Open-World Entity Segmentation

Lu Qi[*1] **Jason Kuen**[*2] **Yi Wang**[1] **Jiuxiang Gu**[2]
Hengshuang Zhao[3] **Zhe Lin**[2] **Philip Torr**[3] **Jiaya Jia**[1,4]
[1]The Chinese University of Hong Kong [2]Adobe Research
[3] University of Oxford [4]SmartMore

Abstract

We introduce a new image segmentation task, termed Entity Segmentation (ES) with the aim to segment all visual entities in an image without considering semantic category labels. It has many practical applications in image manipulation/editing where the segmentation mask quality is typically crucial but category labels are less important. In this setting, all semantically-meaningful segments are equally treated as categoryless *entities* and there is no *thing-stuff* distinction. Based on our unified entity representation, we propose a center-based entity segmentation framework with two novel modules to improve mask quality. Experimentally, both our new task and framework demonstrate superior advantages as against existing work. In particular, ES enables the following: (1) merging multiple datasets to form a large training set without the need to resolve label conflicts; (2) any model trained on one dataset can generalize exceptionally well to other datasets with unseen domains. Our code is made publicly available at https://github.com/dvlab-research/Entity.

1 Introduction

In recent years, image segmentation tasks (semantic segmentation [1–7], instance segmentation [8–16], and panoptic segmentation [17–21], *e.t.c*) have received great attention due to their diverse applications [22–27] and strong progress made possible by deep learning [28–34]. Most image segmentation tasks share the common goal of automatically assigning each image pixel to one of the predefined semantic categories. One of the key application areas of image segmentation is image manipulation [35–40] and editing [41–44]. Segmentation techniques have revolutionized image manipulation and editing applications by enabling users to operate directly on semantically-meaningful regions, as opposed to primitive image elements such as pixels and superpixels.

Although image segmentation holds much promise for user-friendly image manipulation and editing, there are two major weaknesses in image segmentation that adversely affect the image manipulation/editing experience: 1) category label confusion (often caused by having many predefined category labels); 2) lack of generalization for unseen categories. In Fig. 1, we present the segmentation results from a state-of-the-art panoptic segmentation method [20] that demonstrate the two weaknesses separately. These weaknesses can be largely attributed to the standard practice of training models to segment strictly based on the predefined categories.

In image manipulation and editing applications, where category labels are typically not required, the conventional category-oriented image segmentation may be sub-optimal and could introduce unnecessary category-related issues. Is there a better alternative to category-oriented image segmentation? Yes, it comes from omitting the categorization subtask and focusing solely on the segmentation subtask, in a similar spirit to the category-agnostic Region Proposal Network [45] in object detection. This is analogous to infants capable of distinguishing objects by shapes and appearances without knowing the object names [46]. Without the need to assign category labels, the category confusion

[*]Equal Contribution

Preprint. Under review.

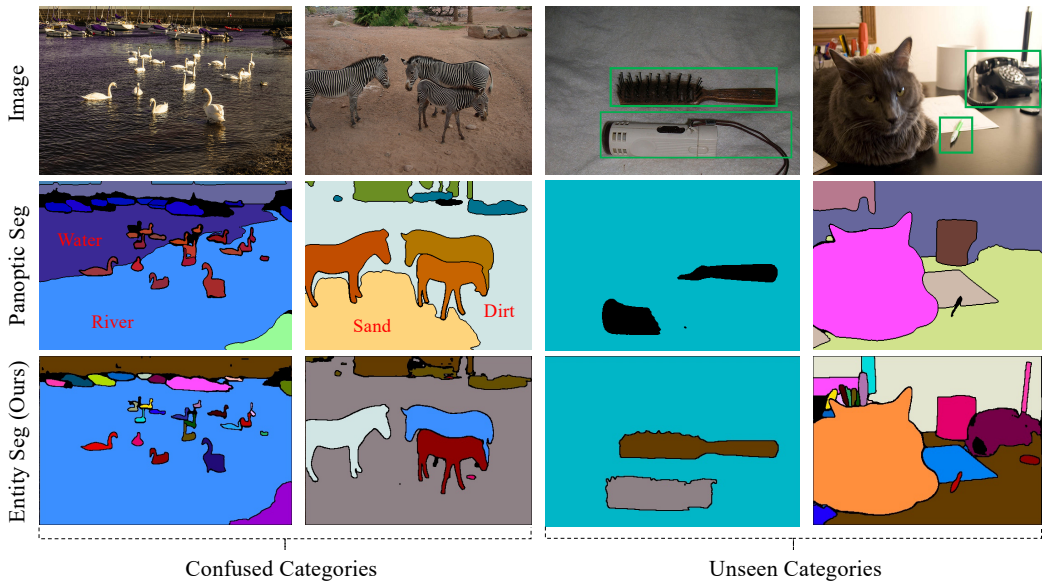


Figure 1: The first two columns show the examples of category confusion. Even with a state-of-the-art panoptic segmentation network[1], there tends to be two or more masks with semantically-overlapped categories (*e.g., river* & *water*, *sand* & *dirt*) predicted for a single entity. The last two second columns illustrate that the network has troubles segmenting unseen objects like *afro pick*, *pencil*, *telephone*.

issue can be largely alleviated. Besides, as demonstrated in object detection [47, 48], category agnosticism provides a strong generalization to unseen categories. Furthermore, this variant of image segmentation relieves the model's burden of trading off segmentation mask quality against categorization performance.

To this end, we propose a new image segmentation task named Entity Segmentation (ES) which aims to generate category-agnostic segmentation masks of an image, we leverage the existing panoptic segmentation datasets [49–51] with both instance (*thing*) and non-instance (*stuff*) masks, but universally treat any of the annotated *thing* and *stuff* masks as a categoryless ***entity***. Since there is no separation of *thing* and *stuff*, the widely used Panoptic Quality (PQ) metric [17] is not applicable to the new task. Therefore, we propose a new average precision (AP_e) evaluation metric that considers all masks as independent entities, instead of treating them differently based on whether they belong to *thing* or *stuff*. Furthermore, unlike instance segmentation AP metrics [49], we incorporate a strict constraint to the evaluation metric for it to accept only non-overlapping segmentation masks. This constraint aligns well with image manipulation and editing applications in which each pixel of an image should only belong to a single entity.

Based on our proposed new task and evaluation metric, we introduce a simple but effective framework to tackle the task. Unlike existing panoptic segmentation methods [17–20] that generally include two output branches for *thing* and *stuff* separately, the proposed framework relies on a dense one-stage detection architecture and a dynamic mask kernel branch to detect and segment all entities (regardless of *thing* or *stuff*) in a unified manner. Besides, we propose a global kernel bank module and an overlap suppression module to further improve the model's performance on the task. The global kernel bank module generates mask kernels to exploit some common properties (*e.g,* textures, edges) shared by many entities and the overlap suppression module encourages the masks not to overlap with each other. Extensive experiments on the challenging COCO dataset [49] show the effectiveness of our proposed method.

Finally, we study the generalization ability of our COCO-trained model through a cross-dataset evaluation involving ADE20K [50] and Cityscapes [51]. Our model shows superior performance both quantitatively and qualitatively on such datasets despite that is has not been trained on them. As shown in Fig. 4, the model is able to correctly segment the entities belonging to unseen categories.

[1]PanopticFCN [20] with ResNet101 [32] backbone and Deformable Convolution v2 [52, 53].

These suggest that the models trained for the proposed ES task are naturally capable of open-world image segmentation.

Overall, our contributions are summarized as follows: (1) We propose a new task called entity segmentation, which aims to segment every entity without predicting its semantic category. By removing the burden from semantic categorization, this task puts a strong emphasis to the mask quality of the *entity* itself and it inherently enables models to generalize well to the open world scenarios and domains. (2) We surprisingly find that any entity (*thing* or *stuff*) can be very effectively and uniformly represented by *center points* in the network. This unified representation provides the basis to our proposed segmentation framework for this new task. And two novel modules integrated to the framework to further improve the performance on this task. (3) The extensive experiments show the remarkable effectiveness and generalization of our proposed method for entity segmentation.

