IEEE 20th International Conference on Automation Science and Engineering (CASE), Bari, Italy, August 2024.

Learning Embeddings with Centroid Triplet Loss for Object Identification in Robotic Grasping

Anas Gouda[1,3], Max Schwarz[2,3], Christopher Reining[1], Sven Behnke[2,3], and Alice Kirchheim[1,3,4]

Abstract— **Foundation models are a strong trend in deep learning and computer vision. These models serve as a base for applications as they require minor or no further fine-tuning by developers to integrate into their applications. Foundation models for zero-shot object segmentation such as Segment Anything (SAM) output segmentation masks from images without any further object information. When they are followed in a pipeline by an object identification model, they can perform object detection without training. Here, we focus on training such an object identification model. A crucial practical aspect for an object identification model is to be flexible in input size (number of input images). As object identification is an image retrieval problem, a suitable method should handle multi-query multi-gallery situations without constraining the number of input images (e.g. by having fixed-size aggregation layers). The key solution to train such a model is the centroid triplet loss (CTL), which aggregates image features to their centroids. CTL yields high accuracy, avoids misleading training signals and keeps the model input size flexible. In our experiments, we establish a new state of the art on the ArmBench object identification task, which shows general applicability of our model. We furthermore demonstrate an integrated unseen object detection pipeline on the challenging HOPE dataset, which requires fine-grained detection. There, our pipeline matches and surpasses related methods which have been trained on dataset-specific data. Code and pretrained models are available.[5]**

I. Introduction

Object perception is a crucial prerequisite for many logistics applications, such as mixed bin picking, which received attention in the Amazon Robotics Challenge [2], [3]. Product verification is another application where object identification is required to eliminate mistakes, e.g., robots mistakenly picking dummy objects like packaging material. Further use cases include multi-order picking and handling of returned goods.

The number of unique objects handled along supply chains reaches millions, posing a significant challenge to object perception systems. Mainly due to the limitations of current object perception methods, many of the above use cases are still not automated, but performed by human operators.

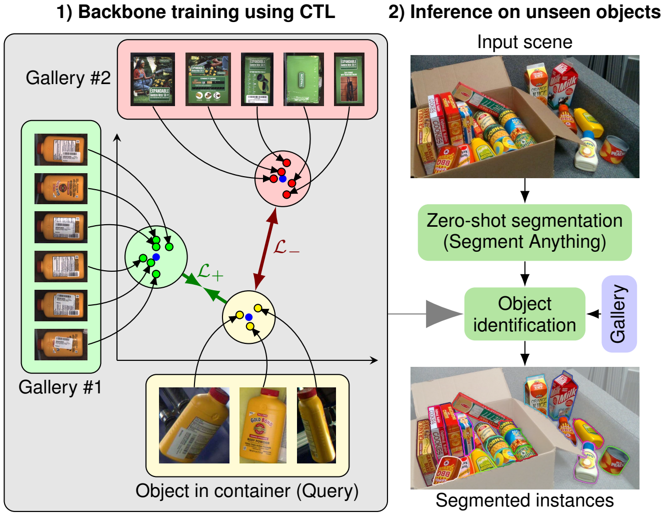

1) Backbone training using CTL
2) Inference on unseen objects
Gallery #2
Gallery #1
$\mathcal{L}_-$
$\mathcal{L}_+$
Object in container (Query)
Input scene
Zero-shot segmentation (Segment Anything)
Object identification
Gallery
Segmented instances

Fig. 1. Overview of our method. We train an object identification backbone on the large-scale ARMBench ID dataset using the centroid triplet loss (CTL [1]), such that the backbone learns to associate query images of objects in cluttered containers to matching gallery images. Crucially, the CTL loss operates on centroids in feature space, allowing the aggregation of an arbitrary number of input images. The trained backbone can then be used for identification of unseen objects that were segmented using a generic object segmentation method such as Segment Anything (SAM), given corresponding gallery images.

While deep-learning based methods have potential to enable automation of these applications, training data scarcity has prevented their breakthrough so far. Recent releases of public large-scale datasets such as ARMBench [4] and MegaPose [5] are potential game changers. In addition, images and data from warehouses themselves become available to researchers, as many retailers capture and offer images of all of their products. Some, such as Walmart, even provide API to retrieve these images for general use. These images could serve as reference data to identify objects in warehouse scenes.

