup, for producing new training images using synthetic transparent object instances created by spatially transforming annotated examples; and ii) a method for scoring the consistency between the predicted segments and rotations of an ideal object template. In our new scoring method, the spatial transformations are produced by an auxiliary neural network, and the scores are then used to filter inconsistent instance predictions. To demonstrate the effectiveness of our method, we present experiments on a new few-shot dataset consisting of seven categories of non-opaque (transparent and translucent) objects, each category varying in the size, shape, and degree of transparency of the objects. Our results show that TrInSeg achieves state-of-the-art performance, improving fine-tuned Mask-RCNN by more than 14% in mIoU, while requiring very few annotated training samples.

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2024

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Few-shot Transparent Instance Segmentation for Bin Picking

Anoop Cherian, Siddarth Jain, and Tim K. Marks

***Abstract*— In this paper, we consider the problem of segmenting multiple instances of a transparent object from RGB or gray scale camera images in a robotic bin picking setting. Prior methods for solving this task are usually built on the Mask-RCNN framework, but they require large annotated datasets for fine-tuning. Instead, we consider the task in a few-shot setting and present *TrInSeg*, a data-efficient and robust instance segmentation method for transparent objects based on Mask-RCNN. Our key innovations in TrInSeg are twofold: i) a novel method, dubbed *TransMixup*, for producing new training images using synthetic transparent object instances created by spatially transforming annotated examples; and ii) a method for scoring the consistency between the predicted segments and rotations of an ideal object template. In our new scoring method, the spatial transformations are produced by an auxiliary neural network, and the scores are then used to filter inconsistent instance predictions. To demonstrate the effectiveness of our method, we present experiments on a new few-shot dataset consisting of seven categories of non-opaque (transparent and translucent) objects, each category varying in the size, shape, and degree of transparency of the objects. Our results show that TrInSeg achieves state-of-the-art performance, improving fine-tuned Mask-RCNN by more than 14% in mIoU, while requiring very few annotated training samples.**

I. INTRODUCTION

From small medicine bottles to the giant window panes of modern buildings, transparent objects are ubiquitous in our daily lives. Commonplace transparent items such as glasses, jars, and bottles—ubiquitous in both home and industrial settings—pose both challenges and opportunities for robotic manipulation. When deploying robotic agents to automate our tasks, it is thus essential to ensure that these agents can perceive and operate on transparent and semi-transparent objects [1]. One such task is robotic bin picking, in which a robot is required to pick up instances of an object from a cluttered bin consisting of many object instances. This task occurs in both factory settings (e.g., for kitting, assembly, and packing) and home/business settings (e.g., picking a glass bottle from a box of bottles to serve juice, or picking wineglasses from a dish washer). The first step in solving the bin picking problem is to segment the object instances from each other and the background to produce a set of instance candidates, which can be used in a grasp and motion planning pipeline for effectuating the pick.

While many approaches to instance segmentation have been proposed [2], [3], [4], [5], they typically assume *in the wild* settings and very general contexts, and they operate mainly on opaque objects. Although there are extensions of these methods for transparent object segmentation [6], [7], [8], [9], [10], the problem of transparent *instance* segmentation (segmenting individual instances of the same object) has not received much attention in settings with limited training data. Typically, the characteristics of transparent objects present significant challenges for robots in perception. For example, these objects often lack discernible surface features such as color and texture, relying heavily on the background of the image for visual distinction. Moreover, the reflective and refractive nature of transparent surfaces complicate the acquisition of precise depth data using depth sensors. Consequently, the collected data may prove invalid or contain unpredictable noise, thereby exacerbating the challenges associated with transparent object perception. Our setting is further complicated by shadows and the existence of non-transparent parts, as well as the overlap of transparent objects on top of each other, making the problem of instance segmentation of transparent objects particularly challenging.

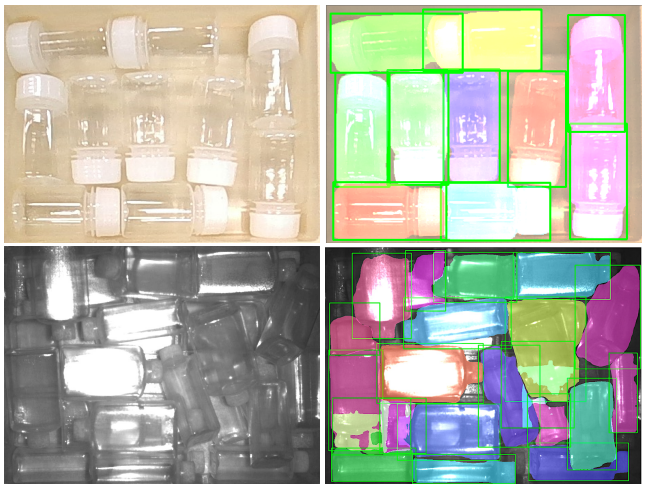


Fig. 1. *Left:* Two example images from our few-shot dataset, each containing multiple instances of a different transparent bottle captured using a RealSense (*top) or Ensenso* (bottom) camera. *Right:* Instance segmentation results on each image using our TrInSeg method.

In this paper, we consider this problem in a bin picking setting using images from a camera overlooking the bin. Figure 1 shows example images of instances in our setup. A typical way to approach this problem is to leverage popular pre-trained deep learning-based instance segmentation frameworks such as Mask-RCNN [2] and fine-tune them on annotated data in a supervised manner. Some extensions of Mask-RCNN for detecting transparent objects leverage transparency cues. For example, in Kalra et al. [11] polarized images are used with Mask-RCNN for transparent instance segmentation, light-field images are considered in [12], en-

The authors are with Mitsubishi Electric Research Labs, Cambridge, MA, USA 02139. Email ids: {cherian, sjain, tmarks}@merl.com

hanced matting is proposed in [13], stereo images are used in [14], and RGB-D images in [15], [8], [16]. However, unlike these approaches, we consider the task in the native RGB or grayscale image setting without the need for any other modality, thereby allowing our method to be generally applicable. A common challenge in such a setting is the demand for large annotated training sets for fine-tuning the pre-trained model [8], [17], [18], [19]; however, producing such training sets could be laborious, expensive, and often inflexible when target objects vary frequently. While simulations could be considered to bypass some of these issues [20], bridging the sim-to-real gap may pose further challenges.

