Team VI-I2R-FF Technical Report on EPIC-KITCHENS VISOR Hand Object Segmentation Challenge 2023

Fen Fang, Yi Cheng, Ying Sun and Qianli Xu
Institute for Infocomm Research, A*STAR, Singapore
{fang_fen, cheng_yi, suny, Qxu}@i2r.a-star.edu.sg

Abstract

In this report, we present our approach to the EPIC-KITCHENS VISOR Hand Object Segmentation Challenge, which focuses on the estimation of the relation between the hands and the objects given a single frame as input. The EPIC-KITCHENS VISOR dataset provides pixel-wise annotations and serves as a benchmark for hand and active object segmentation in egocentric video. Our approach combines the baseline method, i.e., Point-based Rendering (PointRend) and the Segment Anything Model (SAM), aiming to enhance the accuracy of hand and object segmentation outcomes, while also minimizing instances of missed detection. We leverage accurate hand segmentation maps obtained from the baseline method to extract more precise hand and in-contact object segments. We utilize the class-agnostic segmentation provided by SAM and apply specific hand-crafted constraints to enhance the results. In cases where the baseline model misses the detection of hands or objects, we re-train an object detector on the training set to enhance the detection accuracy. The detected hand and in-contact object bounding boxes are then used as prompts to extract their respective segments from the output of SAM. By effectively combining the strengths of existing methods and applying our refinements, our submission achieved the 1st place in terms of evaluation criteria in the VISOR HOS Challenge.

1. Introduction

VISOR [2], a new dataset containing pixel-wise annotations, introduces a cutting-edge benchmark specifically designed for segmenting hands and active objects within the context of EPIC-KITCHENS egocentric videos [1]. The dataset consists of various annotations including 272K manually annotated semantic masks representing 257 object classes, 9.9M interpolated dense masks, and 67K hand-object relations. It encompasses a total of 36 hours of 179 untrimmed videos. Specifically, there are 115 videos allocated for training, 43 videos (including 5 unseen ones) designated for validation, and 21 videos (including 4 unseen ones) reserved for testing purposes. The Hand Object Segmentation (HOS) challenge utilizes the dataset to tackle the task of segmenting instances of hands and objects that are in contact with each other. In addition to segmenting the hand and object regions, the challenge also requires participants to predict the hand side (left/right), the contact state (contact/no-contact), and the offset between the hand and the contacted object. The baseline method PointRend [5] demonstrates accurate hand segmentation, achieving an average precision (AP) of 0.909 and 0.954 on the validation and test sets, respectively. However, the task of segmenting hands in contact with objects presents significant challenges. The baseline method only achieves an AP of 0.307

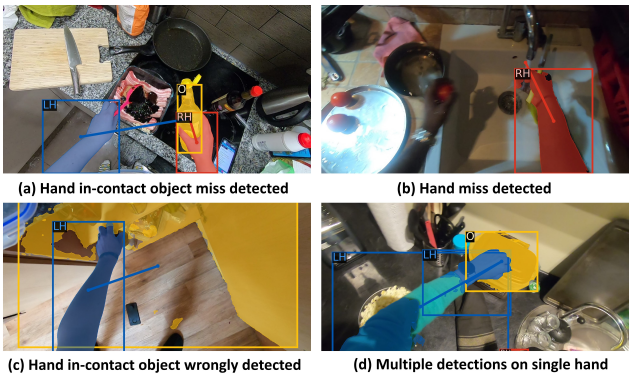

Figure 1. Illustration of the challenges in hand object segmentation on EPIC-KITCHENS VISOR dataset. (a) Missed detection of hand in-contact objects due to partial occlusion caused by objects like sauce on a spoon. (b) Hand misidentification due to blurred hand images. (c) Incorrect detection of hand in-contact objects where the majority of the object is occluded by the hand. (d) Single hand being detected as multiple parts due to strong contrast. To address these various challenges, we propose an extension to the baseline model that incorporates learning for both object segmentation and detection tasks. Additionally, we introduce hand-crafted constraints designed to effectively merge the outputs of segmentation, detection, and SAM.

and 0.337 on the validation and test sets for this specific task. In Fig 1, we present some challenging samples from the EPIC-KITCHENS VISOR dataset. As depicted in the figure, the detection of hands and objects may encounter challenges such as partial occlusion, motion blur, or intense variations in lighting, leading to potential misidentification or incorrect detection.