Traditional methods for instance segmentation (such as Mask R-CNN [6] and its derivatives) assume an object set that is fixed at training time. This is a serious limitation as any change in the set of objects requires extensive retraining. Fine-tuning of segmentation models can mitigate this issue to some extent, but it requires careful attention and is still expensive. To address object set variability, methods for zero-shot and few-shot object identification are a suitable approach, as they do not require any adaptation. Recently, there have been big advancements in zero-shot object segmentation [7], [8]. Other advancements include category-agnostic template matching [9] and object identification based on multimodal large language models [10]. There are several shortcomings in these methods, though. Recent template matching methods as [9] handle only a single type of object; this is due to template images being fed early in the model layers. To detect multiple different object types,

[1] TU Dortmund, anas.gouda@tu-dortmund.de

[2] Autonomous Intelligent Systems - Computer Science VI & Center for Robotics, University of Bonn, Germany

[3] Lamarr Institute for Machine Learning and Artificial Intelligence

[4] Fraunhofer IML

[5] https://github.com/AnasIbrahim/ctl_classification

the methods need to be run repeatedly. This leads to long execution times that are unpractical for some applications. Furthermore, template matching methods are hard to enhance as the segmentation and object identification/matching are developed as a single black box. Fine-tuning a model to eliminate specific drawbacks might not be possible. Methods for object identification such as RoboLLM [10] tailor their query-input size to the application. If the number of query images changes, the model needs to be retrained. This retraining might not even be possible for applications where query images are collected automatically, which is a common practice in many of off-the-shelf-software. RoboLLM can also handle only one query object at a time.

In this paper, we assume the availability of a generic zero-shot segmentation method and focus on the object identification task, i.e. determining which class a segment belongs to. Our objective is to develop a method for object identification that is flexible in the number of input images and scalable in the number of objects. Fig. 1 illustrates our approach for object identification in the context of robotic grasping. We train a backbone network that maps object images to embeddings in an abstract feature space using a centroid triplet loss (CTL) [1]. In this feature space, our approach matches pre-captured object images (gallery images) to query images, which are generated by a zero-shot segmentation model or by application-specific segmentation models. Our method allows for processing any number of gallery and query objects, both described by an arbitrary number of images.

Our contributions include:

1) an approach for training object identification backbones with the centroid triplet loss on large-scale datasets,
2) evaluation of the backbones on ARMBench, where we establish a new state of the art,
3) an integrated architecture for unseen object instance segmentation with said backbone, and
4) evaluation and ablation of the entire pipeline on the HOPE dataset, where we obtain comparable performance to a method trained with object information.

In Section II, we discuss related work in detail. In Section III, we provide details on the centroid triplet loss and the training process for the ARMBench dataset. We report the evaluation results in Section IV.

II. Related Work

Pipelines working with unseen objects typically include multiple stages with separate deep neural networks. As shown in Fig. 2, typically these stages are zero-shot object segmentation, object identification or matching, and 6D localization of unseen objects. The first and second stage combined represent a module by themselves for 2D segmentation of unseen objects. Large-scale datasets with high variation are also a pillar for training these DNNs. This section provides an overview of current research in each stage and elaborates on its limitations.

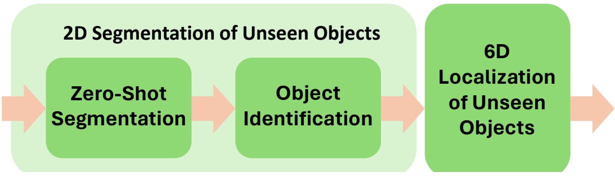

2D Segmentation of Unseen Objects
Zero-Shot Segmentation
Object Identification
6D Localization of Unseen Objects

Fig. 2. Typical stages for handling unseen objects. Zero-shot segmentation and object identification/matching represent by themselves a pipeline for 2D segmentation of unseen objects. Another stage is then optionally added to perform 6D pose estimation of unseen objects.

The development of deep neural networks for zero-shot object segmentation went through different stages. The problem in the context of robotic grasping was known as *category-agnostic* and *unseen object* instance segmentation. SD-Mask-RCNN [11], UOIS [12], MSMFormer [13], and INSTR [14] introduced different network architectures for RGB, depth, or RGB-D modalities. Recently, the Segment Anything Model (SAM) [7] introduced the promptable segmentation task, where a prompt can be a set of points, a bounding box, or even text. With a prompt grid covering the input image, SAM can even be applied in a zero-shot ”segment everything” mode, returning a full (over-)segmentation of the scene. In our approach, we use SAM as the segmentation stage.

Classification can be either object-specific of category-specific. Here we focus on the object identification/matching which is object-specific. The object identification problem falls under the image retrieval problem. This area was lacking datasets that are big enough to achieve practical performance until the release of the ARMBench dataset [4]. The dataset contains multiple images both for query and for gallery (set of object's pre-captured images). The evaluation is carried out in two situations: First using a single image of the object in a bin before picking (pre-pick) and second using multiple (3) images after grasping the object in isolation (called post-pick). The post-pick includes the pre-pick image. RoboLLM [10] achieved great accuracy on the ARMBench dataset. It uses a variant of the BEiT3 [15] model followed a Multi-Layer Perceptron (MLP) for feature aggregation. A drawback of using MLPs is that the network can only use a fixed number of query-images. RoboLLM trained two different models to carry out the evaluation for pre-pick and post-pick situations. In contrast, our approach is flexible regarding the backbone choice and allows any number of query or gallery images. It can also match multiple query objects at once.

