Learning to Detect Every Thing in an Open World

Kuniaki Saito[1] Ping Hu[1] Trevor Darrell[2] Kate Saenko[1,3]

[1]Boston University [2]University of California, Berkeley [3]MIT-IBM Watson AI Lab

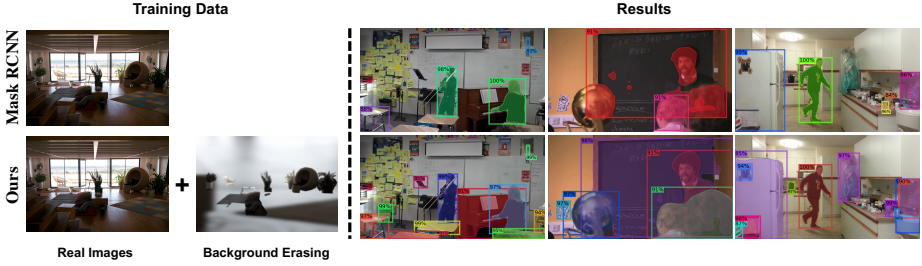


Fig. 1: In an open world detection task, the model must locate and segment all objects in the image irrespective of categories used for training. **Left**: We propose a new multi-domain training scheme using real images and augmented images with "erased" background. **Right**: When training on COCO [30] and testing on UVO [41], our detector correctly localizes many objects that are not labeled in COCO with the help of our new data augmentation and training scheme.

Abstract. Many open-world applications require the detection of novel objects, yet state-of-the-art object detection and instance segmentation networks do not excel at this task. The key issue lies in their assumption that regions without any annotations should be suppressed as negatives, which teaches the model to treat any unannotated (hidden) objects as background. To address this issue, we propose a simple yet surprisingly powerful data augmentation and training scheme we call Learning to Detect Every Thing (LDET). To avoid suppressing hidden objects, we develop a new data augmentation method, BackErase, which pastes annotated objects on a background image sampled from a small region of the original image. Since training solely on such synthetically-augmented images suffers from domain shift, we propose a multi-domain training strategy that allows the model to generalize to real images. LDET leads to significant improvements on many datasets in the open-world instance segmentation task, outperforming baselines on cross-category generalization on COCO, as well as cross-dataset evaluation on UVO, Objects365, and Cityscapes.

Keywords: Open World Instance Segmentation

1 Introduction

Humans routinely encounter new tools, foods, or animals, having no problem perceiving the novel objects as *objects* despite having never seen them before.

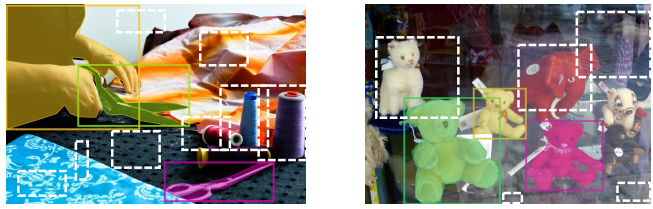


Fig. 2: **Hidden objects:** Existing datasets do not exhaustively label all objects, leading to detectors that are ill-prepared to propose boxes for long-tail categories. Colored boxes are annotated boxes while white-dashed boxes are potential background regions. Many white-dashed regions locate objects, but are regarded as background during training. This can suppress the objectness of novel objects.

Unlike humans, current state-of-the-art detection and segmentation methods [20,31,35,36,29] have difficulty recognizing novel objects as *objects* because these methods are designed with a closed-world assumption. Their training aims to localize known (annotated) objects while regarding unknown (unannotated) objects as *background*. This causes the models to fail in locating novel objects and learning general *objectness*. One way to deal with this challenge is to create a dataset with an exhaustive annotation of every single object in each image. However, creating such datasets is very expensive. In fact, many public datasets [30,15,44] for object detection and instance segmentation do not label all objects in an image (Fig. 2).

Failing to learn general objectness can cause issues in many applications. For instance, embodied AI (e.g., robotics, autonomous driving) requires localizing objects unseen during training. Autonomous driving systems need to detect novel objects in front of the vehicle to avoid accidents though identifying the category is not necessarily required. In addition, zero-shot, and few-shot detection have to localize objects unseen during training. Open-world instance segmentation [41] aims to localize and segment novel objects, but the state-of-the-art model [20] does not perform well as shown in [41].

We find that the failure of current state-of-the-art models is partly due to the training pipeline, *i.e.*, regarding all regions that are not annotated as the foreground objects as background. Even if the background includes *hidden* objects—background objects that are visible but unlabeled—as in Fig. 2, the models are trained not to detect them, which prevents from learning general objectness. To address this, Kim *et al.* [26] proposed to learn the localization quality of region proposals instead of classifying them as foreground vs. background. Their approach samples object proposals close to the ground truth and learns to estimate the corresponding localization quality. While partially mitigating the issue, this approach still needs to carefully set the overlap threshold for positive/negative sampling and risks suppressing hidden objects as non-objects.