Recently, a large foundation model for image segmentation is released by Facebook, achieving high performance in zero-shot segmentation [4]. We leverage its capabilities to significantly improve segmentation accuracy. While SAM has successfully learned a broad understanding of objects and can generate masks for various objects (including unseen objects) in images or videos, it relies on prompts (*e.g.* clicking or drawing box) to indicate the desired object mask. Meanwhile, SAM's output is class-agnostic, meaning it lacks the ability to specifically identify and segment user-specified objects without guidance. In certain instances, the results obtained from the segmentation model can offer valuable guidance for improving segmentation derived from SAM's output. As illustrated in the first row of Fig. 2, despite the failure to segment the dish soap bottle accurately, the predicted contact information can be utilized to extract a partial mask of the bottle. However, in cases where the segmentation model fails to provide any useful information, as depicted in the second row of Fig. 2, even though SAM exhibits accurate segmentation capabilities, it struggles to extract the hands and in-contact objects.

Object detection aims to identify and localize objects within an image, while instance segmentation goes a step further by providing precise instance masks at the pixel level, along with their corresponding class names [7]. This additional level of detail enables instance segmentation to solve both the tasks of object detection and semantic segmentation simultaneously. Hence, object detection is generally considered less computationally demanding than object segmentation when applied to the same training set. By leveraging SAM's ability to utilize bounding boxes as prompts, we propose a solution to tackle missed hand and object segmentation. This involves training an object detection model and implementing tailored constraints for extracting masks from the outputs of segmentation, detection and SAM models.

The architecture of our proposed method is shown in Fig. 3. The system comprises three modules: object segmentation, object detection, and SAM module. The object segmentation module is trained using PointRend neural network, with hand and object masks as inputs. The object detection module is trained using the Faster-RCNN framework [8], utilizing hand and object boxes as inputs. Both modules are also trained to predict hand side (left or right), hand state (in-contact or non-contact), and offset vectors, in addition to detecting and segmenting hands and objects. The SAM module generates zero-shot object, hand, and scene segmentation results. By applying customized constraints to these three outputs, the module produces the final segmentation results for hands and objects.

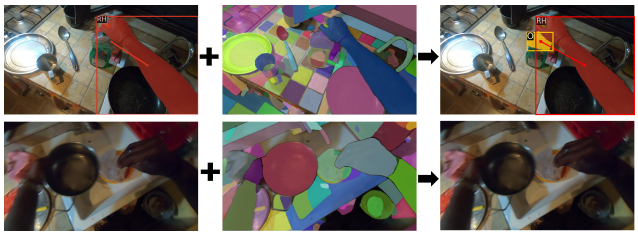

Figure 2. Samples that demonstrate whether the segmentation model can offer valuable information for extracting the desired hand or object segmentation.

2. Our Approach

In this section, we will delve into the technical intricacies of our proposed approach. Our overall architecture, as depicted in Fig. 3, consists of three main modules: PointRend segmentation, Faster-RCNN detection and SAM module, and a tailored constraints algorithm. For the SAM model, we employ it for zero-shot segmentation on the VISOR dataset without retraining it, so we do not delve into technical details of this module. These modules work in tandem to extract the final segmentation from the outputs they generate.

2.1. PointRend Hand and Object Segmentation

Following the baseline provided for the HOS task by the organizers, we utilized the PointRend instance segmentation model [5] implemented in Detectron2 [10]. The model was equipped with an R50-FPN backbone and trained using the standard 1× learning rate scheduled configuration by default. Training was performed for 90,000 iterations with a batch size of 24 and a base learning rate of 0.02. To cater specifically to the HOS task, we incorporated three additional linear layers as heads after the ROI-Pooled feature. These heads were designed to independently predict hand side (feature size: 2), contact state (feature size: 2), and offset (feature size: 3). During the training process, we employed Cross-Entropy loss for hand side and contact state prediction, while Mean Squared Error (MSE) loss was utilized for offset prediction.