Deep template matching is an analogous approach to 2D segmentation of unseen objects. HU et al. [9] and DTOID [16] introduced different approaches to detect and segment objects using only a few gallery images. While architecturally pleasing, combining segmentation and identification in a single model is a hurdle that makes models harder to develop and, most importantly, harder to analyze shortcomings. In contrast, depending on a separate zero-shot segmentation module allows us to leverage modern foundation models for

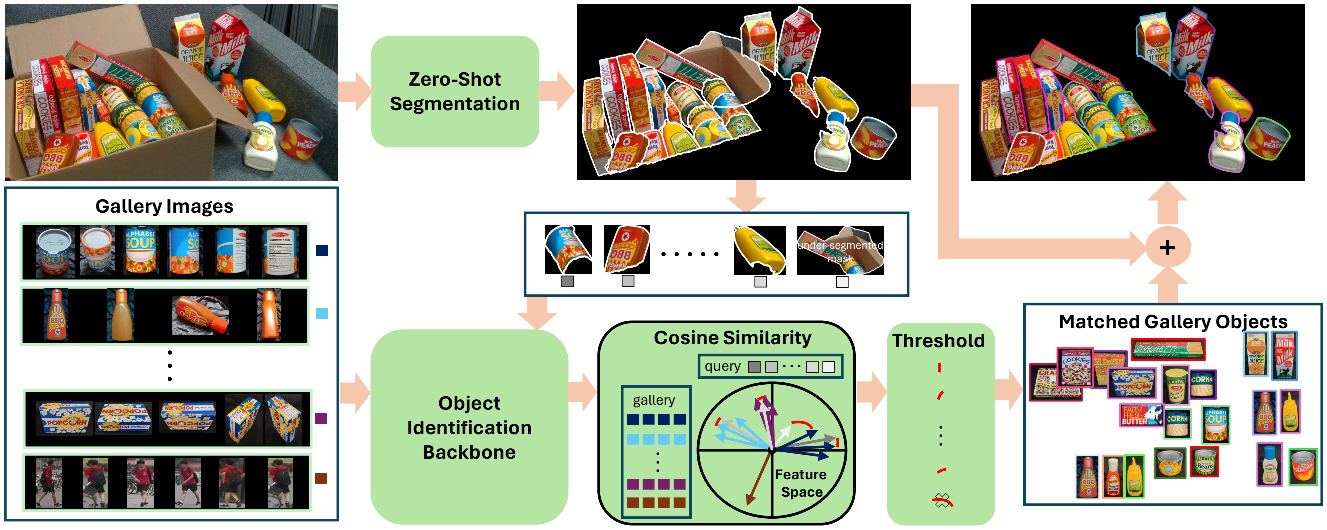


Fig. 3. Full unseen object detector pipeline. A zero-shot segmentation method removes the background and produces object segments, which may be over- or undersegmented and can overlap. Features are extracted by the identification backbone on both gallery images and segments. After finding closest matches in feature space using cosine similarity, badly or un-matched items are rejected by a thresholding operation. Finally, the matches can be used to construct a full instance segmentation of the scene.

this task and results in easy analysis of segmentation and classification performance.

DoUnseen [17] and CNOS [18] pipelines focus on the 2D segmentation of unseen objects. DoUnseen uses a variant of Mask R-CNN to extract object segments followed by a ViT model pre-trained on ImageNet for the object identification. CNOS uses SAM followed by DINOv2 model for the object identification. SAM-6D [19] follows a similar scheme as CNOS for the 2D segmentation followed by a stage for 6D localization trained using the MegaPose [5] dataset. These three pipelines follow the scheme shown in Figure 2 and can enhance their performance by replacing their object identification/matching models with our backbone as we surpass DINOv2 scores as shown in Section IV.

III. Method

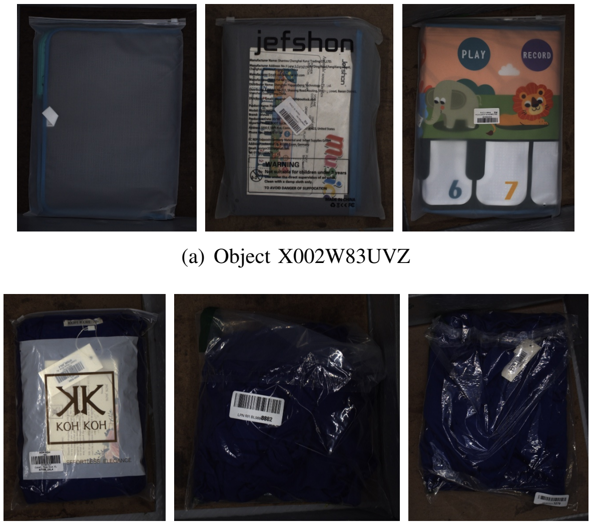
In this section, we explain the centroid triplet loss and training details necessary for applying it efficiently on large-scale multi-query datasets such as ARMBench.