To improve open-world instance segmentation, we propose a simple, yet powerful, learning framework along with a new data augmentation method, called *Learning to Detect Every Thing (LDET)*. To eliminate the risk of suppressing hidden objects, we copy foreground objects using their mask annotation and

paste them onto a background image. The background image is synthesized by resizing a cropped patch. By keeping the cropped patch small, we make it unlikely that the resulting synthesized images contain any hidden objects. However, this background creation process makes synthesized images look very different from real images, *e.g.*, the background may consist only of low-frequency content. Thus, a detector naively trained on such images performs poorly. To overcome this limitation, we decouple the training into two parts: 1) training background and foreground region classification and localization heads with synthesized images, and 2) learning a mask head with real images. We show that such hybrid training on both domains but with a shared backbone makes the model invariant to the domain shift between augmented and real images.

LDET demonstrates remarkable gains in open-world instance segmentation and detection. On COCO [30], LDET trained on VOC categories improves the average recall by 12.8 points when evaluated on non-VOC categories. Surprisingly, LDET achieves significant improvements in detecting novel objects without requiring additional annotation *e.g.*, LDET trained only on VOC categories (20 classes) in COCO outperforms Mask RCNN trained on all COCO categories (80 classes) when evaluating average recall on UVO [41]. As shown in Fig. 1, LDET can generate precise object proposals as well as cover many objects in the scene.

Our contributions are summarized as follows:

- We propose a simple framework, LDET, consisting of new data augmentation and decoupled training for open-world instance segmentation, which is applicable to both one-stage and two-stage detectors.
- We demonstrate that both our data augmentation and decoupled training are crucial to achieving good performance in open-world instance segmentation.
- LDET outperforms state-of-the-art methods in all settings including cross-category settings on COCO and cross-dataset setting on COCO-to-UVO, COCO-to-Object365, and Cityscapes-to-Mapillary.

2 Related Work

Region proposals. Unsupervised region proposal generation used to be a standard approach to localize objects in a scene [45,1,2,39]. These approaches localize objects in a class-agnostic way, but employ hand-crafted features (*i.e.*, color contrast, edge, *etc.*) to capture general objectness.

Closed-World object detection. Much effort has been spent on supervised object detection with a closed world assumption [16,19,36,18,31,35]. The ability to detect known objects has been improving with better architecture designs [28,5,10] or objectives [29]. Also, localizing objects given a few training examples or semantic information is becoming a popular research topic [25,3]. However, these attempts are still constrained by the taxonomy defined by the dataset. Our model can detect more categories than defined by the dataset, which can be very useful in few-shot or zero-shot object detection.

Open-World object detection/segmentation. Open-world recognition problems are gaining attention in image classification, object detection, and segmentation [4,11,6]. Especially, many methods have been proposed for open-set image

classification, where the goal is to separate novel categories from known categories given a closed-set training set [4,32,34,43,38]. On the contrary, the goal of open-world instance segmentation is to detect and segment all objects in a scene without distinguishing novel objects from seen ones. We acknowledge that there is ambiguity in the definition of "object", and follow [41] during evaluation.

Wang *et al*. [41] recently published the first benchmark dataset for open-set instance segmentation, which includes various categories from YouTube videos. However, from a methodological perspective, open-world object detection and segmentation remain understudied despite the importance of the task. Hu *et al*. and Kuo *et al*. [22,27] proposed approaches for predicting masks of various objects, but they require bounding boxes from classes of interest. Jaiswal *et al*. [23] trained a detector in an adversarial manner to learn class-invariant objectness. Joseph *et al*. [24] proposed a semi-supervised learning approach for open-world detection, which regards regions that are far from ground truth boxes but have a high objectness score as hidden foreground objects. Kim *et al*. [26] employed localization quality estimation with the claim that the estimation strategy is more generalizable in open-world instance segmentation. Note that this is a concurrent work.

The core of the open-world detection problem lies in the detector training pipeline: regarding hidden objects as background. This training scheme is common in both two-stage and one-stage detectors. However, none of the approaches listed above solves this issue. Our approach takes the first step in addressing background suppression via novel data augmentation strategies and shows remarkable improvements over baselines despite its simplicity.