2 Related Work

Image segmentation. Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects) [54]. The main tasks of image segmentation include salient object detection [55–59], semantic segmentation [1–7, 60–64], instance segmentation [8–16], and panoptic segmentation [17–21, 65]. Salient object detection mimics the behavior of human in focusing on the most salient region/object in an image while ignoring the category it belongs to [66]. In contrast, semantic/instance/panoptic segmentation aims at densely assigning each image pixel to the one of categories predefined in the training datasets. The segmentation models trained for these tasks are required to make a trade-off between the mask prediction and categorization subtasks in terms of performance. Moreover, such tasks tightly couples segmentation and classification. As a result, it is not easy to independently evaluate the segmentation and classification strengths of their segmentation models.

In this work, we provide a new perspective on image segmentation by introducing the entity segmentation task that handles dense image segmentation like semantic/instance/panoptic segmentation, but without the categorization aspect akin to salient object detection. This task focuses only on category-agnostic segmentation. It treats every segment (whether *thing* or *stuff*) as a visual entity. Compared to the existing segmentation tasks, it is more useful for user-friendly image manipulation and editing applications in which the segmentation mask quality is of utmost importance.

Object detection. Object detection [67, 68, 45, 69–73] requires detecting objects with bounding box representation. The methods could be thoroughly distinguished into two types: (1) one-stage [70, 71] detectors that detect objects using pixelwise features at dense pixel locations; (2) two-stage detectors [67, 68, 45, 69] that first predict category-agnostic region proposals and subsequently perform categorization based on region-of-interest (RoI) features and a separate classification head.

In this paper, our method is built upon FCOS [71], a standard one-stage detector, to detect entities. Different from the standard edition of *object*, entity is more general concept that includes both objects/*things* and *stuffs*. To our best knowledge, our method is the first to demonstrate that center-based detection can effectively handle all entities in a uniform manner, contrary to the existing practice of separately handling *things* and *stuffs*. In contrast to DETR [21], our framework neither requires an excessively long training duration (12 *vs* 300 epochs) nor strong data augmentation.

Open world. The open-world setting has been explored in the context of object detection or segmentation [74–81] with the general goal of identifying new object categories given a closed-world training dataset. Generally, previous open-world methods explicitly differentiate *unknown* and *known* objects by spotting outliers in the embedding space. On the contrary, our entity segmentation task is category-agnostic and does not need to distinguish between *unknown* and *known* categories, allowing the segmentation models to focus entirely on the segmentation task itself.

3 Entity Segmentation

Task definition. The task of entity segmentation (ES) is defined as segmenting all visual entities within an image in a category-agnostic manner. Here, the "entity" refers to either a *thing* (instance) mask or a stuff mask in the common context. As illustrated in Fig. 2, the entity can be any semantically

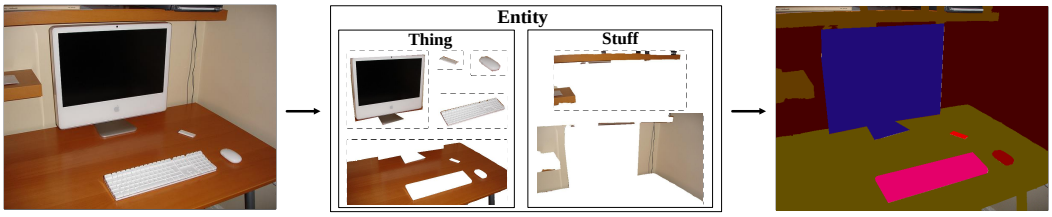


Figure 2: The ground-truth annotations of entity segmentation. To convert the annotations of panoptic segmentation to entity segmentation format, we regard each *thing* or *stuff* as an independent entity, even though some of them may have multiple disconnected parts.

meaningful and coherent region in the open-world setting, *e.g.,* person, television, wall. Given the subjective nature of this task, we conduct a comprehensive user study to validate the personal intuitions for the concept of entity in *supplementary file*.

Task format. Given an input image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, the task expects a pixel-wise prediction map $\mathbf{P} \in \{0, 1, ..., N\}^{H \times W}$ and a list of confidence scores $\mathbf{S} \in \mathbb{R}^{N+1}$ as the output that contains the non-overlapping IDs of the predicted N entities and their confidence scores. Ground truth annotations are encoded in an identical manner as the prediction map. There are no semantic category labels involved and all entities are treated equally without the distinction of *thing* and *stuff*.

Relationship to similar tasks. The newly proposed task is focused on the concept of "entity" itself. It is related to but different from several previous tasks. Different from *semantic segmentation*, ES is instance-aware. In contrast to *instance segmentation*, ES includes *stuff* masks in addition to instance masks. Moreover, unlike the above-mentioned segmentation tasks and the more recent *panoptic segmentation*, ES completely omits the category labels and categorization functionality.

Annotation transformation. Given the commonalities with panoptic segmentation, the annotations in existing panoptic segmentation datasets can be directly transformed into the format defined for ES. As shown in Fig. 2, each *thing* or *stuff* mask is simply regarded as an independent entity.

Evaluation metric. Due to the non-overlapping requirement in downstream image manipulation and editing applications, we propose a new mask-based mean average precision for measurement, denoted as AP_e^m. AP_e^m follows closely the AP^m used in instance segmentation [8–16], except that the AP_e^m gives zero tolerance to overlapped masks of different entities. This simple constraint leads to a lower evaluation number than instance segmentation's AP^m, since it makes the metric significantly more sensitive to the category-agnostic duplicate removal [82] performance and mask quality. As shown in Table 2(a), the number drops by 4.7 if AP_e^m is used in place of AP^m.

4 Unified Entity Representation

The output representation is a defining aspect of any segmentation tasks. Given the close relationship of ES with panoptic segmentation, we first discuss panoptic representation. Then, we introduce our unified entity representation and explain the motivation behind this new representation in addition to our findings that provide support to such a representation.

Due to the different natures of *thing* (instance) and *stuff* (non-instance) categories, current panoptic segmentation methods usually adopt separate strategies [18, 19] for the two outputs. In particular, they are handled through a divide-and-conquer pipeline, *i.e.,* using a branch that emphasizes explicit localization cues for things, while using another branch that focuses on semantic consistency for *stuff* [20]. However, due to the lack of both category labels and *thing-stuff* separation in our ES task, the existing panoptic representation is incompatible with ES and thus existing panoptic approaches are not directly applicable to ES.

As with panoptic segmentation, the question of "how to represent entity?" is crucial to ES. Here, we assume that each entity can be effectively represented by its *center point*, as already been proven effective in earlier tasks: object detection [70, 71] and instance segmentation [13, 12, 16].

Subset	Category-agnostic		AP^b	AP^b_{50}	AP^b_{75}	AP^b_S	AP^b_M	AP^b_L
	training	*evaluation*						
Thing	○	○	37.4	56.0	39.9	15.5	36.5	50.5
	○	✓	40.9	63.9	43.4	18.6	42.1	60.7
	✓	✓	41.6	64.6	43.9	18.5	42.7	61.7
Stuff	○	○	23.2	35.4	23.5	1.4	5.9	26.9
	○	✓	38.7	58.4	39.3	1.8	7.2	46.6
	✓	✓	39.4	60.5	39.3	1.4	6.4	47.2
Thing & Stuff (Entity)	○	○	29.5	45.2	31.0	9.2	23.6	38.0
	○	✓	37.6	60.2	39.3	16.5	35.0	49.5
	✓	✓	39.2	62.5	40.4	16.6	35.5	51.1

Table 1: Ablation study on detection performance of different subsets we use from COCO. The "thing" means we train and test our model only by the annotations of thing. ○/✓ signifies that we do/do-not use category information in the training or evaluation stage. AP^b is the AP for bounding boxes.

The effectiveness of such representation on our new ES task is empirically validated through an experiment on the proxy task of *entity detection*[2], as shown in Table 1.

It can be clearly seen that the center-based representation can effectively represent most of the *things*, *stuffs*, and *entities* (*things* and *stuffs*) in both category-oriented and -agnostic manners, except for the category-oriented *stuffs* which suffers from a much weaker performance. Here, the representation is considered effective for a particular setting (a single row in the table) if it achieves better or comparable results, as against the category-oriented *thing* detection's results shown in the first row of **Thing**. Furthermore, we find that training on only category-agnostic data provides a better category-agnostic evaluation result, compared to category-oriented training, as shown in the second and third rows of **Thing/Stuff/Entity**. These two interesting findings provide strong support as to why our category-agnostic unified representation is the right representation choice for ES, but not necessarily the case for existing category-oriented tasks such as panoptic segmentation.