A. Centroid Triplet Loss for Object Identification

How object images should be fed to a model for training is different to the related image retrieval and association tasks (e.g. person re-identification). In person re-identification datasets any of the gallery images could be used as a positive sample as humans tend to look similar from different perspectives. But in case of objects for robotic grasping, different faces of an object can look very different from each other. As shown in Figure 4, the front of an object can be of different color and texture from the back or the side. Training on single views would discard the relationship to the other views, which is a valuable training signal. Therefore, an object should be fed to the model as a whole. A solution for this is to use the centroid triplet loss (CTL) as inspired by [1]. As shown in Fig. 1, all images of an object (both query or gallery) are fed to a backbone and the resulting features are aggregated to their mean. The triplet loss is then calculated as follows:

$$\mathcal{L}_{\text{triplet}} = \max\left(\underbrace{\|C_a - C_p\|_2}_{\mathcal{L}_+} - \underbrace{\|C_a - C_n\|_2}_{\mathcal{L}_-} + \alpha, 0\right), \quad (1)$$

where C_a, C_p and C_n are the centroids of the query object, the positive gallery object and the negative gallery object, respectively. Finally, α defines the margin of this loss function.


(a) Object X002W83UVZ

(b) Object X0013DYNU7

Fig. 4. Example objects from ARMBench that have gallery images differing largely in texture. Treating each image on its own as a possible match loses a valuable training signal—these belong together. Using CTL, we treat objects as a whole.

Since the number of images per object is not constant for

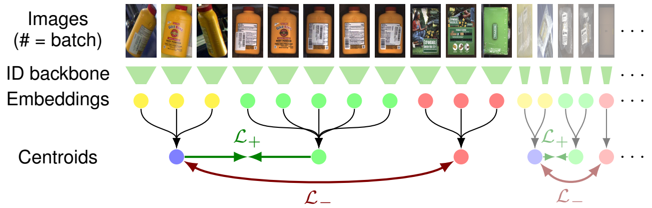


Fig. 5. Batch computations during training. Each batch is filled with triplets until the batch size is reached. Backbone feature vectors are extracted in batched fashion. The resulting features are then aggregated to their corresponding centroid using an *index-add* operation. Finally, the losses are applied.

both query and galleries, efficient training requires careful batch management. To keep an optimal batch size B, we greedily keep adding triplets of the query as well as positive and negative galleries with their corresponding images to the batch until B is reached (see Fig. 5). The embeddings are then extracted in typical batched fashion over all images. Crucially, recording a corresponding centroid index i_c for each image allows efficient batched summation and division, so that each object's centroid can be computed directly. Finally, the centroid triplet loss can be applied for each triplet, also in batched fashion. This method of batching and aggregation ensures optimal GPU utilization and scales to multi-GPU training by centrally pre-computing the batch splits for each epoch and then dividing the total number of batches across GPUs.

During inference, we need a method to match the query centroid q and the gallery images. For this, we use the cosine similarity score

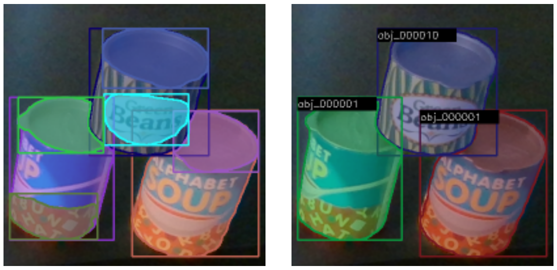
$$s(x,y) = 1 - \frac{x \cdot y}{\|x\|_2 \|y\|_2} \tag{2}$$

and select the most similar gallery entry g with maximum $s(q,g)$. We discard matches with $s < \theta$, where θ is a hyperparameter. This will not only omit bad association, but will particularly stop matching non-existing objects in the query or the scene. For evaluation on the ARMBench object test set this is not necessary, but the thresholding plays an important role when evaluating the whole unseen object instance segmentation pipeline.

B. 2D Segmentation of Unseen Objects

Figure 3 shows our full pipeline for the object detection.