Copy-Paste augmentation. Pasting foreground objects on a background is a widely used technique in many vision applications [12,17,40]. Recently, copy-and-paste augmentation was shown to be a very useful technique in instance segmentation [13,14,17]. Dwibedi *et al*. [14] proposed to synthesize an instance segmentation dataset by pasting object instances on diverse backgrounds and trained on the augmented images in addition to the original dataset. Dvornik *et al*. [13] considered modeling the visual context to paste the objects while Ghiasi *et al*. [17] showed that pasting objects randomly is good enough to provide solid gains. These methods still assume a closed-world setting, whereas our task is the open-world instance segmentation problem. There are two technical differences compared to these methods. First, our augmentation samples background images from a small region of an original image to create a background unlikely to have any objects. This pipeline is designed to circumvent suppressing hidden objects as background and does not require any external background data as used in [14]. Second, we decouple the training into two parts, which is also key to achieving a well-performing open-world detection model. In contrast, all of the existing approaches above simply train on synthesized images.

3 Learning to Detect Every Thing

In this section, we describe the proposed LDET scheme for open-world instance segmentation. During training, we are given an instance segmentation dataset

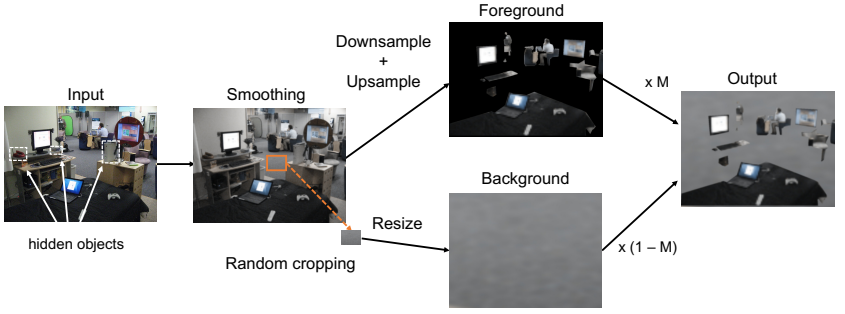


Fig. 3: **Our augmentation strategy** creates images without hidden objects by upscaling small regions to use as background.

with annotations of known classes. In testing, the model is required to locate objects of unknown classes.

Mask-RCNN [20] serves as the base model, but our method is applicable to different architectures such as RetinaNet [29] and TensorMask [7]. We describe details of the data generation process (Fig. 3) and training scheme (Fig. 5) below.

3.1 Data Augmentation: Background Erasing (BackErase)

We propose a new data augmentation to mitigate the bias induced by unlabeled objects prevalent in most training sets. These hidden objects are not given annotation because they do not belong to known classes or are overlooked by annotators. We propose to synthesize fully labeled training images. Using the instance mask, we crop only the annotated foreground regions and paste them on the synthesized background canvas. These synthesized images have fully labeled objects and lead to objectness detectors that generalize better to open world settings.

Background region sampling. First, we apply Gaussian smoothing to the input image before cropping the foreground and background region, and denote the smoothed image as I_1. By smoothing the whole image before this operation, we expect to reduce the discrepancy in high-frequency content between the foreground and background images.

Then, we randomly crop a small region from I_1, where width and height of the region is set as $\frac{1}{8}$ of the original image's. We resize it to the same size as the input image to serve as a background canvas, which we denote as I_2. Cropping a small region entails a much lower risk of including hidden background objects compared to using the original background. Even if it happens to include unannotated objects, drastically upscaling the patch makes the objects' appearance very different, as shown in examples in Fig. 4. We vary the scale of the background canvas in experiments (See Table 4).

Blending pasted objects. To avoid the model learning to separate background and foreground by the difference in frequency information, foreground objects are downsampled and resized to the original size. Then, the foreground objects

Fig. 4: **Examples of original inputs (odd columns) and synthesized image (even columns).** Masked regions are highlighed with colors (odd columns). Using small regions as background avoids the risk of having hidden objects in the background. We achieve to suppress unlabeled objects present in the background.

are pasted on the canvas. To insert copied objects into an image, we use the binary mask (M) of pasted objects using ground-truth annotations and compute the new image as $I_1 \times M + I_2 \times (1 - M)$. We apply a Gaussian filter to the binary mask to smooth the edges of the copied objects. Examples of synthesized images using the COCO dataset with 80 categories are illustrated in Fig. 4. Note that even in datasets with dense annotations like COCO, many objects are not annotated, and our augmentation effectively removes such hidden objects from the background. We do not claim that details such as smoothing and resizing operations are necessarily optimal for open-world instance segmentation, but empirically find they work well.

3.2 Decoupled Multi-Domain Training

Simply training a detector on the synthesized images in the conventional way [20] does not work well due to the domain shift (See Table 3). Since real images and our synthesized images have very distinct content and layout, a detector trained on our synthesized data does not generalize well to real images. In this section, we propose a simple yet effective approach to mitigate this issue. We solve the shift by computing mask loss on real images while calculating detection loss on synthesized images. Because the backbone is trained on both the synthetic and real domains, it learns an invariance between real and augmented object regions. Even though the losses are different for the two domains, they are highly correlated, which makes the backbone network adapted to real images on both tasks. The entire training pipeline is summarized in Fig. 5.