In this paper, we consider the problem of transparent object instance segmentation using a Mask-RCNN backbone (similar to prior methods [11], [17]). Unlike previous work, however we approach the problem from the perspective of few-shot learning in a robotic bin picking setting, which to the best of our knowledge is a direction that has not been explored before. Specifically, in comparison to prior methods that need hundreds or thousands of annotated training images, our method requires far fewer (tens of annotated images). A key insight of our method is based on the inherent symmetry and rigidity of the transparent objects that we consider (as illustrated in Figures 1 and 5); specifically, each *instance* is a rigid spatial transformation of an underlying *object model* that is rendered into the camera image and adapted using physical characteristics of translucency. We do not assume access to 3D CAD models of the objects; instead, we use the annotated instances to form an approximate template of the object. Based on this observation, we propose *TrInSeg*, our few-shot **Tr**ansparent **In**stance **Seg**mentation method, which leverages the training examples in two ways: i) We generate a potentially infinite synthetic training set (for training any deep learning instance segmentation backbone) using the approximate object model obtained from the instance annotations—a method we call *TransMixup*; and ii) we filter the instances predicted by the backbone by scoring their consistency with the object model. Our scoring method is based on an auxiliary spatial transformer neural network that predicts the rotation parameters of the object model that are consistent with the instance predictions by the backbone. This consistency has the additional advantage of inferring instance occlusions.

To empirically evaluate our method, we provide extensive experiments on a new real-world instance segmentation dataset, consisting of RGB and gray-scale images of seven different categories of objects in a bin picking setting. The objects vary in their transparencies, shapes, sizes, and their number in the bin. Our experiments clearly show that our method greatly improves the instance segmentation accuracy of Mask-RCNN, using mean intersection-over-union (mIoU). Compared to fine-tuning the backbone using the original annotated examples, TransMixup improves the mIoU by nearly 8%, and our instance filtering method improves mIoU by an additional 6%. Our method requires fewer than 30 annotated training samples; even if we use fewer than five annotated examples, it can still achieve over 80% mIoU.

II. RELATED WORKS

Transparent Object Segmentation: Initial research on this topic used hand-crafted characteristics, such as boundary features [21]; however, recent years have seen a shift towards deep learning methods [22], [9], [23], [10]. For example, Mask-RCNN is extended to detect individual transparent objects in [7], a Transformer encoder-decoder is introduced in [24], and a three-stream encoder-decoder model that fuses RGB, infrared, and RGB-IR images is presented in [25]. A few studies address transparent object grasping by leveraging both visual and tactile data [26]. While these methods consider transparent object segmentation, or isolated instance segments, we deal with the problem of segmentation with several instances overlapping, where the model needs to identify each instance of the same underlying object class.

Mixup Methods: Our approach has similarities to the popular cut-mix [27] and mixup [28] methods. There are also adaptations of cut-mix to transparent data [6], [29]. The main idea of cut-mix is to cut annotated image patches and paste them at random locations in other images—so as to augment training data—whereas mixup methods mix data features extracted from an intermediate layer of a deep learning model. While both methods are related to our proposed scheme, there are two important differences. First, our setting is for transparent instance segmentation, as opposed to the object detection setting of these prior works. Second, cut-mix methods do not consider any semantic context when mixing up the data, and at a conceptual level act as training regularizers—in contrast, TrInSeg offers a way to produce large in-distribution training data from few-shot examples.

Depth Completion: Another approach is to improve the quality of transparent depth surfaces through depth completion from RGB images. In Zhang et al. [16], a two-stage depth completion pipeline is proposed that uses surface normals and occlusion boundaries. However, this requires solving a global optimization objective, which is improved by Tang et al. [30] using a self-attentive adversarial network. Zhu et al. [31] present a local implicit function for depth refinement, while Xu et al. [32] combine completion techniques with point clouds. More recently, multi-view depth completion methods using NeRF [33] and physics-based networks [34] are proposed to reconstruct the depth. Real-world datasets for robotic applications, such as that of ClearGrasp [8], are constrained by a limited sample size due to the time- and labor-intensive process for generating missing data. Synthetic datasets are commonly used for tasks demanding precise ground-truth sensor data, as exemplified by the Omniverse object dataset [31].

To improve generalization and applicability in real-world robotic applications, there is a need for a few-shot model that is tailored to transparent instance segmentation. Few-shot segmentation holds particular importance for transparent objects due to their unique optical properties and the scarcity of suitable labeled datasets, which are often difficult to obtain in sufficient quantity and quality. Our model needs only a small amount of real-world data for effective deployment.

III. METHOD

Suppose X denotes an RGB or grayscale image of height H and width W, containing multiple instances of an object $\mathscr{O}$. Let $\mathscr{Y} = \{Y_1, Y_2, \ldots, Y_{\#(X)}\}$ be the set of instance masks for all the instances in X – one for each instance, where $\#(X)$ indicates the total number of instances in X. We assume the masks Y are of the same spatial dimensions as the image X, but containing zeros everywhere except at pixel locations overlapping with the corresponding instance in the image, where the mask takes a constant numeric value identifying the instance. We assume this instance identifier is unique across the masks in $\mathscr{Y}$. Further, let $\mathscr{D} = \{(X_i, \mathscr{Y}_i)\}_{i=1}^{n}$ denote the few-shot training set consisting of n such pairs of an image and all of its instance annotation masks. We assume all the instances in a given image are annotated and the total number of annotated instances in $\mathscr{D}$ is small. For example, in the Medical Bottle object class used in our experiments, the number of training images is about 10, and each image has 1–5 annotated instances. Our objective is to train a deep learning model $\mathbf{M}_\theta$ with parameters θ that can take as input an image X and predict an instance segmentation mask similar to those in the ground truth $\mathscr{Y}$. Note that in the prediction refinement that we detail subsequently, we do not assume a model will predict all of the ground truth instances; however, the produced instances must have a quality score higher than a specified threshold.