According to [2], the model achieved accurate segmentation of hands when tested on a separate test set, with the side being easier to predict compared to the contact state. However, it is challenging to segment the objects accurately due to the long tail distribution of object classes and heavy occlusion. Meanwhile, there are several challenges on predicting hand contact state and segmenting objects in contact, including the difficulty in distinguishing hand

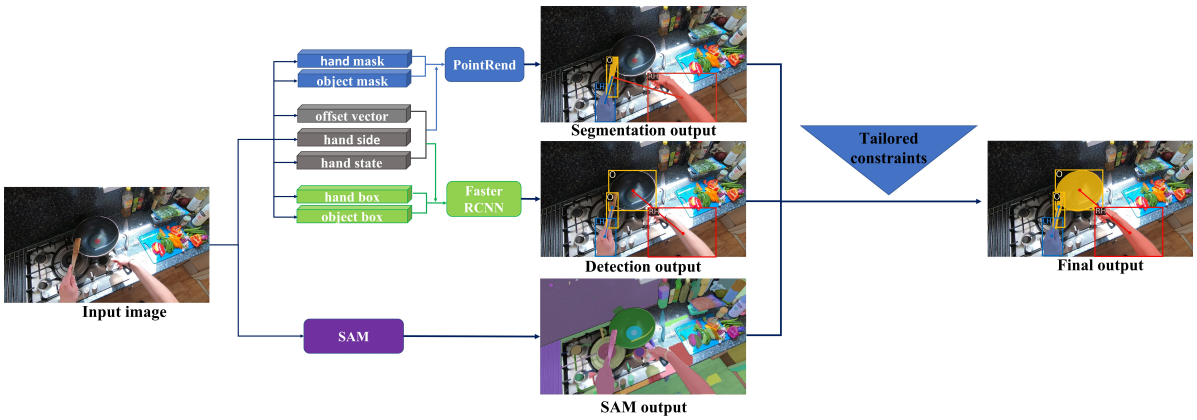

Figure 3. Overall architecture of the proposed framework. For the hand and object segmentation task, we employ the PointRend networks, utilizing hand and object mask annotations along with hand side, hand state, and offset vector. On the other hand, for the hand and object detection task, we utilize the Faster-RCNN training framework, incorporating hand and object boxes along with hand side, hand state, and offset vector. The segmentation and detection outputs are then fused with SAM's output and subjected to tailored constraints to generate the final output. This is best viewed in color.

overlap and contact, the diverse range of held objects, and occlusion caused by interactions between hands and objects or between multiple hands. Typically, leveraging multiple frames extracted densely from a video can enhance performance in these tasks. However, in the case of the test set for the Hand-Object Segmentation (HOS) task, images are sparsely extracted from videos, limiting the potential benefit from utilizing the continuity of action to improve performance.

2.2. Faster-RCNN Hand and Object Detection

Similar to the approach in [9], our system is constructed upon a well-established object detection framework, Faster-RCNN [8], with the inclusion of auxiliary predictions and losses per-bounding box. We intentionally selected Faster-RCNN as the foundation for its recognized performance in detection tasks, and any enhancements made to the base network are independent of our contributions. We utilize a ResNet-101 [3] backbone as the core architecture for our model. The backbone is trained for 10 epochs, employing a learning rate of 0.001, and utilizing a batch size of 4. Specifically, we extend Faster-RCNN's capabilities to detect two specific object classes: human hands and contacted objects. The bounding box annotations for both hand and object can be readily derived from their respective mask annotations. The network, akin to standard Faster-RCNN, predicts whether each anchor box contains an object, determines its category, and refines the bounding box regression adjustments—these aspects remain unchanged. Moreover, similar to the PointRend segmentation model, we incorporate auxiliary outputs (hand side, hand state and offset vector) within the Faster-RCNN model. These auxiliary outputs are derived from the same ROI-pooled features used for standard classification outputs.