Typically, the training objectives for instance segmentation models consist of two major terms: object detection loss and instance mask loss. In methods like Mask RCNN [20], the object detection loss is composed of a region proposal classification loss and a box regression loss, which are used to train both the region proposal network (RPN) and the region of interest (ROI) heads. For simplicity, we unify the objectives for RPN and ROI as one loss.

Each predicted box B_i includes predictions for a box location $\hat{t}_i$, objectness score $\hat{y}_i$, and mask prediction $\hat{m}_i$. During training, B_i comes with corresponding

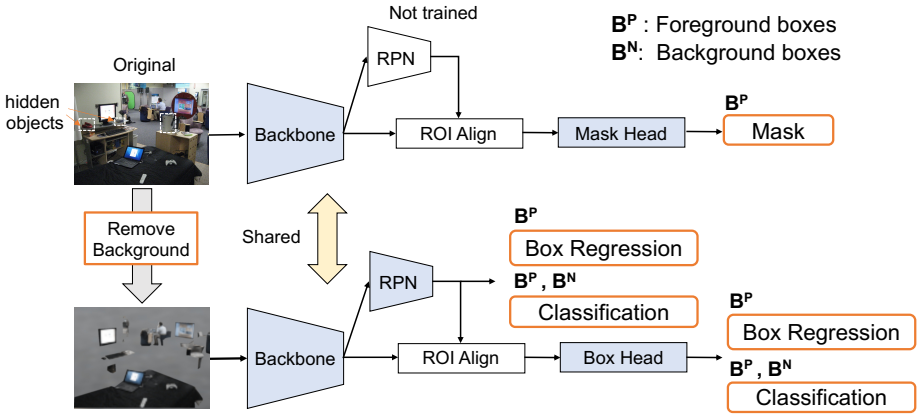


Fig. 5: **Training pipeline.** Given an original input image and synthesized image, we train the detector on the mask loss computed on the original image, and classification, regression losses on the synthesized image.

ground truth, t_i, y_i, and m_i. Let B^p and B^n, the positive (foreground) and negative (background) boxes, denote a set of boxes with $y = 1$ and $y = 0$ respectively. The label of positive or negative, y, is decided based on the Intersection over Union (IoU) with the nearest bounding box during training, *e.g.*, regions with IoU smaller/larger than 0.5 are background/foreground respectively in the case of ROI head. Note that, in general, the proposal classification loss is computed for both positive and negative boxes, $B = B^p \cup B^n$, while the box regression and mask loss are computed for B^p.

We highlight that B^n from synthetic images (I_S) are unlikely to contain unlabeled objects due to our augmentation while unlabeled objects can be present in those from real images (I_R). Given this fact, the detection loss should be computed on boxes of I_S. However, training only on I_S will not make the model generalizable to I_R since I_R and I_S look very different. Then, to bridge the domain gap between I_S and I_R, we propose to compute the instance mask loss on I_R. Why does the mask loss help to mitigate the gap w.r.t. the detection task? Mask prediction aims to separate background and foreground pixels within a foreground bounding box whereas the box classification decides the objectness for one box. The two tasks are very similar in that both attempt to separate background and foreground samples except that the mask loss is computed only for B^p. Then, the mask loss training signal, which should be useful to solve the detection task for I_R, is propagated to a deep backbone network shared among the region proposal network, bounding box head, and mask head. The features obtained from the backbone will improve the performance of the box head in I_R. Furthermore, the model will not learn to suppress unlabeled objects by using the mask loss because the loss is computed on foreground boxes only. In summary, the use of mask loss on real images will make the backbone network adapted to real images without suppressing unlabeled objects.

Method	Non-VOC						All			
	Box			Mask			Box		Mask	
	AP	AR_{10}	AR_{100}	AP	AR_{10}	AR_{100}	AR_{10}	AR_{100}	AR_{10}	AR_{100}
Mask RCNN [20]	1.5	8.8	10.9	0.7	7.2	9.1	19.3	23.1	16.7	19.9
Mask $RCNN^P$	1.1	8.7	10.7	0.6	7.2	8.9	19.1	23.0	16.5	19.8
Mask $RCNN^S$	3.4	13.2	18.0	2.2	11.3	15.8	21.7	27.4	19.2	24.4
LDET	**5.0**	**18.2**	**30.8**	**4.7**	**16.3**	**27.4**	**24.4**	**36.8**	**22.4**	**33.1**

Table 1: **Results of VOC → COCO generalization**. LDET outperforms all baselines and showing large improvements on Mask RCNN.