In this paper, we use a pre-trained Mask-RCNN backbone for $\mathbf{M}_\theta$. We assume that during its pre-training, this backbone has not seen the transparent objects that we will use. Thus, a zero-shot transfer of the backbone model is prone to errors, especially when dealing with transparent objects. A naïve approach to adapt the backbone to our data setting is then to fine-tune the parameters θ on the images in $\mathscr{D}$, but given only a few annotated images, the training can be ineffective. However, if there is a way we can produce a larger training set from the few-shot examples, where this additional training data spans the space of the object appearances more densely, that could lead to a better training of the Mask-RCNN model. Inspired by this insight, we propose TrInSeg, in which our key innovations are: (i) Using a method that we call TransMixup, we leverage the few-shot instance masks to approximately characterize an object model, which we then use to synthetically produce an unlimited supply of training data; and (ii) we use the object model as a way to score the quality of the instance predictions produced by our fine-tuned model, encouraging the selection of instances that are unoccluded (i.e., not overlapped by other instances), have complete masks so that their boundaries are clear, and are potentially separable so that a grasping approach can be applied. In this section, we explain these ideas in detail.

A. TransMixup: Transparent Instance Mixup

Our key idea to augment the few-shot training set is to use the annotated examples to produce a shape and appearance model of the object using randomly sampled annotated masks and their respective image instance patches, which are then spatially transformed and blended with the image to produce

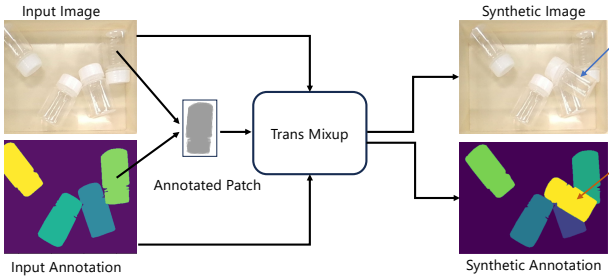


Fig. 2. Illustration of our TransMixup approach for synthetic data generation from few-shot annotated data samples (marked by arrows).

diverse training images containing an arbitrary number of instances in diverse spatial and overlapping instance configurations. In Figure 2, we illustrate the key steps of this method. Formally, let $Y \sim \mathscr{Y}$ be a random mask from the set $\mathscr{Y}$ for an image X, and let $X_Y = \text{crop}_Y(X[Y])$ denote the image patch produced after the operations of applying a pixel-wise Hadamard product between X and the mask Y (i.e., $X[Y] = X \odot Y$) followed by an image crop using the bounding box of the instance in the mask Y. Similarly, let Y_Y denote the corresponding instance crop of the mask in Y. We only select a mask Y from $\mathscr{Y}$ if it is isolated (not overlapping with other masks), and thus X_Y captures *an* appearance of the underlying object $\mathscr{O}$. In order to produce augmentations, let $\mathscr{T}$ be a set of affine spatial transformations (include spatial rotations, shrinking/skewing, and others) operating on patches. To produce an augmented patch $\tilde{X}$ and its corresponding mask $\tilde{Y}$, we select a random transformation $\mathbf{t} \sim \mathscr{T}$ to produce $(\tilde{X}_Y, \tilde{Y}_Y) \leftarrow (\mathbf{t}(X_Y), \mathbf{t}(Y_Y))$, followed by pasting the image and mask patches at a random spatial location z on a canvas of zeros the size of the image, producing an augmented mask $\tilde{Y} = \text{paste}_z(\tilde{Y}_Y)$ and the respective masked image $\tilde{X} = \text{paste}_z(\tilde{X}_Y)$. To be clear, $\tilde{X}$ and $\tilde{Y}$ are an image and mask pair with the same spatial resolution as the input image, but containing only a single augmented instance. Next, we compose this transformed patch with the original image to create a new training image. In order to account for the transparency of the instances, we use alpha compositing, denoted $\text{blend}(X, \tilde{X}, \tilde{Y} | \alpha)$ with a blending parameter $0 \leq \alpha \leq 1$, updating the input image as:

$$X[\tilde{Y}] \leftarrow (1-\alpha)X[\tilde{Y}] + \alpha\tilde{X}[\tilde{Y}], \tag{1}$$

where with a slight abuse of notation, we assume $X[\tilde{Y}]$ *selects* image pixels at locations where $\tilde{Y}$ is non-zero.

As our new instances are always introduced above the previous instances in the depth order, the mask instance identifiers for the new instances supersedes those of previous instances, and we blend the masks in the depth order when using it for training the backbone. In order to produce diverse training samples, we apply TransMixup recursively on the same image, sampling the augmentation parameters and the object masks. Our full TransMixup algorithm is summarized in Alg. 1. Examples of synthetic transparent instance data produced using TransMixup are provided in Figure 3. As is clear, our method produces synthetic training samples that

Fig. 3. Examples of synthetic data produced using our mixup approach.

look very similar to the original images, and the alpha-blending step produces complex segmentation settings, especially with regard to instance overlaps and transparencies.

Algorithm 1 TransMixup Algorithm

Require: $\mathscr{D} := \{(X_i, \mathscr{Y}_i)\}_{i=1}^{n}$
  $\mathscr{D}' \leftarrow \mathscr{D}$ ▷ $\mathscr{D}'$ is our augmented dataset
  while $|\mathscr{D}'| < N$ **do** ▷ N is the augmented size
    $(X, \mathscr{Y}) \sim \mathscr{D}, (X', \mathscr{Y}') \sim \mathscr{D}$
    $K \sim [K_{max}]$ ▷ maximum number of augmentations
    for $k = 1 \rightarrow K$ **do**
      $X_Y \leftarrow \text{crop}_Y(X[Y]), Y_Y \leftarrow \text{crop}_Y(Y[Y]),$ for $Y \sim \mathscr{Y}$
      $(\tilde{X}_Y, \tilde{Y}_Y) \leftarrow (\mathbf{t}(X_Y), \mathbf{t}(Y_Y)), \mathbf{t} \sim \mathscr{T}$
      $(\tilde{X}, \tilde{Y}) \leftarrow (\text{paste}_z(\tilde{X}_Y), \text{paste}_z(\tilde{Y}_Y)), z \sim [H] \times [W]$
      $X' \leftarrow \text{blend}(X', \tilde{X}, \tilde{Y} | \alpha)$ for $\alpha \sim [0.5, 0.75]$
      $\tilde{Y}[\tilde{Y}] \leftarrow |\mathscr{Y}'| + 1$ ▷ increment the instance id
      $\mathscr{Y}' \leftarrow \mathscr{Y}' \cup \{\tilde{Y}\}$
    end for
    $\mathscr{D}' \leftarrow \mathscr{D}' \cup \{(X', \mathscr{Y}')\}$
  end while