5 Method

As explained in Sec. 4, all entities are detected based on the unified entity representation through box prediction, similar to that in [21]. We adapt the FCOS object detection method for detecting entities, representing each entity by its *center point*. In this section, we first describe how we build on top of the object detector to develop a complete *segmentation framework* tailored for center-based unified entity representation. Then, we introduce two modules to improve the segmentation quality of predicted entity masks: *global kernel bank* and *overlap suppression*.

5.1 Segmentation Framework

As described in Sec. 3, a non-overlapping entity ID prediction map $\mathbf{P}$ and a list of confidence score map $\mathbf{S}$ are expected, given an input $\mathbf{I}$. Inspired by recent one-stage segmentation approaches [13, 15, 20, 16], we adopt a similar *generate-kernel-then-segment* pipeline. Specifically, with a single stage feature $\mathbf{X}_i \in \mathbb{R}^{H_i \times W_i \times C_i}$ from the i-th stage in Feature Pyramid Network (FPN) [69], four branches are added to handle the four output types required to perform ES: *entityness*, *centerness*, *localization*, and *kernel*. Here, entityness and centerness branches respectively provide a probability map $\mathbf{E}_i \in [0,1]^{H_i \times W_i}$ and a centerness map $\mathbf{C}_i \in [0,1]^{H_i \times W_i}$. $\mathbf{E}_i$ indicates each pixel's probability of being an entity while $\mathbf{C}_i$ estimates centerness values [71]. Localization branch is used to regress the bounding box offsets $\mathbf{L}_i$ of entities, which are used for efficient non-maximium suppression (NMS). For mask generation, we draw inspirations from dynamic kernel work [83, 13, 20, 16] and employ a kernel branch to generate entity-specific dynamic kernel weights.

[2]We use FCOS [71] with ResNet50 backbone following its default hyper-parameter setting. The bounding box and segmentation mask could be regarded as representations of entity from coarse to fine. There is slight possibility for a representation to perform well in mask but bad in box.

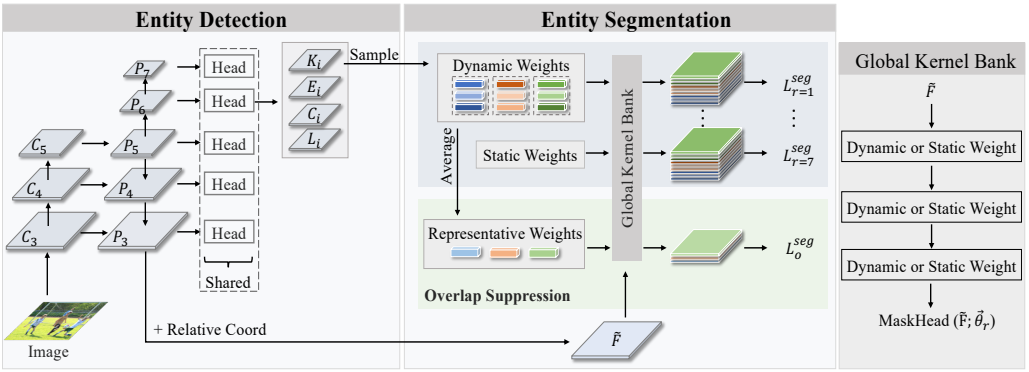


Figure 3: The entity segmentation framework. L_o^{seg} here stands for $L_{o,r=7}^{seg}$ which indicates that the global kernel bank's path with entirely dynamic weights is used for the overlap suppresion module.

5.2 Global Kernel Bank

The use of dynamic kernels to generate entity masks assumes that the cues needed to segment different entities are dissimilar, but there exists several common properties (*e.g.* textures, edges) which are shared by different entities. This motivates us to leverage static kernel weights for training the mask head, alongside the dynamic kernel weights. The combination of dynamic and static kernel weights, referred to as *global kernel bank*, allows the network to strike a good balance between learning entity-specific features and becoming aware of features that are commonly useful to many entities.

Global kernel bank consisting of three pairs of dynamic and static convolutions can be easily incorporated into our segmentation framework. To strengthen the interactions between dynamic and static kernels, we consider either dynamic or static kernels at each convolutional layer, resulting in seven possible network paths (minus the one with solely static kernels) in a 3-layer mask head, as presented in Fig. 3. During training, we train with all seven paths simultaneously to minimize the mask prediction (Dice [84]) loss with respect to ground truth $\mathbf{Y}$:

$$\mathcal{L}_r^{seg} = \lambda_r \times \text{Dice}(\text{Sigmoid}(\text{MaskHead}(\tilde{\mathbf{F}}; \vec{\theta}_r)), \mathbf{Y}), \tag{1}$$

where r and λ_r respectively denote the path's index and the path-specific hyperparameter that weights the loss. $\vec{\theta}_r$ refers to the kernel weights contained in the r-th path and $\tilde{\mathbf{F}} \in R^{H/8 \times W/8 \times (C_{mask}+2)}$ indicates the concatenation of backbone features (encoded for mask prediction) and relative coordinates [85]. Despite the many paths trained, during inference, we find that using the path with solely dynamic kernels alone for mask prediction is sufficiently effective while being efficient.

5.3 Overlap Suppression

The ES task (Sec. 2) expects no overlapping masks. However, dynamic kernel-basel approaches tend to generate overlapping masks with high confidence due to the concept overlap among adjacent entities and independent losses being used for different kernels. Although postprocessing strategies [18, 20] can resolve mask overlaps, they are driven by heuristics which are less effective. Instead, we propose a method that encourages the suppression of overlaps among the predicted masks of entities.

During training, there are M number of sampled kernels and each of the M kernels is assigned to one of the N ground-truth entities according the mask-based sample assignment strategy [13]. This can be viewed as having N clusters with one or more kernels within each cluster. We represent the *representative* kernel in each cluster by its cluster mean:

$$\vec{\theta}_o^n = \text{Average}(K_n) = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M} \mathbb{1}^{m \in n} K_n}{\sum_{n=1}^{N}\sum_{m=1}^{M} \mathbb{1}^{m \in n}} \tag{2}$$

where $\mathbb{1}^{m \in n}$ is the indicator function that indicates whether the m-th kernel weight is assigned to the n-th entity. Given the representative kernels, we generate the representative entity masks and apply a softmax function to induce a strong suppression of non-maximal entities in the pixel-wise mask

Model	PQ	AP^m	AP^m_e
PanopticFPN [18]	39.4	-	23.2
PanopticFCN [20]	41.1	-	24.5
DETR [21]	43.4	-	24.8
Ours	-	34.6	29.8

(a)

$\mathcal{L}^{seg}_R$	$\mathcal{L}^{seg}_o$	AP^m_e
○	○	28.3
✓	○	29.1
○	✓	29.3
✓	✓	29.8

(b)

Softmax	Sigmoid	AP^m_e
○	○	28.3
○	✓	28.6
✓	○	29.1
✓	✓	29.0

(c)

Table 2: **(a)**: **Comparison with the existing panoptic segmentation methods**. For the existing panoptic segmentation methods, we merely convert their panoptic results to the ES format and obtain the entity scores for "*stuff*" entities by averaging the scores within each *stuff* mask. PQ is the evaluation metric for conventional category-aware panoptic segmentation. AP^m is the overlap-tolerated entity segmentation evaluation metric similar to common instance segmentation metrics. **(b)**: **Proposed modules**. The effect of global kernel bank and overlap suppression module. **(c)**: **Overlap suppression**. The ablation study of scoring activation function in the module.

prediction. For overlap suppression, we adopt a separate training loss similar to $\mathcal{L}^{seg}_r$:

$$\mathcal{L}^{seg}_o = \text{Dice}(\text{Softmax}(\text{MaskHead}(\tilde{\mathbf{F}}; \vec{\theta}_o)), \mathbf{Y}), \tag{3}$$

Note that the area without any annotation is ignored in L^{seg}_o.