Method	Top-5					Worst-5				
	bear	bed	microwave	elephant	t-bear	carrot	tie	skis	broccoli	donut
Mask RCNN	**78.6**	45.2	36.5	28.6	20.3	0.4	0.4	1.2	1.2	1.2
LDET	76.5	**57.6**	**59.5**	**67.2**	**45.9**	**6.2**	**1.9**	**8.1**	**8.3**	**15.7**

Table 2: **AR on top- and worst-5 classes detected by Mask RCNN baseline in VOC → Non-VOC.**

Specifically, the loss is computed as follows:

$$\sum_{B_i \in B^p_{aug}} L_{reg}(\hat{t}_i, t_i) + \sum_{B_i \in B_{aug}} L_{cls}(\hat{y}_i, y_i) + \sum_{B_i \in B^p_{real}} L_{mask}(\hat{m}_i, m_i) \qquad (1)$$

where L_{reg}, L_{cls}, and L_{mask} indicate the regression, object classification, and mask loss respectively. Note that B_{aug} and B_{real} are used to differentiate the boxes from synthetic and real images.

Class agnostic inference. Since our goal is to detect objects in a scene without classifying them into closed-set classes, class agnostic inference is preferred. We apply a class agnostic inference method to a class discriminative object detector. Given the classification output of a region, we sum up all scores of (known) foreground classes, deeming the result as an objectness score. Mask and box regression are performed for the class with the maximum score.

4 Experiments

We evaluate LDET on two settings of open-world instance segmentation: *Cross-category* and *Cross-dataset*. The Cross-category setting is based on the COCO [30] dataset, where we split annotated classes into known and unknown classes, train models on known ones, and evaluate detection/segmentation performance on unknown and all classes separately. Since the model can be exposed to a new environment and encounter novel instances, the Cross-dataset setting evaluates models' ability to generalize to new datasets. For this purpose, we adopt either COCO [30] or Cityscapes [9] as a training source, with UVO [41], Obj365 [37] and Mappilary Vista [33] as the test datasets.

Implementation. Detectron2 [42] is used to implement LDET. Mask R-CNN [20] with ResNet-50 [21] as feature pyramid [28] is used unless otherwise specified.

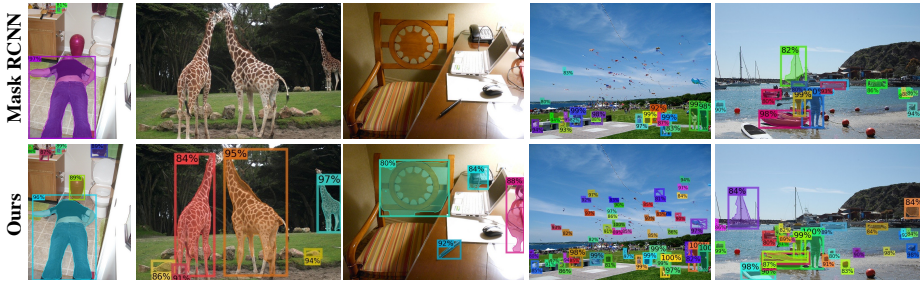


Fig. 6: **Visualization in VOC to Non-VOC in COCO dataset. Top: Mask RCNN. Bottom: LDET.** Note that training categories do not include giraffe, trash box, pen, kite, and floats. LDET detects many novel objects better than Mask RCNN.

Method	Box		Mask		Box		
	Real	Synth	Real	Synth	AR_{10}	AR_{100}	AR0.5
Plain	✓		✓		8.8	10.9	19.1
Synth Only		✓			1.6	4.3	11.7
Synth Only*		✓		✓	3.0	9.5	23.8
LDET		✓	✓		**18.2**	**30.8**	**53.2**

Table 3: **Ablation study of data and training method in VOC → Non-VOC.** We change the data used to compute detection and mask loss. Training only on synthetic data does not perform well while LDET, which is trained on both data with decoupled training, performs the best.

Following [41], we utilize the standard hyperparameters of Mask R-CNN [20] as defined in Detectron2. See appendix for more details *e.g.*, hyper-parameters. We will release the full code to reproduce our results upon acceptance.

Baselines. Since open-world instance segmentation is a new task, we develop several baselines as follows. See appendix for more details of baselines.

1) Mask R-CNN. We adopt the default model without changing any objectives or input data. Comparison to this baseline will reveal the difference from standard training.

2) Mask RCNN^S. We avoid sampling background regions with hidden objects by sampling background boxes from the regions mostly inside the ground truth boxes. We assume that these regions are less likely to contain hidden objects. We compute the area of intersection with ground truth boxes over the area of the proposal box and sample background boxes with a large value of this criterion.

3) Mask RCNN^P. Inspired by [24], we implement a pseudo-labeling based open-set instance segmentation baseline. The idea is to assign pseudo-labels of foreground classes to the background regions (IoU with GT < 0.5) that have high objectness scores. A model is trained to minimize the box classification loss on pseudo-labels.