B. Instance Prediction Refinement Using Templates

As is well-known, Mask-RCNN is built over the prior Faster-RCNN backbone [35], which first produces object proposals in the form of instance bounding boxes; these are then scored to select boxes of high confidence, which are used to produce mask segmentations using a deep convolutional segmentation head. While the quality of the segmentations produced may be high in relation to the scoring metric used in Mask-RCNN, these scores may not strongly correlate to the quality metrics useful for a downstream task, such as robotic grasping. Consider, for example, the case of overlapping instances; while it is possible for Mask-RCNN to produce segmentation masks of high confidence for instances that are overlapped by other instances, these instances underneath might not be useful when deciding which instances to grasp in a bin picking application. Another common issue with Mask-RCNN is that it produces false positives in regions where there are no instances at all, for example, due to specular reflections off the bin.

Extending our approach described above, we propose a simple scheme to refine the predictions. Our key idea is a variant of TransMixup, where we again select instance mask patches that are representatives—called *templates*—of the underlying object model. However, instead of using these templates to augment the training data, we collect the instance proposals produced by the backbone and score each proposal for conformity with the templates; the ones that are most conformal are selected as high-quality segmentations. Those instances that are occluded, overlapped, or were falsely detected will naturally have a low conformity (are a bad match) with a template. However, a challenge when implementing this setup is the problem of how to choose the templates, given that instances can can vary in spatial orientation, illumination, translucency, and specularity.

We observe that if we restrict the conformity to the silhouette shape of the instance, as characterized by the annotated instance masks, then the above issues can be easily circumvented. Further, in real conditions, it is often seen that there exist strong self-symmetries in the objects of our interest (e.g., glass bottles, glass jars, wineglasses, etc.), and many common transparent objects are long and thin with only a few resting poses when dropped into a bin, e.g., along their major axis (although they may occasionally have other pose variations). Taking into account these factors, we need only a few (or even a single) annotation mask(s) to characterize the object template, and the conformity can be computed for a predicted instance mask with the template if we can compute the pose that aligns the template with the prediction.

To this end, in order to effectuate the refinement, TrInSeg crops image patches based on the bounding boxes of the predictions produced by Mask-RCNN, and passes each patch to a new rotation prediction network that selects a template among the model templates and predict the spatial pose (angles) of the instance in the patch in relation to the template; when this pose is used to transform the instance mask template, it should produce the instance mask that Mask-RCNN generated. The conformance of the predicted template mask and the Mask-RCNN-predicted mask lets us decide the quality of Mask-RCNN's predictions, as well as the possibility that the instance has undergone occlusions/overlaps; the latter comes directly from the fact that the template masks are assumed to come from non-occluded instances.

Formally, let us denote the set of such templates as $\mathscr{C}$, where each template $\mathbf{c} \in \mathscr{C}$ is a cropped and centered annotated instance patch. In order to produce the pose of the instance in a proposal image patch, our idea is to train a template rotation predictor neural network $\mathbf{R}_\beta : \mathbb{R}^{h \times w \times c} \rightarrow [-\pi, \pi]^k$ with trainable parameters β, where this network takes as input the image patch cropped around the proposal instance – with c color channels and resized to spatial size $h \times w$ – and produces as output the k rotation angles of the template.[1] Suppose $\hat{Y}$ is an instance mask produced by Mask-RCNN for an input image X and if $X_{\hat{Y}}$ is the corresponding

[1] In our case, we use $k = 1$, as the objects in our dataset have only rotations about an axis perpendicular to the base of the bin.

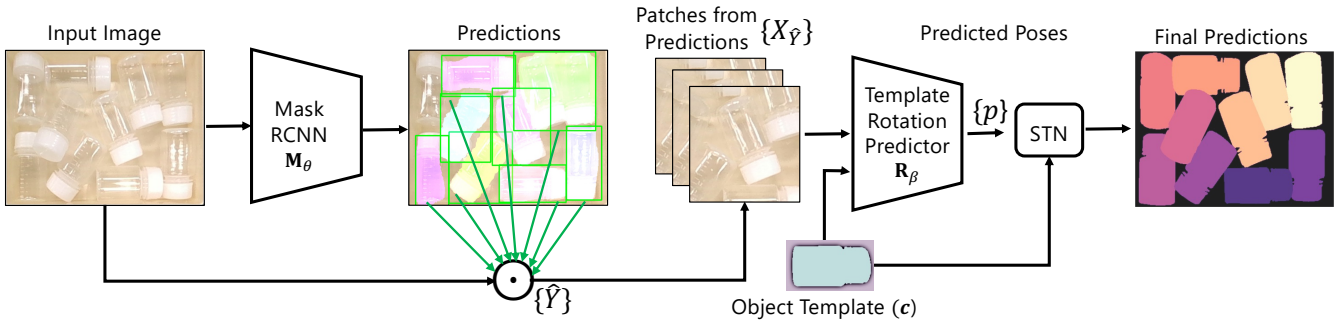


Fig. 4. Architecture of our pose prediction and mask filtering setup using a template of the transparent object.

image patch cropped around $\hat{Y}$ (using the notation described in the last section), then our training objective for $\mathbf{R}$ is given by:

$$\min_{\beta} \mathop{\mathbb{E}}_{(X,\mathscr{Y})\sim\mathscr{D}'} \mathop{\mathbb{E}}_{Y\sim\mathscr{Y}} \left\|\mathbf{t}_{\text{rot}}(\mathbf{c};p) - Y_Y\right\|_1, \text{where } p = \mathbf{R}_\beta(X_Y), \tag{2}$$

and $\mathbf{t}_{\text{rot}}(\mathbf{c};p)$ denotes the spatial rotation of the template $\mathbf{c}$ by angles in p. When selecting the data for training using (2), we assume the masks Y are taken from our augmented dataset and are selected to have no other instance on top of the selected instances in the depth order; thus we ensure that occluded instances are not used during training. Note that $\mathbf{R}$ is implemented as a spatial transformer network (STN) and the template $\mathbf{c}$ is a pixel mask of an instance. See Figure 4 for an illustration of the refinement pipeline.