For the zero-shot segmentation we use a combination of Mask R-CNN from [6] and SAM [7]. The Mask R-CNN is only responsible for background extraction. The Mask R-CNN is trained on NVIDIA Falling Things dataset [20] with only one class representing any warehouse object. The image is then segmented by SAM and any segmentation masks belonging to the background (according to Mask R-CNN) are discarded. We note this method as (Mask R-CNN+SAM) in our upcoming evaluation. As SAM is prompt-based, SAM is prompted with a grid of points to segment all object in the image. This can lead to many over- and under-segmented masks that even overlap. We depend on the thresholding operation defined above to filter the over- and under-segmented masks as shown in Section IV.



(a) segmentation output (b) after classification

Fig. 6. Filtration of over/under-segmented masks by our object identification model. The filtration also handles removing of background segments and objects in gallery that are not present in the image.

IV. Experiments

In this section, we carry out two evaluations: first on the test set of the ARMBench object identification dataset [4] and, second, on the HOPE dataset [21].

A. Object Identification on ARMBench

To train a model for object identification a dataset of thousands of objects is required. This is what the object identification section of the ARMBench dataset [4] provides. It contains 190K gallery objects (Reference images) with multiple images for each gallery object and contains 235K query scene (Picks) also with multiple images for each query object. This large number of objects provides enough data to train a model that is able to generalize to new objects at inference. From the 235K query scenes, 50k are used for test.

We follow the evaluation protocol in [10] and report the Recall@k metric for $k \in 1, 2, 3$. We also differentiate between the pre-pick and post-pick situations, where pre-pick is the captured image of the object inside the bin before the robot arm grasps it. The post-pick situation includes the pre-pick image and other images of the object while it is grasped by the robot.

Our approach is generic to the actual backbone architecture. In our evaluation, we focus on ResNet [26] and ViT [27]. Both models were pretrained on ImageNet. For ResNet we select ResNet-50 and train for 100 epochs (1 week) on one A100 GPU with a learning rate of 1e-3, 1e-4 and 1e-5 for 40, 30 and 30 epochs, respectively. For ViT, we use a ViT-b-16 instance and train for 100 epochs on eight A100 GPUs (approximately three days) using SGD with a learning rate of 0.05. Following Kumar et al. [28], we freeze the first layer of the ViT model. To improve robustness, we train an additional 100 epochs with data augmentation (TrivialAugment [29]).

To prepare the model for single query images, we select the first query image of ARMBench instead of the query centroid during training with a random chance of 50%.

TABLE I

EVALUATION ON ARMBENCH OBJECT IDENTIFICATION TEST SET

			Recall@1		Recall@2		Recall@3	
Method	# query images	Trained on	pre	post	pre	post	pre	post
ResNet50-RMAC [22]	any	ImageNet	71.7	72.2	81.9	82.9	87.2	88.2
DINO-ViTS [23]	any	ImageNet	77.2	79.5	87.3	89.4	91.6	93.5
DINO-V2 [24]	any	ImageNet	72.3	75.1	84.2	87.5	89.7	92.6
ViT-b-16-CTL-instance (ours)	any	ArmBench	97.2	**99.3**	97.8	99.4	98.3	99.6
ViT-b-16-CTL-centroid (ours)	any	ArmBench	97.2	98.6	**99.0**	**99.5**	**99.4**	**99.7**
Resnet-50-CTL-instance (ours)	any	ArmBench	88.4	97.0	90.8	98.0	92.6	98.5
Resnet-50-CTL-centroid (ours)	any	ArmBench	86.9	94.3	94.5	98.1	96.9	99.0
RoboLLM [10]	1 or 3	ArmBench	**97.8**	98.0	97.9	98.1	98.0	98.2

We report the Recall@k metric for $k \in 1, 2, 3$ for the pre-pick and post-pick situations. Our ViT model scores the highest accuracy on the test set for the multi-query (post-pick) situation. Using the closest gallery object instance (shown as ”-instance”) gives higher accuracy than searching for the closest object centroid (”-centroid”) and is thus recommended for inference.

TABLE II

EVALUATION ON THE HOPE VALIDATION SET (BOUNDING BOXES)

Method	Training on HOPE	AP	AP50	AP75	AP$_M$	AP$_L$	AR$_1$	AR$_{10}$	AR$_{100}$	AR$_M$	AR$_L$
Mask R-CNN	HOPE-Video	0.196	0.377	0.206	0.090	0.200	0.231	0.294	0.294	0.087	0.298
Bonani et al. [25]	Meshes + unlabeled HOPE-Video	0.338	**0.552**	0.364	0.172	**0.380**	**0.387**	**0.452**	**0.452**	**0.220**	**0.457**
SAM + DINOv2	none	0.316	0.431	0.346	**0.180**	0.317	0.339	0.383	0.383	0.217	0.386
SAM + ViT-CTL (Ours)	none	**0.349**	0.494	**0.377**	0.105	0.367	0.384	0.438	0.438	0.148	0.447
DINOv2 (GT masks)	none	0.581	0.581	0.581	0.277	0.587	0.582	0.671	0.671	0.275	0.679
Ours (GT masks)	none	0.740	0.740	0.740	0.277	0.755	0.680	0.776	0.776	0.275	0.791

We report the standard COCO metrics. Note: AP$_S$ and AR$_S$ are not applicable, since the dataset does not contain ”small” segments. Highest numbers (except for ground truth baselines) are highlighted in bold. The segmentation method (Mask R-CNN+SAM) is simply denoted (SAM) in this table.