Evaluation. In this work, Average Recall (AR) is mainly employed for performance evaluation following [41]. When class labels are available, we compute

AR for each class given the objectness score and average over all classes as done in the standard COCO evaluation protocol. Unless otherwise specified, AR is computed following the COCO evaluation protocol, *i.e.*, AR at 100 detections. Average precision (AP) is computed in a class agnostic way.

4.1 Cross-category generalization

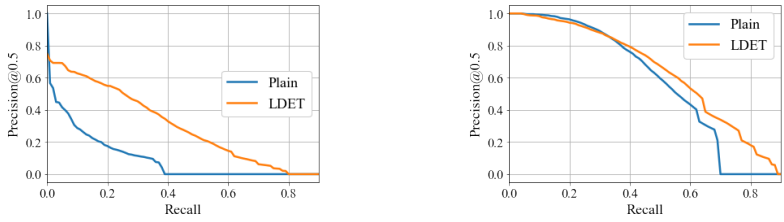
Setup. We split the COCO dataset into 20 seen (VOC) classes and 60 unseen (non-VOC) classes. We train a model only on the annotation of 20 VOC classes. The hyper-parameters of baselines and LDET are chosen based on the performance of the randomly selected 20 Non-VOC classes. Then, the whole validation split of COCO is adopted for evaluation. To better understand the results, we report AR on two settings *i.e.*, 60 non-VOC classes only (novel class evaluation) and all 80 classes (generalized evaluation). In evaluating AR in novel class evaluation, we do not count the "seen class" detection boxes into the budget of the recall when computing the score. This is to avoid evaluating any recall on seen-class objects.

Comparison with baselines. As shown in Table 1, our method outperforms baselines in all metrics with a large margin. The difference is more evident in the results on non-VOC classes. Some visualizations are available in Fig. 6. Mask RCNN tends to overlook non-VOC class objects even when they are in the dominant and salient region, *e.g.*, *trash can* (leftmost column) and *giraffes* (second leftmost column). On the other hand, LDET generalizes well to novel objects such as *comb*, *towel*, *giraffes*, *pen*, *phone*, *kites*, and *floats*. Table 2 describes AR of top- and worst-5 classes in Mask RCNN. Mask RCNN outperforms LDET on *bear* probably because there are several categories similar to *bear*, *e.g.*, *dog* and *horse* or maybe because there are no 'bear' hidden objects. Although both LDET and the baseline do not excel at detecting classes whose appearance is dissimilar to VOC classes, LDET outperforms Mask RCNN.

Precision-Recall curve. Fig. 7 (a) shows precision and recall curve measured on non-VOC classes. In most points, the precision of LDET is better than that of the plain model, which means that LDET outputs more precise bounding boxes for novel objects.

Ablation study for learning objectives. Table 3 shows an ablation study of training data and objectives. If a model learns only from synthetic data (Synth Only), it fails to detect and segment objects. Interestingly, adding the synthetic mask loss on top of the synthetic detection data (Synth Only*) improves the performance. This supports our claim that mask prediction and detection tasks are highly correlated. Adding the mask loss even for synthetic data provides a better understanding of the location of the objects and improves performance. Computing detection loss of synthetic data and mask loss on real data obtains the best results. These results indicate that our proposed decoupled training is very suitable for the open-world instance segmentation and detection task.

Visualization of the learnt objectness map. Fig. 8 visualizes the confidence scores map of the region proposal network, computed by averaging outputs from all feature pyramids. Models trained only on synthetic data (first and second from the leftmost) cannot separate foreground and background well though they



(a) VOC to Non-VOC. (b) VOC to UVO.

Fig. 7: **Precision-Recall Curve.** The precision is measured at the IoU threshold of 0.5. The comparison demonstrates that LDET detects novel objects more precisely than the plain model.

Background ratio	AP	AR_{10}	AR_{100}	AR_{small}	AR_{medium}	AR_{large}
2^{-1}	**5.2**	16.8	26.2	17.0	27.0	40.6
2^{-2}	5.0	16.5	25.9	17.3	27.1	38.9
2^{-3}	5.0	**18.2**	30.8	18.8	34.6	44.5
2^{-4}	**5.2**	17.5	**31.8**	**20.1**	**35.8**	**45.6**

Table 4: **Varying the size of background regions.** 2^{-m} indicates cropping background region with 2^{-m} of width and height of an input image. Sampling background from smaller region tends to improve AR.

seem to cover many foreground objects. While the model trained only with real data (second from rightmost) suppresses the score for many objects, LDET (rightmost) correctly captures objectness for diverse objects. For instance, it captures objectness well in an image crowded with objects in the first row. The mask loss on real images seems to help the detector to separate foreground and background well.

Size of background regions. The size of background regions can be important in our data augmentation: if the size is close to an original image, the background will include many hidden objects. We analyze the effect of the region's size in Table 4, wherein AR gets better with smaller sizes. This indicates that smaller backgrounds prevent sampling hidden objects and coverage of novel objects (AR) becomes better.