Once the model $\mathbf{R}$ is trained, for a given test image patch $X_{\hat{Y}}$ from the predicted mask $\hat{Y}$, we compute the quality score with respect to $\mathbf{c}$ using intersection-over-union (IoU) as:

$$\text{score}_{\mathbf{c}}(X_{\hat{Y}}) = \text{IoU}(\mathbf{t}_{\text{rot}}\left(\mathbf{c};\mathbf{R}_\beta(X_{\hat{Y}})\right), \hat{Y}_{\hat{Y}}). \tag{3}$$

A higher score suggests better conformance between the transformed template and the predicted mask. Further, predicting the poses p using a separate network $\mathbf{R}$ makes the filtering process robust to biases in scoring (e.g., by the backbone). An instance prediction is finally selected using a combination of the template-based score and the Mask-RCNN score score_M, i.e., $(\text{score}_{\mathbf{c}} > \eta_{\mathbf{c}}) \wedge (\text{score}_M > \eta_M)$ using thresholds $\eta_{\mathbf{c}}, \eta_M > 0$.

IV. EXPERIMENTS

To empirically validate our approach, we present experiments on real world images captured using an Intel RealSense D35 and Ensenso cameras. Our dataset consists of seven categories of bottles in a bin setting taken using a downward facing camera directly into the bin. The RealSense camera images are RGB, while Ensenso images are grayscale. The object categories we use are: (i) Small Bottle, (ii) Large Bottle, (iii) Mayo Bottle, (iv) Pet Bottle, (v) Medical Bottle, (vi) Sauce Bottle, and (vii) Soy Bottle—all the categories constitute everyday objects. Example images from these categories are shown in Figure 5 first row. As is clear from the figure, each category is varied in object shape, transparency, size, and the number of instances in the bin. We collected 20 images per category and manually annotated all the instances, each image consisting of 1–10 instances for each categories except Soy Bottle, which has up to 50 instances in an image.

A. Training Details

We used a pre-trained Mask-RCNN model based on the ResNet-50 backbone that was trained on the MS-COCO dataset. We replaced its mask and the box prediction heads with randomly initialized layers. We used 10 of the annotated images for training/validation and the remaining 10 for testing; all the training images had less than 5 instances per image, while the test set images had 5-10 instances. We used less than 25 annotated instances for training in total. For TransMixup, we produced a maximum of 5 synthetic instances per image as we found that a larger number of instances produced too many overlapping instances that resulted in making the model too difficult to train. For fine-tuning the Mask-RCNN model, the entire training used new instances, and thus the augmented data size $N = \text{batch size} \times$ number of training iterations in Alg. 1. Each training iteration took about 3 seconds (on an NVIDIA 3090 GPU) with a synthetic batch size of 32, and the model was trained for about 640 iterations when the performance was seen to saturate on the validation set. The second phase in TrInSeg, prediction filtering, was trained using the augmented dataset and used a fixed object template produced from the original training images. We used a ResNet-18 pre-trained model (trained on ImageNet) as the backbone, where the last layer was replaced to predict a scalar angle for the template pose. Training this module took 2.5 seconds per iteration and was trained for about 1600 iterations.

B. Comparison Methods

To the best of our knowledge, this is the first paper describing segmentation of homogeneous transparent objects in a bin, and thus there are no prior methods to compare to on this task. To this end, we evaluate methods that are designed for general purpose segmentation problems, some of which we have adapted to our setup. Specifically, we evaluate against: i) the Segment Anything (SAM) model [3], which is a general purpose segmenter trained on millions of images

Method	Small	Large	Mayo	Medical	Pet	Sauce	Soy	Avg.
RealSense RGB Images								
SAM [3]	68.1	58.5	78.9	78.1	22.4	68.9	66.4	63.1
Lang-SAM [36]	87.2	78.7	87.3	84.4	88.3	60.8	14.7	71.6
InstaSeg [37]	43.1	48.7	47.1	41.9	46.1	47.7	31.7	43.7
Mask-RCNN (FT) [2]	84.1	80.9	76.9	79.8	79.7	67.7	71.1	77.2
iFS-RCNN (FT) [38]	87.7	80.6	84.7	80.6	87.7	64.9	78.7	80.7
TrInSeg (ours)	90.1	88.9	88.4	85.3	88.1	74.6	82.5	85.4
TrInSeg + Filtering (ours)	**92**	**95.5**	**93.1**	**92.2**	**93.7**	**84.4**	**85.4**	**90.9**
Ensenso Grayscale Images								
Mask-RCNN (FT)	84.1	81.9	44.0	80.8	85.7	62.4	65.1	72.0
iFS-RCNN (FT) [38]	91.4	80.5	76.9	90.2	83.8	63.5	73.5	79.9
TrInSeg + Filtering (ours)	**98.1**	**96.6**	**93.1**	**96.3**	**96.8**	**88.5**	**91.1**	**94.3**

TABLE I

COMPARISONS TO PRIOR INSTANCE SEGMENTATION METHODS ON MIOU ON BOTH RGB AND GRAYSCALE IMAGES ON OUR TEST SET. NOTE THAT [3], [36] ARE ZERO-SHOT, [2], [37] ARE RE-PURPOSED FOR FEW-SHOT FINE-TUNING, AND [38] IS A FEW-SHOT MODEL.