5.4 Training and Inference

In the training stage, the segmentation network is trained with the overall loss defined as:

$$\mathcal{L} = \mathcal{L}^{det} + \mathcal{L}^{seg}_o + \mathcal{L}^{seg}_R = \mathcal{L}^{det} + \mathcal{L}^{seg}_o + \sum_r \mathcal{L}^{seg}_r. \tag{4}$$

In the inference stage, we first sort all entity detections according to the (aggregated) confidence score: $\sqrt{\mathbf{C}_i(\text{centerness})} \times \mathbf{E}_i(\text{entityness probability})$, and their corresponding boxes are obtained from the localization branch for box-level NMS. After the duplicate removal, each remaining entity is encoded into an entity-specific dynamic kernel, resulting in N kernels for N entities. With the generated dynamic kernels and encoded features shown in Fig. 3, the segmentation masks of N entities are produced by a sequence of convolutions directly. At last, the final non-overlapping prediction map $\mathbf{P}$ is acquired by choosing the entity ID with the maximum confidence score at each pixel [18].

6 Experiments

6.1 Main Evaluation

We perform the main evaluation studies in the single-dataset setting on COCO dataset [49]. Following common practice, we train our models with 115,000 train images and reported results on the 5,000 validation images. Without specification, we train our network with ImageNet-pretrained ResNet-50 [32] backbone using batch size 16 for 12 epochs. The longer edge sizes of the images is 1333. The shorter edge sizes of the images is random sampled from 640 to 800 with stride 32. We decay the learning rate with 0.1 after 8 and 11 epochs respectively.

Comparison with the existing panoptic segmentation methods. In Table 2(a), we compare: (1) our proposed evaluation metric AP^m_e with existing AP^m [49] and PQ [17] metrics; (2) our segmentation method with PanopticFPN [18], and recently-proposed DETR [21] and PanopticFCN [20]. We find that AP^m_e results in smaller numbers than the standard AP^m, due to the non-overlapping constraint in AP^m_e that makes the evaluation stricter than the overlap-tolerant AP^m. Furthermore, with the non-overlapping constraint, the performance becomes more sensitive to the overlap-resolving strategy in use. This is one of the reasons that explain the much worse AP^m_e performance of existing panoptic methods that do not take the constraint into account during training. Especially, our proposed method obtains better AP^m_e than DETR, which also unified thing and stuff. This manifests convolution network is capable of representing thing and stuff in a unified fashion as well as popular transformer, which is more complicated and time-consuming. The numbers of AP^m_e are also smaller than PQ's.

111	110	101	100	011	010	001	AP^m_e	$AP^m_{e_{50}}$	$AP^m_{e_{75}}$
1.0	0.0	0.0	0.0	0.0	0.0	0.0	28.3	48.7	28.8
2.0	0.0	0.0	0.0	0.0	0.0	0.0	28.1	48.4	28.5
1.0	1.0	0.0	0.0	0.0	0.0	0.0	28.9	49.3	29.1
1.0	1.0	1.0	0.0	0.0	0.0	0.0	29.1	49.8	29.6
1.0	1.0	1.0	1.0	1.0	1.0	1.0	27.8	47.8	27.7
1.0	1.0	1.0	0.25	0.25	0.25	0.25	29.3	50.2	29.8

(a)

MODEL	AP^m_e	$AP^m_{e_{50}}$	$AP^m_{e_{75}}$
Baseline	29.8	50.3	30.9
R-50	31.8	53.5	33.8
R-50-DCNv2	33.7	56.1	35.6
R-101	33.2	55.5	34.8
R-101-DCNv2	35.5	58.2	37.1
Swin-L	38.6	62.4	40.8

(b)

Table 3: Ablation studies. **(a)**: **Global Kernel Bank**. The r in $\mathcal{L}^{seg}_R$ ranges from 1 to 7. In the first row, "xxx" is binary representation of r. For example, "100" corresponds to the 4-th path with the first layer (1) and last two (00) layers using dynamic and static weights, respectively.Each entry below path IDs indicates the loss weight λ_r. **(b)**: **High-Performance Regime**. The performance of our models enhanced by stronger backbones and a longer training duration. "Swin-L" and "DCNv2" refer to Swin Transformer [86] in large series with window size 7 and deformable Convolution v2 [53].

This is because PQ merely uses a single IoU threshold of 0.5 to determine true positives, while AP^m_e considers a range of IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05.

Proposed modules. Table 2(b) summarizes the performance improvement introduced by the two proposed modules to the baseline. The baseline is a simple segmentation framework that uses only the loss $\mathcal{L}^{seg}_0$ in mask branch, and it obtains 28.5 AP^m_e. The incorporation of $\mathcal{L}^{seg}_R$ and $\mathcal{L}^{seg}_o$ improves it by 0.6, 0.8 AP^m_e, respectively, demonstrating the effectiveness of the proposed modules. The last line shows that using these two modules achieves an even greater improvement of 1.4 AP^m_e. This indicates that the two proposed modules are complementary with each other.

Overlap suppression. Table 2(c) provides the ablation study on the activation function used in $\mathcal{L}^{seg}_o$. Compared to the baseline without any score function, the Sigmoid and Softmax functions respectively bring 0.1 and 0.6 AP^m_e improvements. Compared to the Sigmoid function, Softmax activation forces the network to suppress the mask prediction of non-maximal entities at every pixel and it helps to suppress mask overlaps more effectively. In the last row of Table 2(c), we use both Sigmoid and Softmax functions with the loss weight of 0.5 for each.

Global kernel bank. Table 3(a) presents the ablation study on the choice of the mask head paths used in the training stage. The second row is dynamic-kernel baseline that obtains 28.5 AP^m_e, using only the dynamic kernel weights for all layers. Increasing the loss weight of this path does not further improve the performance. Simply using static weights in the last one or two layers improves the AP^m_e by 0.4 and 0.6. The best performance is achieved from using moderate loss weights for the paths with static kernel weights. This allows the dynamic kernel weights to be more aware of common property of entities like textures or edges, while still preserving a decent entity discriminability.

High-performance regime. Table 3(b) shows the performance of our proposed method in the high-performance training regime: stronger backbones and a longer training duration. With the longer training duration, we train the networks using batch size 16 for 36 epochs. We decay the learning rate with 0.1 factor at the 33-th and 35-th epochs. The models trained with our method benefit from the various techniques of high-performance training regime. The strongest one with ResNeSt-101 and Deformable Convolution v2 obtains 36.2 AP^m_e and it is used for the cross-dataset visualization in the next section.

6.2 Cross-Dataset Evaluation

To demonstrate the generalization advantage of entity segmentation task, we consider the setting where the evaluation set comes from a dataset different than the one/ones used for training. We experiment with COCO, ADE20K, and Cityscapes. Table 4(b) shows the statistics of these datasets. All these three datasets are converted into the ES format. For fair comparison, all models are trained with pre-sampled 141,944 images and 12 epochs regardless which datasets are used. Please refer to the supplementary file for more details.

Dataset	COCO	ADE20K	City	CO+A	CO+A+CI
COCO	29.9	19.4	4.2	30.6	30.6
ADE20K	20.1	24.1	3.7	26.3	26.4
City	25.4	16.4	29.5	22.0	26.2

(a)

Dataset	categories	Train Images	Val Images
COCO	133	118,287	5,000
ADE20K	150	20,210	2,000
Cityscapes	19	2,975	500

(b)

Table 4: Cross-dataset evaluation. **(a)**: **Quantitative performance**. Each column and row represent the training and validation dataset we used, respectively. The "CO+A" means we use both COCO and ADE20K training dataset. It is similar to "COCO+A+CI". **(b)**: **Statistics of the datasets**.

Figure 4: The illustration of our model's generalization ability. We train the model on COCO dataset but apply it to ImageNet images. Note that most segmented entities are outside of COCO categories.

Quantitative evaluation. Table 4(a) shows the evaluation results in the cross-dataset setting. The model trained on only COCO dataset obtains 30.3, 20.1 and 25.4 AP_e^m on COCO, ADE20K and Cityscapes validation sets, signifying the superior generalization advantages of our task and proposed segmentation framework. The performance on ADE20K than that on COCO by 10.2 AP_e^m. There are two reasons: (1) certain categories common to COCO and ADE20K are distinguished as *thing* and *stuff* differently; (2) ADE20K has more noisy annotations. As a result, directly using ADE20K training set only obtains the 24.1 AP_e^m on ADE20K validation split. Despite that, the qualitative results of ADE20K that we include in the supplementary file indicate that our model achieves a reasonably good performance. Furthermore, there is a big performance gap on COCO validation step between using COCO and ADE20K for training. This is caused by ADE20K's low number (20,210) of training samples that are insufficient to train the network well. One interesting aspect of our proposed task and framework is that multiple datasets of different domains can be directly combined to form a large dataset for training. When COCO, ADE20K and Cityscapes are combined for training, the model achieves the best overall performance on all datasets.