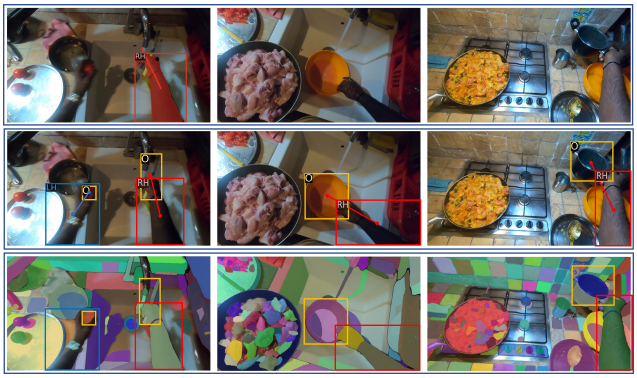

Figure 4. Performance comparison among segmentation (first row), detection (second row) and SAM model (last row). It is evident from the results that the detection model performs better than the segmentation model when trained on the same samples.

By considering the detection task as a form of dimension reduction for the segmentation task on a specific dataset, we can potentially achieve improved performance according to their respective evaluation metrics. This can be supported by visualized samples in Fig. 4. The first two rows showcase the segmentation and detection results on the same input images. It is evident that the detection model is capable of identifying some hand and object instances that were initially missed by the segmentation model. The third row demonstrates the impressive class-agnostic SAM output of the input images. This implies that with accurate bounding boxes for hands and objects, there is a high probability of successfully segmenting their respective masks.

2.3. Tailored Constraints for Mask Selection

Selecting the desired object masks from a specific region of the SAM output is not a straightforward task. The difficulties mainly arise from two aspects. Firstly, the mask of the desired object may not be the dominant region within the specified area, as object shapes can vary significantly. An illustrative example is given in the first image of Fig. 4, where the right hand's in-contact object (a knife) does not occupy the largest portion within the yellow box. Secondly, the SAM output itself may suffer from over-segmentation on objects. An instance of this can be seen in the first row of Fig. 2, where the dish soap bottle is segmented into four separate parts. Hence, we developed an algorithm that incorporates customized constraints specifically designed to address these challenges and thereby generating accurate object masks. In essence, our algorithm adheres to the following rules: (i) It prioritizes the selection of the hand mask followed by the object mask. (ii) Pixels belonging to the hand or object, as identified by the segmentation model, are given higher priority for inclusion in the final output. The detail algorithm is depicted in Algorithm 1.

Algorithm 1: Mask selection in a region

```
H for hand, O for object, P for parts in SAM ;
Constraints used for mask selection;
C1: whether the center of the region is on P (0 or 1);
C2: the ratio of P in the region;
C3: the ratio of the region pixels on P;
for each input image do
    Seg=segmentation_model(input) ;
    Det=detection_model(input) ;
    Sam=SAM_model(input) ;
    if H ∈ Det and H ∃ Seg then
        if iou(box_H_det, box_H_seg) > iou_th then
            H_final=H_seg;
        else
            H_id=iou(H_seg,P_sam).index(max(iou(H_seg,P_sam)));
            H_final=P_sam[H_id];
        end
    else
        if H ∈ Det and H not ∃ Seg then
            for each P in box_H_det do
                F=w1*C1+w2*C2+w3*C3;
                H_id=F.index(max(F));
                H_final=P_sam[H_id];
            end
        else
            continue;
        end
    end
    if O ∈ Det and O ∃ Seg then
        if iou(box_O_det, box_O_seg) > iou_th then
            O_final=O_seg;
        else
            Remove confirmed H from P_sam;
            O_id=iou(O_seg,P_sam).index(max(iou(O_seg,P_sam)));
            O_final=P_sam[O_id];
        end
    else
        if O ∈ Det and O not ∃ Seg then
            if contact point detected then
                O_final= P(contact point ∈ P)
            else
                F=w1*C1+w2*C2+w3*C3;
                O_id=F.index(max(F));
                O_final=P_sam[O_id];
            end
        else
            continue;
        end
    end
end
```

3. Experiments

3.1. Datasets

The EPIC-VISOR dataset (VISOR2022) consists of a training set comprising 32,857 images distributed across 242 classes. Additionally, there is a validation set containing 7,747 images from 182 classes, with 9 of these classes being unseen during training. Lastly, the dataset includes a test set consisting of 10,125 images belonging to 160 classes, among which 6 classes are unseen during training.