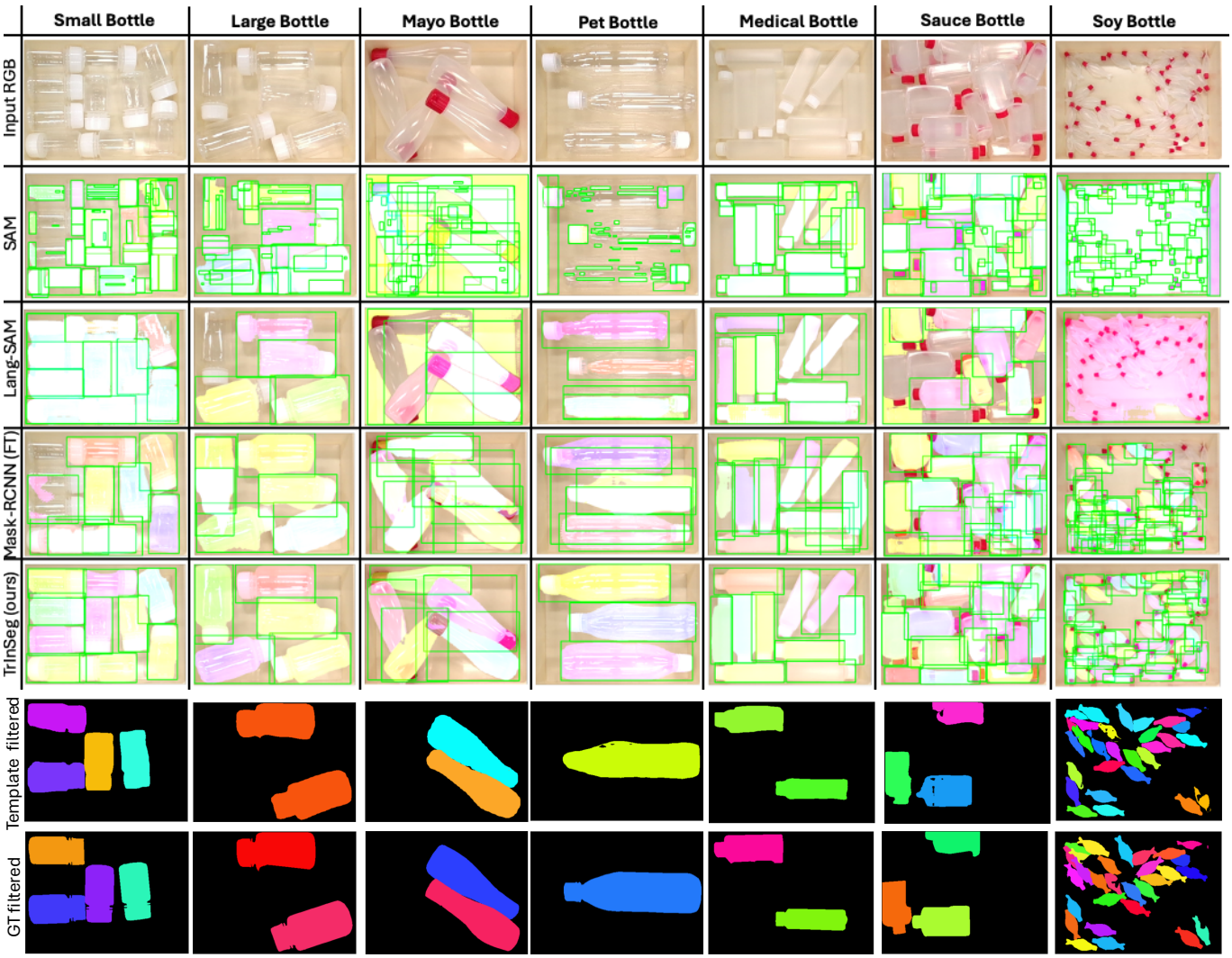
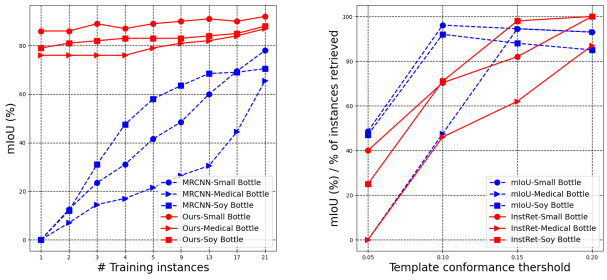


Fig. 5. First row shows RealSense RGB camera images from the seven categories of objects we use in our dataset and their instances. We also show qualitative comparisons of detection and segmentation of our method TrInSeg versus state-of-the-art approaches. For the results on Mask-RCNN, we used a model that is fine-tuned on our dataset, however without our TransMixup augmentation. The last two rows show the template filtered instances—the first row is the predicted mask by TrInSeg, and the last row is the ground-truth mask.

(and thus we assume that it must have seen instances similar to ours in its training set), ii) Lang-SAM [36], which is a zero-shot extension of SAM guided by language, iii) Mask-RCNN (FT), which is Mask-RCNN fine-tuned on our images but without the synthetic augmentation, iv) iFS-RCNN [38] which is a recent few-shot incremental instance segmentation



(a) (b)

Fig. 6. In (a), we plot mIoU against increasing number of few-shot examples. We compare fine-tuning Mask-RCNN with data generated using our TrInSeg and fine-tuning with original training data. In (b), we compare the sensitivity of the filtering threshold and also reporting the ratio (in %) of *instances retrieved* over the total number of instances.

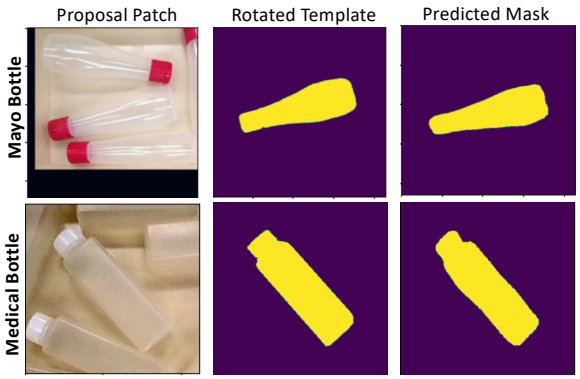


Fig. 7. Qualitative results showing the input RGB patch to the filtering model (left), the transformed template using the rotation predicted by the network (middle), and the TrInSeg predicted mask (i.e., $\hat{Y}$) (right).

method built on Mask-RCNN, and iv) InstaSeg [37], a recent segmentation model for opaque objects using depth images we repurposed to use RGB images as well.