Qualitative results. Fig. 4 shows the our model's qualitative results on ImageNet. The model used here is based on the R-101-DCNV2 backbone described in Table 2(b) and trained only on COCO dataset. Please refer to the supplementary file for many more cross-dataset qualitative results on ADE20K, Cityscapes, Places2 [87], and Objects365 [88].

7 Conclusion

This paper proposes a new task named entity segmentation (ES) which is aimed to serve downstream applications such as image manipulation/editing that have high requirements for segmentation mask quality but not for category labels. In this regard, an entity is defined as any semantically-meaningful and -coherent segment in the image. In the absence of semantic labels, we design a new metric mAP_e to measure the ES performance. To effectively represent entities, we introduce a center-based representation that can handle both *thing* and *stuff* in this task by assigning one entity ID to each pixel. The proposed global kernel bank and overlap suppression modules further improve the segmentation performance. Experiments show that the models trained for ES task demonstrate strong generalization strengths and they perform exceptionally well in the open-world setting.

Broader impact. This work has the potential to create a positive impact to the community through the lens of data bias. One problem with the widely-used public segmentation datasets is that they tend to suffer from severe geographical biases because the data annotation efforts have been historically led by the researchers and organizations from wealthy countries who can afford the hefty annotation costs. The models trained for our entity segmentation task have superior generalization that makes it possible to segment unseen entities images from under-represented geographical regions. Thus, it may help to mitigate the category bias issues present in existing datasets. Additionally, our proposed task and method enable the straightforward merging of multiple training datasets which helps to prevent the trained model from being biased towards any specific dataset. Despite the potential bias-related benefits, the models trained with our approach are inevitably affected by such dataset biases in some way, but our approach does not introduce additional biases, as far as we are aware. It may not necessarily count as a negative impact, but the resulting segmentation models from our task cannot be used for certain life-critical applications, such as the ones for visually-impaired people and autonomous driving that heavily depend on semantic label predictions for navigation, taking responsive actions, *etc.* Such a limitation can be addressed by augmenting our segmentation network with a classification branch or a separate classification network, which we leave for future work.

Acknowledgement. We would like to thank Yanwei Li (the author of PanopticFCN [20]), Jingbo Wang, Yukang Chen, and Wenbo Li for deep discussions.

References

[1] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[2] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv:1706.05587*, 2017.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[5] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*, 2018.

[6] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019.

[7] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *CVPR*, 2019.

[8] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, 2016.

[9] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[11] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, 2018.

[12] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *CVPR*, 2020.

[13] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020.

[14] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, 2020.

[15] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *NeurIPS*, 2020.

[16] Lu Qi, Yi Wang, Yukang Chen, Yingcong Chen, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Pointins: Point-based instance segmentation. *TPAMI*, 2021.

[17] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019.

[18] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019.

[19] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019.

[20] Yanwei Li, Hengshuang Zhao, Xiaojuan Qi, Liwei Wang, Zeming Li, Jian Sun, and Jiaya Jia. Fully convolutional networks for panoptic segmentation. In *CVPR*, 2021.

[21] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[22] Guang Shu. Human detection, tracking and segmentation in surveillance video. 2014.

[23] Douglas Morrison, Adam 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 *ICRA*, 2018.

[24] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *CVPR*, 2019.

[25] Yu Rong, Takaaki Shiratori, and Hanbyul Joo. Frankmocap: Fast monocular 3d hand and body motion capture by regression and integration. *arXiv preprint arXiv:2008.08324*, 2020.

[26] Kaidi Cao, Yu Rong, Cheng Li, Xiaoou Tang, and Chen Change Loy. Pose-robust face recognition via deep residual equivariant mapping. In *CVPR*, 2018.

[27] Yu Rong, Ziwei Liu, Cheng Li, Kaidi Cao, and Chen Change Loy. Delving deep into hybrid annotations for 3d human recovery in the wild. In *ICCV*, 2019.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[31] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *arXiv*, 2016.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[33] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[34] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[35] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. *TOG*, 2005.

[36] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *NeurIPS*, 2018.

[37] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[38] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *arXiv:1711.11585*, 2017.

[39] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018.

[40] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019.

[41] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *TOG*, 2009.

[42] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image melding: Combining inconsistent images using patch-based synthesis. *TOG*, 2012.

[43] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. *ACM SIGGRAPH*, 2004.

[44] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM SIGGRAPH*, 2003.

[45] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NeurIPS*, 2015.

[46] Infant Cognition Laboratory (UC Davis). Infant Categorization Development. https://oakeslab.ucdavis.edu/infant-categorization-development.html.

[47] Ayush Jaiswal, Yue Wu, Pradeep Natarajan, and Premkumar Natarajan. Class-agnostic object detection. In *WACV*, 2021.

[48] Shafin Rahman, Salman Khan, and Fatih Porikli. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In *ACCV*, 2018.

[49] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[50] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

[51] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[52] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017.

[53] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *CVPR*, 2019.

[54] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[55] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *TIP*, 2015.

[56] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational visual media*, 2019.

[57] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. Salient object detection: A discriminative regional feature integration approach. In *CVPR*, 2013.

[58] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *CVPR*, 2019.

[59] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *CVPR*, 2016.

[60] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, 2018.

[61] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *CVPR*, 2018.

[62] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *CVPR*, 2020.

[63] Xiaokang Chen, Kwan-Yee Lin, Jingbo Wang, Wayne Wu, Chen Qian, Hongsheng Li, and Gang Zeng. Bi-directional cross-modality feature propagation with separation-and-aggregation gate for rgb-d semantic segmentation. In *ECCV*, 2020.

[64] Yajie Xing, Jingbo Wang, and Gang Zeng. Malleable 2.5 d convolution: Learning receptive fields along the depth-axis for rgb-d scene parsing. In *ECCV*, 2020.

[65] Huanyu Liu, Chao Peng, Changqian Yu, Jingbo Wang, Xu Liu, Gang Yu, and Wei Jiang. An end-to-end network for panoptic segmentation. In *CVPR*, 2019.

[66] Inam Ullah, Muwei Jian, Sumaira Hussain, Jie Guo, Hui Yu, Xing Wang, and Yilong Yin. A brief survey of visual saliency detection. *Multimedia Tools and Applications*, 2020.

[67] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[68] Piotr Dollár and C Lawrence Zitnick. Fast edge detection using structured forests. In *PAMI*, 2015.

[69] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[70] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[71] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.

[72] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. In *NeurIPS*, 2016.

[73] Lu Qi, Jason Kuen, Jiuxiang Gu, Zhe Lin, Yi Wang, Yukang Chen, Yanwei Li, and Jiaya Jia. Multi-scale aligned distillation for low-resolution detection. In *CVPR*, 2021.

[74] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. *arXiv preprint arXiv:2104.04691*, 2021.

[75] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *CVPR*, 2015.

[76] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019.

[77] Pedro O. Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *NeurIPS*, 2015.

[78] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick. Learning to segment every thing. In *CVPR*, 2018.

[79] Ayush Jaiswal, Yue Wu, Pradeep Natarajan, and Premkumar Natarajan. Class-agnostic object detection. In *WACV*, 2021.

[80] A. R. Dhamija, M. Günther, J. Ventura, and T. E. Boult. The overlooked elephant of object detection: Open set. In *WACV*, 2020.

[81] K J Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, 2021.

[82] Lu Qi, Shu Liu, Jianping Shi, and Jiaya Jia. Sequential context encoding for duplicate removal. In *Advances in Neural Information Processing Systems*, 2018.

[83] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *NeurIPS*, 2016.

[84] F Milletari, N Navab, SAV Ahmadi, and V-net. Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016.

[85] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *NeurIPS*, 2018.

[86] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[87] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

[88] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8430–8439, 2019.

[89] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *arXiv preprint arXiv:1705.02999*, 2017.