TABLE III

EVALUATION ON THE HOPE VALIDATION SET (SEGMENTATION)

Method	Training on HOPE	AP	AP50	AP75	AP$_M$	AP$_L$	AR$_1$	AR$_{10}$	AR$_{100}$	AR$_M$	AR$_L$
Mask R-CNN	HOPE-Video	0.182	0.354	0.188	0.031	0.186	0.212	0.272	0.272	0.030	0.279
Bonani et al. [25]	Meshes + unlabeled HOPE-Video	0.333	**0.564**	0.378	**0.202**	0.373	0.371	0.434	0.434	**0.240**	0.441
SAM + DINOv2	none	0.337	0.436	0.371	0.171	0.340	0.359	0.405	0.405	0.217	0.409
SAM + ViT-CTL (Ours)	none	**0.374**	0.520	**0.403**	0.099	**0.395**	**0.405**	**0.462**	**0.462**	0.135	**0.472**
DINOv2 (GT masks)	none	0.581	0.581	0.581	0.277	0.587	0.582	0.671	0.671	0.275	0.679
ViT-CTL (GT masks)	none	0.740	0.740	0.740	0.277	0.755	0.680	0.776	0.776	0.275	0.791

We report the standard COCO metrics. Note: AP$_S$ and AR$_S$ are not applicable, since the dataset does not contain ”small” segments. Highest numbers (except for ground truth baselines) are highlighted in bold. The segmentation method (Mask R-CNN+SAM) is simply denoted (SAM) in this table.

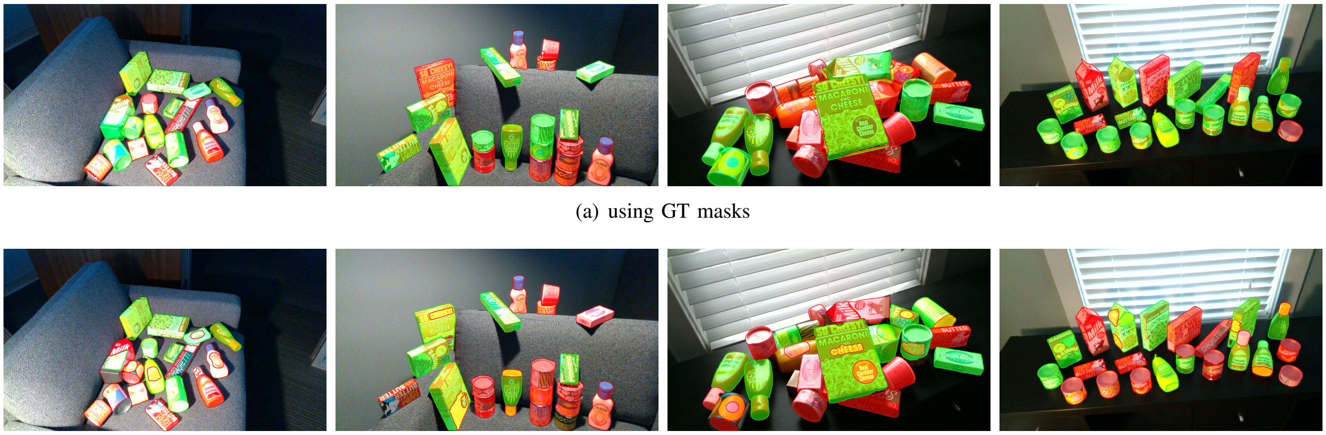
Table I shows the result of our trained ViT and ResNet. We compare our model to ResNet-50-RMAC [22] and DINO-ViTS [23] which are trained on ImageNet as evaluated in the ARMBench dataset paper [4]. Our variant with ViT scores the highest post-pick results among all methods. It is also worth noting that the accuracy of the ViT increases from 97.2 % to 99.3 % when using multiple query images (post-pick) instead of a single image (pre-pick). This shows the accuracy can increase with the addition of more query images, which is useful in applications where more query images are collected automatically.