External data for background. Using an external background dataset is an alternative way to synthesize background regions although an image without background is not always accessible. In this experiment, we use DTD [8] (texture image dataset). The background region is replaced with a randomly cropped patch from DTD dataset, and a model is trained in the same way as LDET. See the appendix for more details of training. The resulting AR is 26.7 (LDET -3.1) in bounding box localization. The dataset includes a considerable number of objects despite being primarily a texture dataset, which is probably the cause of degradation in AR.

Region proposal network (RPN) and region of interest (ROI) head. We compare the performance of RPN and ROI heads in Table 5. In Mask RCNN,

Method	Detector	AR_{10}	AR_{50}	AR_{100}
Plain	RPN	**11.0**	**19.4**	**22.9**
Plain	ROI	8.8	10.8	10.9
LDET	RPN	15.4	26.4	30.8
LDET	ROI	**18.2**	**28.0**	**30.8**

Table 5: **Comparison between region proposal network and region of interest head.**

Detector	Method	AR_{10}	AR_{50}	AR_{100}
RetinaNet	Plain	9.9	15.7	17.8
	LDET	**15.3**	**26.7**	**31.0**
TensorMask	Plain	10.6	17.6	19.7
	LDET	**16.3**	**26.8**	**31.1**

Table 6: **Results on RetinaNet and TensorMask**

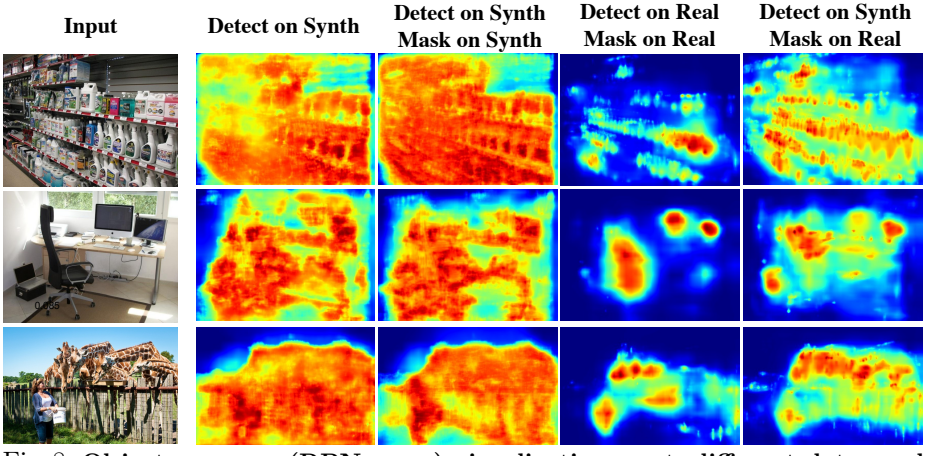


Fig. 8: **Objectness map (RPN score) visualization w.r.t. different data used to compute detection and mask loss.** A model only with the detection loss for synthetic data (leftmost) does not suppress background regions well. Adding mask loss on real data (rightmost) captures objectness of various categories whereas a plain model (second from the right) tends to suppress many objects.

RPN covers more novel objects, which means ROI learns to suppress many novel objects. In contrast, ROI of LDET is comparable or better than RPN in all metrics.

Evaluation on different architectures. We evaluate LDET on one-stage detectors, RetinaNet [29] and TensorMask [7] in Table 6. See appendix for the details of the experiment. LDET shows clear gains over the baseline, which shows that LDET is a universal approach that can be integrated into diverse detectors.

4.2 Cross-dataset generalization

COCO to UVO. First, we utilize UVO [41], which covers many categories outside COCO as 57% object instances do not belong to any of the 80 COCO classes. Since UVO is based on Youtube videos, the appearance is very different from COCO *e.g.*, some videos are egocentric views and have significant motion blur. We test models trained on the COCO-VOC split or the whole COCO. The validation split is used to measure the performance. Since this dataset does not

Method	Train	Box					Mask				
		AP	AR	AR_{small}	AR_{med}	AR_{large}	AP	AR	AR_{small}	AR_{med}	AR_{large}
Mask RCNN	VOC (COCO)	19.8	30.0	10.7	21.3	43.0	15.5	23.9	9.2	18.5	32.8
Mask $RCNN^P$		19.2	30.1	10.6	21.3	43.3	15.4	24.1	9.4	18.4	33.2
Mask $RCNN^S$		19.7	32.0	10.0	23.3	46.0	14.1	25.9	9.5	20.2	35.4
LDET		**22.4**	**43.7**	**24.7**	**39.9**	**52.9**	**18.4**	**36.0**	**22.1**	**34.8**	**41.4**
Mask RCNN	COCO	25.3	42.3	22.2	38.3	52.0	20.6	35.9	19.6	33.9	42.6
Mask $RCNN^P$		24.4	41.9	22.3	37.8	51.5	20.1	35.4	19.7	33.6	41.8
Mask $RCNN^S$		23.4	40.5	17.6	34.9	52.3	18.0	34.7	16.6	31.5	42.8
LDET		**25.8**	**47.5**	**29.1**	**44.8**	**55.6**	**21.9**	**40.7**	**26.8**	**40.0**	**45.7**

Table 7: **Results of COCO → UVO generalization. Top rows: Models trained on VOC-COCO. Bottom rows: Models trained on COCO.** LDET demonstrates high AP and AR in all cases compared to baselines.