3.2. Implementation Details

The training detail of PointRend segmentation model and Faster-RCNN detection model are illustrated in sections 2.1 and 2.2. For the SAM model, we used the ViT-H version. All models and algorithms are trained and tested on a PyTorch platform using a machine equipped with 4 NVIDIA GeForce GPUs.

The evaluation of hand performance encompasses three schemes: (1) Hand Segmentation: This scheme focuses on hand segmentation tasks, utilizing the COCO Mask AP [6] as the evaluation metric. (2) Hand + Side: In addition to hand segmentation, this scheme involves hand side classification, distinguishing between left and right hands. (3) Hand + Contact: This scheme involves hand state classification, determining whether the hand is in contact or not. The performance evaluation of in-contact object segmentation task is conducted using the Mask AP metric. To further assess the explicit association between hands and in-contact objects, each hand is combined with its corresponding in-contact object mask (if applicable) and evaluate them as a

Table 1. The performance comparison between the baseline method and our method on the EPIC-KITCHENS-VISOR test set.

	Hand	Hand + Side	Hand + Contact	Object	Hand + Object
Baseline	0.9541	0.9238	0.7870	0.3373	0.60
Ours	0.9566	0.9515	0.7873	0.3511	0.6485

single class referred to as "Hand+Object". This allows for a comprehensive evaluation of the joint performance of hands and in-contact objects.

3.3. Results

We show the performance comparison between baseline segmentation model and our method on the test set in Table 1. Our method achieves marginal improvement on the hand segmentation task. This can be attributed to the fact that the baseline method already accurately segments the hands, achieving a high AP 0.954. Consequently, the scope for further enhancement is limited. For the in-contact object segmentation task, our method yields a gain of 1.4% in AP, which is noteworthy considering the test set included a total of 160 object classes, including 6 unseen classes. This achievement underscores the impressive nature of our AP gain. For hand side classification, our method exhibits an approximately 3% improvement in performance. This enhancement can potentially be attributed to the compensation provided by our detection model. In terms of hand state classification, our method achieves a slight improvement in accuracy over the baseline method. On hand and object combination, our method demonstrates a significant performance improvement of approximately 4.7%. This indicates that our approach is capable of delivering enhanced overall performance for both hand and in-contact objects in a joint manner.

4. Conclusion

In this report, we describe the technical details of our approach to the EPIC-KITCHENS VISOR HOS Challenge. Specifically, we propose a solution to tackle missed hand and object segmentation by involving training an object detection model and implementing tailored constraints for extracting masks from the outputs of segmentation, detection and SAM models. The experimental results on the EPIC-KITCHENS VISOR dataset demonstrate the effectiveness of our proposed method. Notably, our final submission ranks first on the leaderboard in terms of average of COCO Mask AP evaluation metric on five tasks (e.g., Hand segmentation, Hand+Side, Hand+Contact, Object segmentation, Hand+Object).

Acknowledgments

We would like to thank Dr. Joo Hwee Lim for his support and guidance. This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision. *Proceedings of the European Conference on Computer Vision*, 2018. 1

[2] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2022. 1, 2

[3] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. 3

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. 2

[5] A. Kirillov, Y. Wu, K. He, and R. Girshick. Pointrend: Image segmentation as rendering. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9796–9805, 2020. 1, 2

[6] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014. 4

[7] Sharma Rabi, Saqib Muhammad, C.T. Lin, and MICHAEL. Blumenstein. A survey on object instance segmentation. *SN Computer Science*, 2022. 2

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 2, 3

[9] D. Shan, J. Geng, M. Shu, and D. F. Fouhey. Understanding human hands in contact at internet scale. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9866–9875, 2020. 3

[10] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 2