B. 2D Segmentation of Unseen Objects

In this section we evaluate the whole pipeline on the HOPE[6] dataset (validation split). The reason we choose the HOPE dataset is that its objects look very similar, requiring fine-grained classification, which is challenging for an object identification model and tests the hard upper limits of the model. The evaluation is done using the COCO metrics for bounding box detection and segmentation. Similar to other datasets in the BOP format, the HOPE dataset offers modal (visible) and amodal (full) masks of the objects in 2D. Here we evaluate with modal masks. The gallery images are taken manually for each object covering all the unique faces of each objects. The gallery images are then augmented by rotating each image multiples of 45 degree.

We offer two baselines: First, we train a Mask R-CNN

[6] Evaluation was conducted using HOPEv1 [21]. A more recent version, HOPEv2, was subsequently released in May 2024 for the BOP challenge 2024.



(a) using GT masks

(b) using Mask R-CNN+SAM

Fig. 7. Qualitative examples of segmentation and identification on the HOPE dataset. Green color represents objects that have been segmented and identified correctly. Red color represents segments that are identified as a wrong class.

model in supervised fashion for instance segmentation using the HOPE-video dataset [30]. We note that this baseline requires an annotated training set. The second baseline considers the case where object models and unlabeled real data is available. For this unlabeled data the HOPE-Video dataset is used but the labels are disregarded and not used. For this second baseline, we apply the method of Bonani et al. [25], which trains a semantic segmentation network in supervised fashion on synthetic data, and uses SAM to regularize the network's output on the unlabeled real data. We note that this method produces a semantic segmentation, which is then converted to instance segmentation by finding connected components. This adaptation is obviously sub-optimal when objects of the same class overlap.

We evaluate our pipeline for 2D segmentation of unseen objects once with a pre-trained DINOv2 backbone and once with our object identification model. Each backbone is again evaluated twice: Once with the full pipeline, and once with ground truth segment masks. The evaluation with the ground truth masks helps estimate the contribution of the zero-shot segmentation and our object identification model to performance. For these evaluations we use a cosine similarity score threshold of $\theta = 0.6$. As described in Section III (Eq. (2)), if score of the best match is lower than θ we discard this query segment as it might be an object that is not present in the gallery or an over-/under-segmentation.

Tables II and III show quantitative results with metrics calculated on bounding boxes and segmentation masks, respectively. Interestingly, our method clearly outperforms Mask R-CNN, which was trained in supervised fashion. This may indicate that the (comparably small) domain shift between HOPE and HOPE-Video is difficult for this purely supervised method to overcome. Furthermore, the method of Bonani et al. [25] also beats Mask R-CNN, showing the usefulness of synthetic data generated from the object models, even in the absence of annotations on real training data. Our full pipeline matches and even surpasses this performance in most metrics, which is highly interesting as this baseline has access to object models and unlabeled real data. Finally, we can see that our object identification model surpasses DINOv2 across all COCO metrics. When removing the effects of segmentation & segment filtering by using ground truth masks, our model scores an AP of 0.740 against DINOv2 with an AP of 0.581 making our model score 27.4% higher than DINOv2. This supports our claim that our model can enhance pipelines for unseen object detection that are using DINOv2 such as CNOS [18] or SAM-6D [19].

Another interesting result is the filtration our object identification model can contribute. Figure 6 (a) shows the output of the zero-shot segmentation with SAM. Figure 6 (b) shows how our model filters the over-segmented masks as they score below the threshold.

Finally, we show exemplary qualitative results in Fig. 7.

V. Conclusion

In this work we introduced how to use the centroid triplet loss for training models for object identification. We successfully showed that CTL scores the highest accuracy on the ARMBench test set. In particular, the trained model is able to process any number of query or gallery images. It is capable of matching multiple query object to multiple gallery object at once, shortening inference time. The model performance surpasses DINOv2 on ARMBench and can serve as an improved backbone for 2D segmentation and 6D localization of unseen objects. The near-perfect score on ARMBench indicates that it can be used in real-world applications.

When combined with a generic zero-shot segmentation method such as SAM, the result is a complete segmentation pipeline. While performance on the challenging HOPE dataset is still limited in absolute terms, our pipeline beats several related methods that can access object information during training. Tighter integration with the segmentation module and intelligent proposal filtering might improve results further.

ACKNOWLEDGMENT

This work is funded by the German Federal Ministry of Education and Research (BMBF) in the course of the Lamarr Institute for Machine Learning and Artificial Intelligence.

REFERENCES

[1] M. Wieczorek, B. Rychalska, and J. Dabrowski, "On the unreasonable effectiveness of centroids in image retrieval," in *28th International Conference on Neural Information Processing (ICONIP)*, ser. Lecture Notes in Computer Science, vol. 13111. Springer, 2021, pp. 212–223.