Open-World Entity Segmentation: Supplementary Material

Lu Qi[*1] **Jason Kuen**[*2] **Yi Wang**[1] **Jiuxiang Gu**[2]
Hengshuang Zhao[3] **Zhe Lin**[2] **Philip Torr**[3] **Jiaya Jia**[1,4]
[1]The Chinese University of Hong Kong [2]Adobe Research
[3] University of Oxford [4]SmartMore

In this supplementary file, we provide more empirical results to further demonstrate the benefits of our proposed entity segmentation task, method, in addition the applications enabled by our task and method, as listed in the following:

- User study on the definition of entity.
- Details of cross-dataset training.
- Visualization results and survey study for comparison between PanopticFCN [20] and *ours*.
- Additional visualization results on six datasets: COCO [49], ADE20K [50], CityScapes [51], Places2 [87], ImageNet [28], Object365 [88].
- Applications enabled by entity segmentation.

8 User study on the definition of entity

We conducted a user study in which there were 480 individuals who were identified as Adobe Photoshop users who regularly used the software for image manipulation/editing. In this user study, we randomly selected 40 images of COCO dataset and provided the users a visualization of the *entity* ground truths drawn with distinct colors. For each image, we asked each user about his/her degree of satisfaction about treating semantically-meaningful and -coherent segments as entities, with respect to their relevance to and suitability for image manipulation/editing applications. The satisfactory scores aggregated from all users for the individual images are given in Table 5. We find that the average score of each image is large than 7.8 on the condition that the maximum score is 10. Most of the selected images' scores are larger than 6.0, This confirms that the users are highly satisfied with our task's definition of *entity* (and the lacks of category labels and *thing-stuff* separation) in the context of image manipulation/editing. To better present the user study's findings, we summarize the data of Table 5 in Fig. 5(a) & (b). The image IDs with the minimum, median and maximum user scores are 20, 19, and 35. We show these images and their corresponding entity annotations in Fig. 6.

9 Details of cross-dataset training

As mentioned in the main paper, all models are trained in COCO setting with presampled 118,287 images and 12 epochs, regardless of the dataset or dataset combination we use. Given that the number of training images in ADE20K [50] and CityScapes [51] are 20,210 and 2,975, for each of these datasets, we accordingly adjust the number of training epochs and learning rate schedule to match the fixed number of presampled samples/images. We train ADE20K with 71 epochs ($\frac{118287\times12}{20210}$) and a learning rate of 0.1 decayed at the 48-th and 66-th epochs and CityScapes with 478 epochs ($\frac{118287\times12}{2975}$) and a learning rate of 0.1 decayed at the 318-th and 436-th epochs. The merged datasets: COCO+ADE20K or COCO+ADE20K+CityScapes follow similar settings. Other

[*]Equal Contribution

Preprint. Under review.

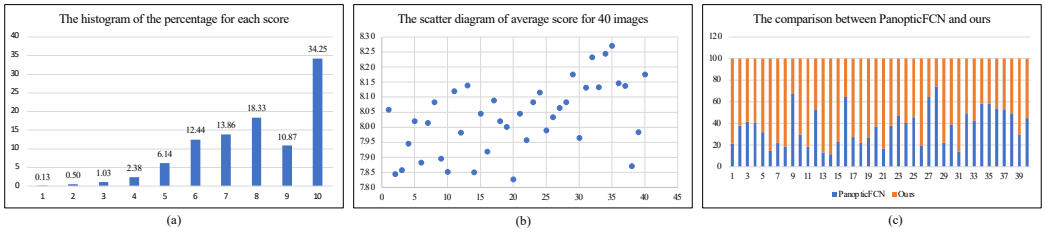


Figure 5: The statistical visualizations of the data from our user and survey studies. **(a)** The histogram here represents the distribution of the users based on their given scores in the entity definition user study. **(b)** The scatter diagram that showcases the averaged scores (averaged across all users) for each of the 40 images. **(c)** Each horizontal bar here indicates the proportions of votes given by the survey participants to PanopticFCN [20] (blue) and *ours* (orange) on each of the 40 images.

the above-mentioned, we strictly follow the standard COCO training and inference settings such as keeping the short and long sides of the input image to 800 and 1,333 pixels respectively.

10 Visualization results and survey study for comparison between PanopticFCN and *ours*

The segmentation results produced by a recent panoptic segmentation method and our entity segmentation framework are visually compared in Fig. 7 and 8. For panoptic segmentation, PanopticFCN with ResNet101 [32] backbone and Deformable Convolution v2 [53] is used as a strong contender for comparison here, as it achieves state-of-the-art performance among all publicly-released codes. Furthermore, we randomly selected 40 images (also included in Fig. 7 and 8) and conducted a survey study of 480 people about their preference for the segmentation results produced by PanopticFCN and *ours*. As shown in Fig. 5(c), the survey participants, to a great extent, preferred the segmentation results of our entity segmentation framework over PanopticFCN's.

11 Additional visualization results

We provide more visualization results of our entity segmentation approach on COCO (Fig. 9, 10, 11, and 12), ADE20K (Fig. 13 and 14), CityScapes (Fig. 15), Places2 (Fig. 16), ImageNet (Fig. 17, 18, 19, and 20), and Object365 (Fig. 21 and 22). **the model is only trained once on COCO dataset without additional fine-tuning on other datasets**.

Image ID	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	AVG
1	0.00	0.00	1.88	3.13	8.13	11.25	11.25	16.88	9.38	38.13	8.06
2	0.63	0.00	3.13	3.75	5.63	15.00	13.75	13.13	10.00	35.00	7.84
3	1.25	0.63	1.88	1.88	10.63	10.63	15.00	12.50	8.13	37.50	7.86
4	0.63	0.63	0.63	3.13	7.50	14.38	13.13	14.38	8.75	36.88	7.95
5	0.00	0.63	2.50	1.25	7.50	11.88	13.13	18.13	7.50	37.50	8.02
6	0.63	0.63	1.25	1.25	6.88	15.00	14.38	20.00	7.50	32.50	7.88
7	0.00	0.00	0.63	3.13	6.25	13.75	12.50	20.63	10.63	32.50	8.01
8	0.00	0.00	1.25	1.88	8.13	11.88	15.63	13.75	9.38	38.13	8.08
9	0.00	0.63	1.25	3.75	8.13	10.63	15.63	18.13	8.13	33.75	7.90
10	0.00	0.63	3.13	2.50	5.00	16.88	15.00	14.38	6.88	35.63	7.85
11	0.00	0.00	0.00	1.25	5.63	17.50	10.00	19.38	13.75	32.50	8.12
12	0.63	0.63	0.63	2.50	5.00	10.63	16.88	22.50	11.87	31.88	8.00
13	0.00	0.00	0.00	2.50	5.00	14.38	15.00	15.63	12.50	35.00	8.14
14	0.00	0.63	1.25	3.75	6.25	13.75	13.75	19.38	12.50	28.75	7.85
15	0.00	0.00	0.63	1.88	8.13	13.75	11.25	20.63	9.38	34.38	8.05
16	0.63	0.63	1.25	2.50	7.50	11.88	13.75	18.13	11.25	32.50	7.91
17	0.00	0.00	0.63	1.88	4.38	13.75	18.13	16.25	11.88	33.13	8.09
18	0.00	0.00	0.00	3.13	5.63	16.88	15.00	13.75	11.25	34.38	8.02
19	0.00	0.63	0.63	1.88	6.25	15.00	16.88	14.38	8.75	35.63	8.00
20	0.00	0.00	0.63	3.13	6.25	10.63	12.50	20.63	10.63	32.50	7.83
21	0.00	1.88	0.63	0.63	5.00	11.88	15.63	22.50	8.13	33.75	8.05
22	0.00	0.63	1.25	3.13	8.75	10.63	13.75	16.25	11.88	33.75	7.96
23	0.00	0.63	0.63	1.88	5.63	13.75	13.75	18.75	9.38	35.63	8.08
24	0.00	0.63	0.63	1.88	4.38	11.88	15.63	20.00	11.88	33.13	8.11
25	0.00	1.88	0.63	1.88	7.50	11.88	11.88	19.38	11.25	33.75	8.00
26	0.00	1.25	0.63	2.50	5.00	11.25	16.88	18.75	9.38	34.38	8.03
27	0.00	0.00	1.25	1.25	7.50	12.50	13.13	20.63	9.38	34.38	8.06
28	0.00	0.00	0.63	4.38	3.75	12.50	13.13	20.00	13.13	32.50	8.08
29	0.00	0.00	0.63	3.13	5.00	8.13	16.88	18.75	13.75	33.75	8.18
30	0.00	1.25	1.25	5.00	5.63	11.25	11.25	18.13	11.88	34.38	7.96
31	0.00	0.63	1.88	0.63	5.00	12.50	12.50	20.00	12.50	34.38	8.13
32	0.00	0.63	1.88	0.00	4.38	11.25	15.00	18.75	9.38	38.75	8.23
33	0.00	0.63	0.63	2.50	5.00	13.13	9.38	23.75	9.38	35.63	8.13
34	0.00	1.25	0.63	2.50	5.00	11.88	6.88	18.75	15.63	37.50	8.25
35	0.00	0.63	0.63	1.25	4.38	10.00	12.50	21.88	13.13	35.63	8.27
36	0.00	0.63	0.63	3.13	5.00	10.63	13.13	18.13	14.38	34.38	8.15
37	0.00	0.00	0.63	2.50	3.75	10.00	18.13	18.75	16.25	30.00	8.14
38	0.63	0.00	0.63	2.50	10.00	10.63	13.13	23.13	10.00	29.38	7.87
39	0.00	1.25	1.88	1.88	4.38	10.63	18.13	17.50	13.75	30.63	8.00
40	0.00	0.00	0.63	2.50	6.88	11.88	11.25	16.88	13.75	36.25	8.18