In Table I, we analyze the quantitative performances of each of these methods on the predicted segmentations versus their ground truths measured using the standard mean-intersection-over-union (mIoU) metric using the RealSense RGB images. Qualitative examples of segmentations are provided in Figure 5. As is clear, SAM either over-segments the image or the segmentations miss parts of the instance, perhaps because it is not given cues on what needs to be segmented in the given images. To address this, in Lang-SAM, we provide the name of the object category as input, which improves its segmentation quality; however, the performance is significantly low for objects that it has perhaps not seen in its training, such as the Soy Bottle class. The Mask-RCNN (FT) and iFS-RCNN (FT) methods perform relatively better as they are trained on our dataset, however, due to the few-shot nature of the task, the performance is poor. The RGB-D method InstaSeg performs poorly as the depth images for transparent objects are noisy. Overall, we find that our proposed method demonstrates large-margin improvements in mIoU across the object categories, improving iFS-RCNN (FT) by nearly 5% on average. We also find that our prediction filtering further improves the accuracy by $\sim$6%.

In Figure 5 (last two rows), we show several qualitative examples of our pose filtering scheme that selects a subset of the Mask-RCNN predictions that are conforming to the class template used. We show the selected instances as well as their ground truth masks, clearly showing their conformance. In Figure 7, we show two example inputs, the ground truth instance masks, and the objects' templates transformed using the predicted poses. Note that the input to the filtering module may have many instances; however, the rotation prediction model needs to predict the pose of the instance at the center of the patch, which is a difficult learning task.

In order to validate the generalizability of our method, we also compare its performance on grayscale images from the Ensenso camera, under the same training and test data settings as described above. Qualitative results and quantitative comparisons to Mask-RCNN (FT) and iFS-RCNN (FT) are provided in Figure 8 and Table I (last row), respectively. As is clear from these results, our method generalizes well to few-shot training on grayscale images.

C. Ablation Studies

In this section, we provide ablation studies to gain insights into the various hyperparameters in our model.

How many instances are needed to train TrInSeg? In Figure 6(a), we plot the performance (mIoU) of the prediction model against the number of annotated examples needed for three of our object categories. We compare Mask-RCNN (FT) that uses only the original images and their instance annotations for training (along with standard augmentations) against our TransMixup method. As is clear, while Mask-RCNN struggles to perform at a lower number of available training instances, our method demonstrates nearly 85% accuracy even when only a single instance is annotated.

Sensitivity of the filtering threshold? In Figure 6(b), we evaluate two properties of our prediction filtering scheme: i) how to select the template conformance threshold η_c and ii) what fraction of the ground truth instances are retrieved by the method for a given threshold. For the latter, we compute the ratio of the number of instances returned by the filtering module against the total number of instances annotated in the image. We change the threshold from 0.05 to 2.0 in increments of 0.05. We use the same setting for all three object categories. As expected, the plot shows that when the threshold is low and reasonable, the accuracy of the retrieved instances is high (nearly 95%), but the number of instances retrieved is low (about 50%); increasing the threshold to higher values lead to a slight drop in performance while reaching 100% retrieval accuracy. We use the setting of 0.1 for the experiments reported above.

V. CONCLUSIONS

In this paper, we proposed a simple, modular, and efficient scheme for transparent object instance segmentation in a few-shot setting. Our key idea is to extract segments from the few annotated examples to produce synthetic examples, rendering these instances through alpha compositing. Our experiments show that this simple scheme can be used to effectively train

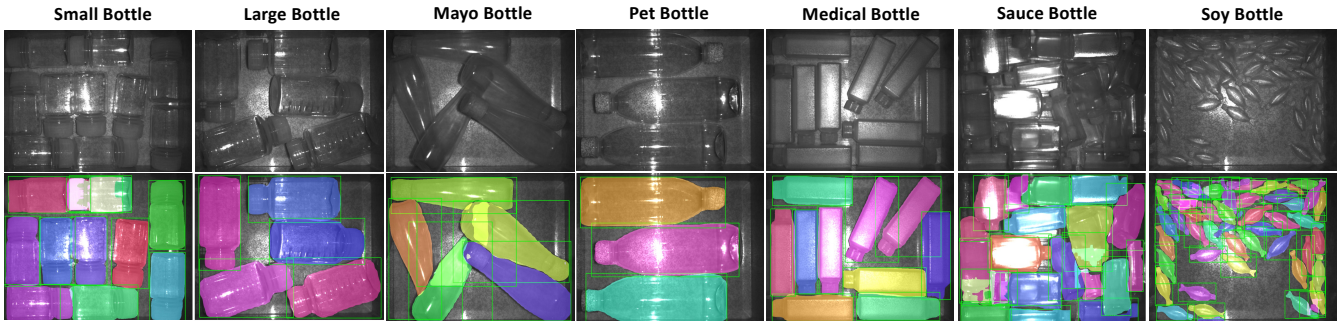


Fig. 8. *Top row:* Ensenso grayscale images from the seven categories. *Bottom row:* The final instance segmentations produced by TrInSeg.

a Mask-RCNN model to achieve high performance. Further, we also proposed a novel method for filtering the masks predicted by the backbone via using one of the annotated masks to form an object template and predicting a spatial transformation using a neural network. Going forward, we plan to integrate this approach with a grasp pose prediction scheme and test on a real robot.

REFERENCES

[1] J. Jiang, G. Cao, J. Deng, T.-T. Do, and S. Luo, "Robotic perception of transparent objects: A review," *Transactions on AI*, 2023.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017, pp. 2961–2969.

[3] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," in *ICCV*, 2023, pp. 4015–4026.

[4] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *ICCV*, 2019, pp. 9157–9166.

[5] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: state of the art," *Intl. journal of Multimedia Information Retrieval*, vol. 9, no. 3, pp. 171–189, 2020.

[6] Y. Cao, Z. Zhang, E. Xie, Q. Hou, K. Zhao, X. Luo, and J. Tuo, "Fakemix augmentation improves transparent object detection," *arXiv preprint arXiv:2103.13279*, 2021.