[2] D. Morrison, A. W. Tow, M. Mctaggart, R. Smith, N. Kelly-Boxall, S. Wade-Mccue, J. Erskine, R. Grinover, A. Gurman, T. Hunn *et al.*, "Cartman: The low-cost Cartesian manipulator that won the Amazon Robotics Challenge," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7757–7764.

[3] M. Schwarz, C. Lenz, G. M. García, S. Koo, A. S. Periyasamy, M. Schreiber, and S. Behnke, "Fast object learning and dual-arm coordination for cluttered stowing, picking, and packing," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3347–3354.

[4] C. Mitash, F. Wang, S. Lu, V. Terhuja, T. Garaas, F. Polido, and M. Nambi, "ARMBench: An object-centric benchmark dataset for robotic manipulation," in *International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9132–9139.

[5] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, and J. Sivic, "MegaPose: 6D pose estimation of novel objects via render & compare," in *Conference on Robot Learning (CoRL)*, 2022.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.

[7] 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 *International Conference on Computer Vision (ICCV)*, 2023, pp. 4015–4026.

[8] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang, "Fast segment anything," *arXiv preprint arXiv:2306.12156*, 2023.

[9] Z. Hu, R. Tan, Y. Zhou, J. Woon, and C. Lv, "Template-based category-agnostic instance detection for robotic manipulation," *Robotics and Automation Letters (RA-L)*, vol. 7, no. 4, pp. 12 451–12 458, 2022.

[10] Z. Long, G. Killick, R. McCreadie, and G. A. Camarasa, "RoboLLM: Robotic vision tasks grounded on multimodal large language models," 2023.

[11] M. Danielczuk, M. Matl, S. Gupta, A. Li, A. Lee, J. Mahler, and K. Goldberg, "Segmenting unknown 3D objects from real depth images using Mask R-CNN trained on synthetic data," in *Int. Conf. Robotics and Automation (ICRA)*, 2019.

[12] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "Unseen object instance segmentation for robotic environments," *IEEE Transactions on Robotics (T-RO)*, 2021.

[13] Y. Lu, Y. Chen, N. Ruozzi, and Y. Xiang, "Mean shift mask transformer for unseen object instance segmentation," *arXiv preprint arXiv:2211.11679*, 2022.

[14] M. Durner, W. Boerdijk, M. Sundermeyer, W. Friedl, Z.-C. Márton, and R. Triebel, "Unknown object segmentation from stereo images," in *International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4823–4830.

[15] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som, and F. Wei, "Image as a foreign language: BEiT pretraining for vision and vision-language tasks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[16] J.-P. Mercier, M. Garon, P. Giguère, and J.-F. Lalonde, "Deep template-based object instance detection," in *Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1506–1515.

[17] A. Gouda and M. Roidl, "DoUnseen: Tuning-free class-adaptive object detection of unseen objects for robotic grasping," 2023.

[18] V. N. Nguyen, T. Groueix, G. Ponimatkin, V. Lepetit, and T. Hodan, "CNOS: A strong baseline for CAD-based novel object segmentation," in *International Conference on Computer Vision (CVPR)*, 2023, pp. 2134–2140.

[19] J. Lin, L. Liu, D. Lu, and K. Jia, "SAM-6D: Segment anything model meets zero-shot 6d object pose estimation," *arXiv preprint arXiv:2311.15707*, 2023.

[20] J. Tremblay, T. To, and S. Birchfield, "Falling things: A synthetic dataset for 3D object detection and pose estimation," in *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018, pp. 2038–2041.

[21] S. Tyree, J. Tremblay, T. To, J. Cheng, T. Mosier, J. Smith, and S. Birchfield, "6-DoF pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark," in *International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[22] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of CNN activations," *arXiv preprint arXiv:1511.05879*, 2015.

[23] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *International Conference on Computer Vision (ICCV)*, 2021, pp. 9650–9660.

[24] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski, "Vision transformers need registers," *arXiv preprint arXiv:2309.16588*, 2023.

[25] M. E. Bonani, M. Schwarz, and S. Behnke, "Learning from SAM: Harnessing a segmentation foundation model for Sim2Real domain adaptation through regularization," *arXiv preprint arXiv:2309.15562*, 2023.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[28] A. Kumar, R. Shen, S. Bubeck, and S. Gunasekar, "How to fine-tune vision models with SGD," *arXiv preprint arXiv:2211.09359*, 2022.

[29] S. G. Müller and F. Hutter, "TrivialAugment: Tuning-free yet state-of-the-art data augmentation," in *International Conference on Computer Vision (ICCV)*, 2021, pp. 774–782.

[30] Y. Lin, J. Tremblay, S. Tyree, P. A. Vela, and S. Birchfield, "Multi-view fusion for multi-level robotic scene understanding," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6817–6824.