Table 5: Data from the user study on the definition of *entity*. P_x indicates the percentage of users who give "x" as the score to represent their degrees of satisfaction. "x" ranges from 1 to 10, and 10 represents the highest degree of satisfaction. "AVG" indicates the the score averaged across all users for each image.

12 Applications using entity segmentation

Given our entity segmentation results, we can perform image editing more conveniently, like how we obtain the bokeh and recoloring examples in Fig. 23. The visually convincing editing effects, as natural and delicate boundaries, in Fig. 23 validate that our mask quality is sufficiently accurate for these image manipulation tasks. With the segmentation maps from our model, we can obtain the foreground mask about the *object of interest* by selecting it with a simple click. Then bokeh effect can be rendered by blurring the background region. Similarly, we can recolor the *object of interest* by a deep colorization model [89]. Specifically, we recolor the whole picture first and then only update the color in the *region of interest* based on the foreground mask.

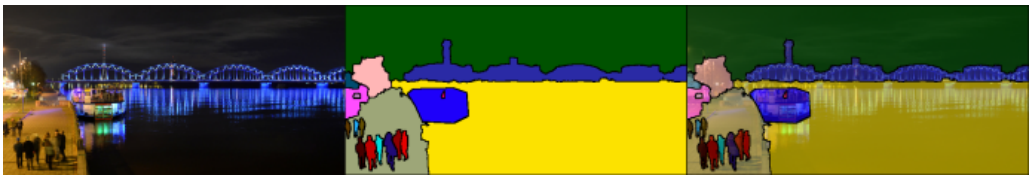


Image ID 20 with lowest user score

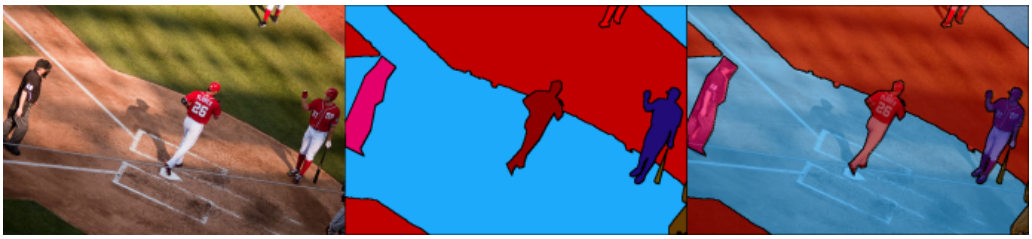


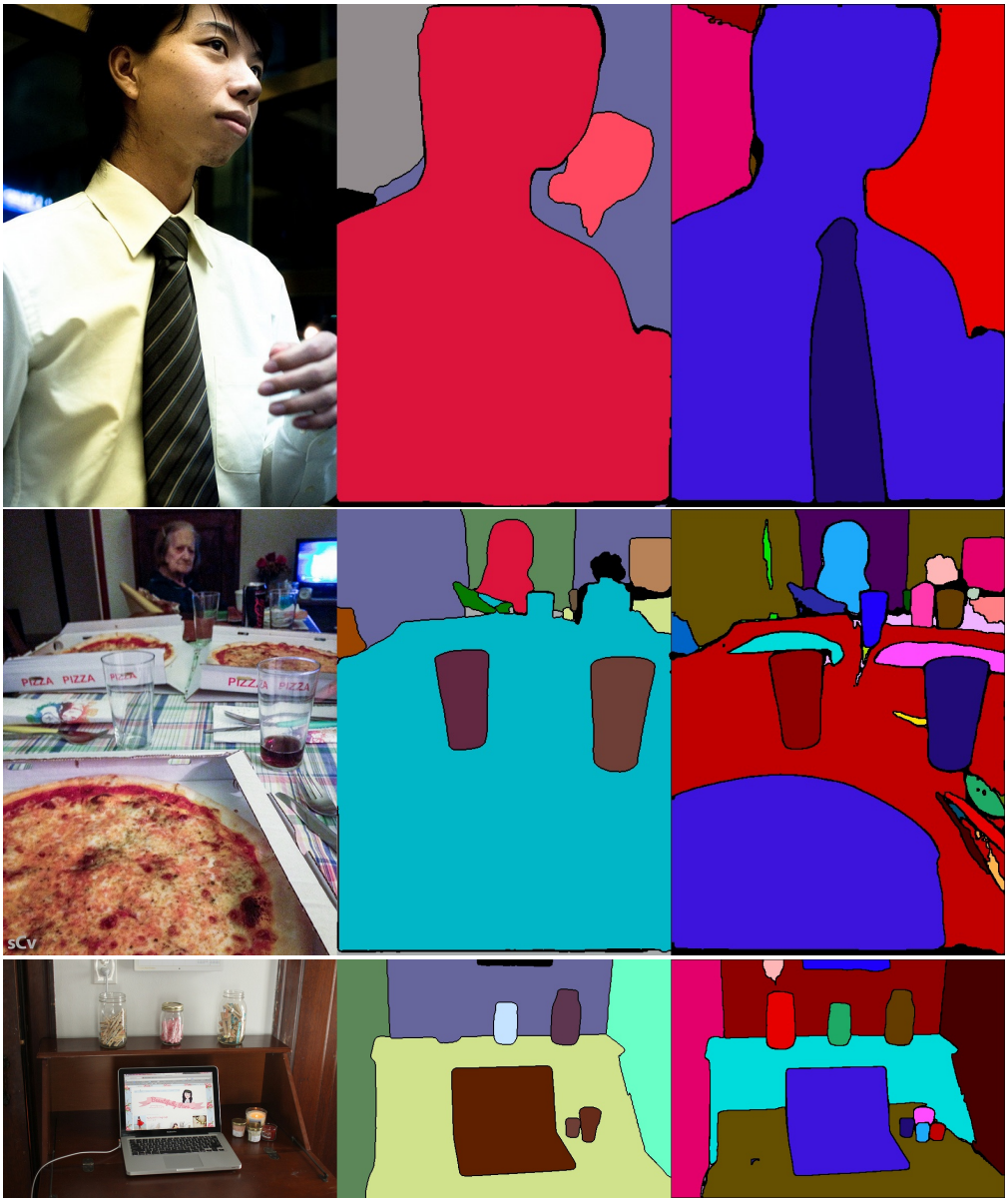
Image ID 19 with median user score

Image ID 35 with highest user score

Figure 6: The visualization of images and their corresponding entity annotations, which are used for our user study.

(1) The input (2) Panoptic FCN. (3) Our results.

Figure 7: Visual comparisons on COCO.



(1) The input (2) Panoptic FCN. (3) Our results.

Figure 8: Visual comparisons on COCO.

(1) The input (2) Our results.

Figure 9: Entity segmentation on COCO.

(1) The input (2) Our results.

Figure 10: Entity segmentation on COCO.

(1) The input (2) Our results.