[7] A. H. Madessa, J. Dong, X. Dong, Y. Gao, H. Yu, and I. Mugunga, "Leveraging an instance segmentation method for detection of transparent materials," in *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing*. IEEE, 2019, pp. 406–412.

[8] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear grasp: 3d shape estimation of transparent objects for manipulation," in *ICRA*. IEEE, 2020, pp. 3634–3642.

[9] H. Fan, H. A. Miththanthaya, S. R. Rajan, X. Liu, Z. Zou, Y. Lin, H. Ling, *et al.*, "Transparent object tracking benchmark," in *ICCV*, 2021, pp. 10 734–10 743.

[10] E. Xie, W. Wang, W. Wang, P. Sun, H. Xu, D. Liang, and P. Luo, "Segmenting transparent object in the wild with transformer," *arXiv preprint arXiv:2101.08461*, 2021.

[11] A. Kalra, V. Taamazyan, S. K. Rao, K. Venkataraman, R. Raskar, and A. Kadambi, "Deep polarization cues for transparent object segmentation," in *CVPR*, 2020, pp. 8602–8611.

[12] Y. Xu, H. Nagahara, A. Shimada, and R.-i. Taniguchi, "Transcut: Transparent object segmentation from a light-field image," in *ICCV*, 2015, pp. 3442–3450.

[13] L. Ma, B. Dong, J. Yan, and X. Li, "Matting enhanced mask r-cnn," in *ICME*. IEEE, 2021, pp. 1–6.

[14] M. Durner, W. Boerdijk, M. Sundermeyer, W. Friedl, Z.-C. Márton, and R. Triebel, "Unknown object segmentation from stereo images," in *IROS*. IEEE, 2021, pp. 4823–4830.

[15] J. Lee, S. Back, T. Kim, S. Shin, S. Noh, R. Kang, J. Kim, and K. Lee, "Fusing rgb and depth with self-attention for unseen object segmentation," in *ICCAS*. IEEE, 2021, pp. 1599–1605.

[16] Y. Zhang and T. Funkhouser, "Deep depth completion of a single rgb-d image," in *CVPR*, 2018, pp. 175–185.

[17] P. Raj, S. Bhadang, G. Chaudhary, L. Behera, and T. Sandhan, "Bin-picking of novel objects through category-agnostic-segmentation: RGB matters," *arXiv preprint arXiv:2312.16741*, 2023.

[18] X. Liu, S. Iwase, and K. M. Kitani, "Stereobj-1M: Large-scale stereo image dataset for 6d object pose estimation," in *ICCV*, 2021, pp. 10 870–10 879.

[19] X. Chen, H. Zhang, Z. Yu, A. Opipari, and O. Chadwicke Jenkins, "Clearpose: Large-scale transparent object dataset and benchmark," in *ECCV*. Springer, 2022, pp. 381–396.

[20] K. Halupka, R. Garnavi, and S. Moore, "Deep semantic instance segmentation of tree-like structures using synthetic data," in *WACV*. IEEE, 2019, pp. 1713–1722.

[21] T. Wang, X. He, and N. Barnes, "Glass object localization by joint inference of boundary and depth," in *ICPR*. IEEE, 2012, pp. 3783–3786.

[22] G. Chen, K. Han, and K.-Y. K. Wong, "Tom-net: Learning transparent object matting from a single image," in *CVPR*, 2018, pp. 9233–9241.

[23] P. Tripathi, "Weakly supervised transparent object segmentation in the wild," Ph.D. dissertation, State University of New York, 2021.

[24] E. Xie, W. Wang, W. Wang, M. Ding, C. Shen, and P. Luo, "Segmenting transparent objects in the wild," in *ECCV*. Springer, 2020, pp. 696–711.

[25] A. Okazawa, T. Takahata, and T. Harada, "Simultaneous transparent and non-transparent object segmentation with multispectral scenes," in *IROS*. IEEE, 2019, pp. 4977–4984.

[26] J. Jiang, G. Cao, A. Butterworth, T.-T. Do, and S. Luo, "Where shall I touch? vision-guided tactile poking for transparent object grasping," *Transactions on Mechatronics*, vol. 28, no. 1, pp. 233–244, 2022.

[27] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *ICCV*, 2019, pp. 6023–6032.

[28] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[29] X. Hu, R. Gao, S. Yang, and K. Cho, "Cagnet: A multi-scale convolutional attention method for glass detection based on transformer," *Mathematics*, vol. 11, no. 19, p. 4084, 2023.

[30] Y. Tang, J. Chen, Z. Yang, Z. Lin, Q. Li, and W. Liu, "Depthgrasp: depth completion of transparent objects using self-attentive adversarial network with spectral residual for grasping," in *IROS*. IEEE, 2021, pp. 5710–5716.

[31] L. Zhu, A. Mousavian, Y. Xiang, H. Mazhar, J. van Eenbergen, S. Debnath, and D. Fox, "RGB-D local implicit function for depth completion of transparent objects," in *CVPR*, 2021, pp. 4649–4658.

[32] H. Xu, Y. R. Wang, S. Eppel, A. Aspuru-Guzik, F. Shkurti, and A. Garg, "Seeing glass: joint point cloud and depth completion for transparent objects," *arXiv preprint arXiv:2110.00087*, 2021.

[33] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-NeRF: Using a neural radiance field to grasp transparent objects," *arXiv preprint arXiv:2110.14217*, 2021.

[34] Z. Li, Y.-Y. Yeh, and M. Chandraker, "Through the looking glass: neural 3D reconstruction of transparent shapes," in *CVPR*, 2020, pp. 1262–1271.

[35] R. Girshick, "Fast R-CNN," in *ICCV*, 2015, pp. 1440–1448.

[36] L. Medeiros, "Lang-Segment-Anything," https://github.com/luca-medeiros/lang-segment-anything, 2023.

[37] A. Cherian, S. Jain, T. K. Marks, and A. Sullivan, "Discriminative 3D shape modeling for few-shot instance segmentation," in *ICRA*. IEEE, 2023, pp. 9296–9302.

[38] K. Nguyen and S. Todorovic, "iFS-RCNN: An incremental few-shot instance segmenter," in *CVPR*, 2022, pp. 7010–7019.