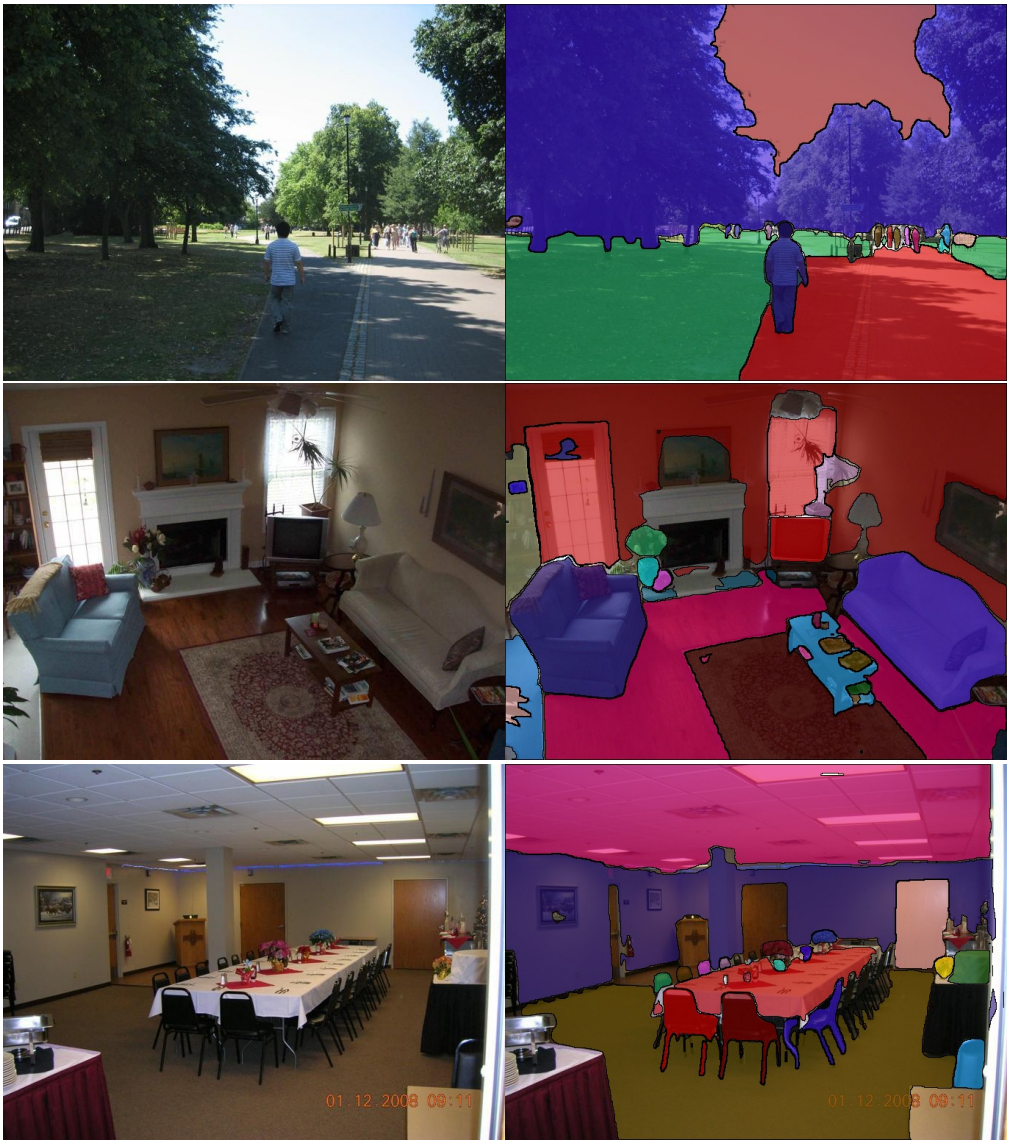
Figure 11: Entity segmentation on COCO.

(1) The input (2) Our results.

Figure 12: Entity segmentation on COCO.

(1) The input (2) Our results.

Figure 13: Entity segmentation on ADE20K.



(1) The input (2) Our results.

Figure 14: Entity segmentation on ADE20K.

(1) The input (2) Our results.

Figure 15: Entity segmentation on Cityscapes.

(1) The input (2) Our results.

Figure 16: Entity segmentation on Places2.

(1) The input (2) Our results.

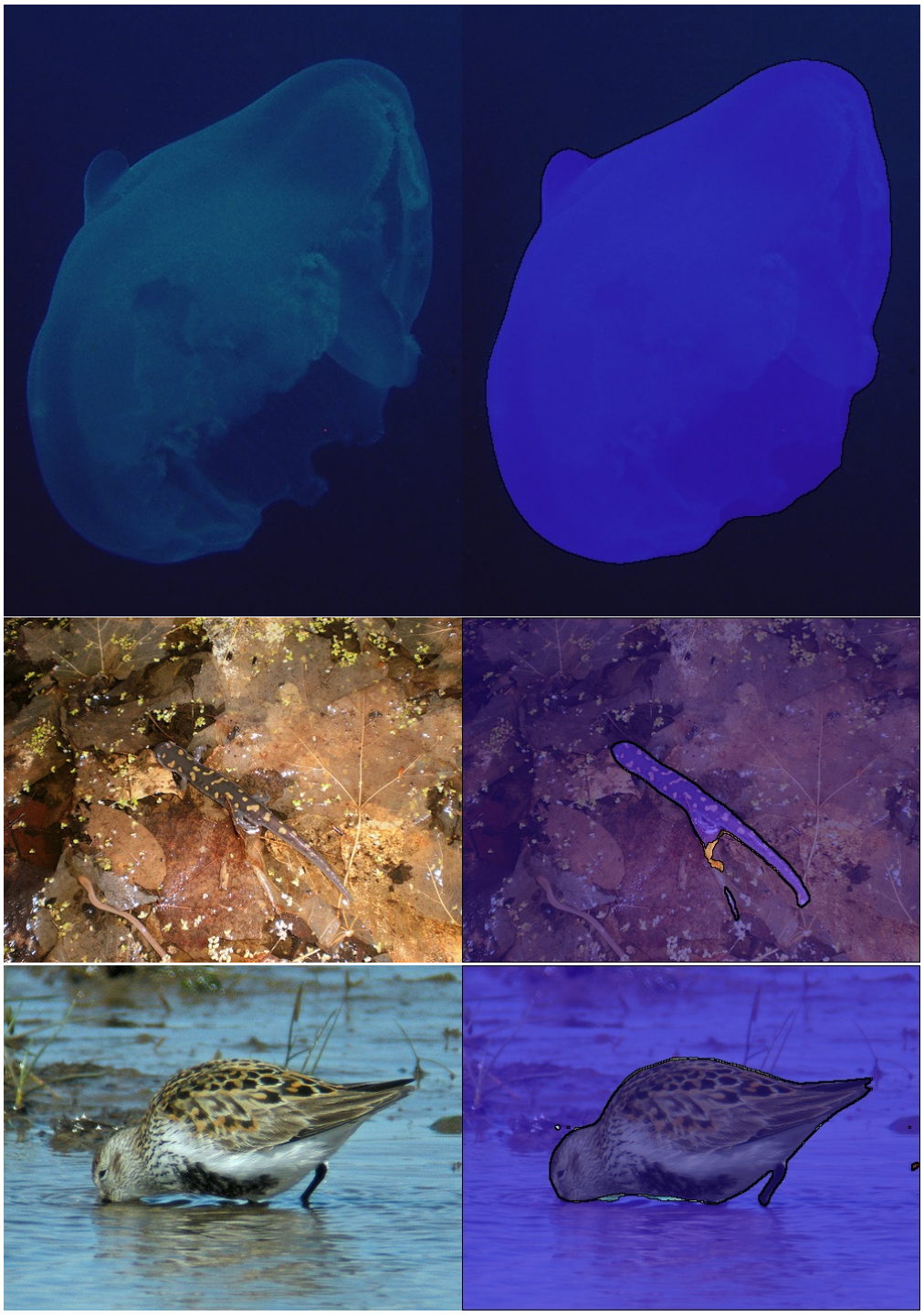
Figure 17: Entity segmentation on ImageNet.

(1) The input (2) Our results.

Figure 18: Entity segmentation on ImageNet.

(1) The input (2) Our results.

Figure 19: Entity segmentation on ImageNet.



(1) The input (2) Our results.

Figure 20: Entity segmentation on ImageNet.

(1) The input (2) Our results.

Figure 21: Entity segmentation on Object365.

(1) The input (2) Our results.

Figure 22: Entity segmentation on Object365.

Figure 23: The bokeh and recoloring image editing applications enabled by our entity segmentation.