Method	Non-COCO					All			
	AP	AR	AR_{small}	AR_{med}	AR_{large}	AR	AR_{small}	AR_{med}	AR_{large}
Mask RCNN [20]	11.9	34.4	21.2	36.0	45.8	38.5	24.0	40.1	50.2
Mask $RCNN^P$	11.8	32.7	17.5	33.5	47.1	38.6	24.5	39.8	50.8
Mask $RCNN^S$	10.9	34.6	21.9	35.7	46.6	35.9	18.9	36.6	50.2
LDET	**12.9**	**38.9**	**25.5**	**41.8**	**50.2**	**41.1**	**26.1**	**43.8**	**52.8**

Table 8: **Results of COCO → Obj365 generalization**. Improvement on Mask RCNN is shown next to each result in the row of LDET. LDET outperforms all baselines and showing large improvements on Mask RCNN.

provide class labels, we evaluate the performance in a class-agnostic way. As shown in Table 7, in both COCO-VOC and COCO settings, LDET outperforms baselines with a large margin. Note that our VOC-COCO model outperforms Mask RCNN trained on COCO in many metrics. This indicates the remarkable label efficiency of LDET in open-world instance segmentation. Unlike the result in VOC-NonVOC experiment, the AP of Mask $RCNN^S$ drops compared to Mask RCNN, probably because their region sampling leads to imbalanced sampling with regions with different scales. Fig. 7(b) describes the trade-off between precision and recall in this setting, which shows the advantage of LDET at most points.

COCO to Obj365. Second, we evaluate models on the validation split of Obj365 [37] detection dataset, wherein 60% object instances do not belong to any of the 80 COCO classes. We test models trained on the whole coco, and evaluation is done in the way as cross-category setting. As shown in Table 8, in both non-COCO categories and all categories, LDET outperforms all baselines. This result confirms that LDET is generalizable to detect various categories of objects.

Cityscape to Mapillary. We examine performance in autonomous driving scenes. Detectors are trained on Cityscape [9] (8 foreground classes, *person, rider, car, truck, bus, train, motorcycle,* and *bicycles*) and tested on the validation set of Mapillary Vistas [33] with 35 foreground classes including not only vehicles, but also animals, *trash can, mailbox*, and so on. In Table 9, LDET shows solid

Method	Box				Mask			
	AP	AR_{10}	AR_{100}	$AR_{0.5}$	AP	AR_{10}	AR_{100}	$AR_{0.5}$
Mask RCNN	8.2	7.7	11.1	20.2	7.3	6.1	8.4	16.3
Mask $RCNN^P$	6.9	7.4	10.8	19.3	7.5	5.5	7.9	16.3
Mask $RCNN^S$	8.3	6.7	13.3	26.9	6.3	5.5	10.2	21.0
LDET	**8.5**	**8.0**	**14.0**	**28.0**	**7.8**	**6.7**	**10.6**	**21.8**

Table 9: **Results of Cityscapes → Mappilary Vista generalization.** LDET is effective for autonomous driving dataset. $AR_{0.5}$ denotes AR with IoU threshold = 0.5

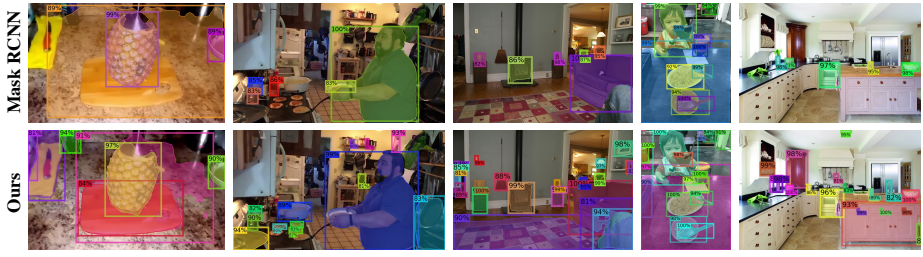


Fig. 9: **Visualization of results for models trained on COCO.** Top: Mask RCNN. Bottom: LDET. Leftmost two images are from UVO, the others are from COCO validation images.

gains over baselines, though this setting is very challenging. Note that the model is trained only on 8 classes and is required to generalize to all the 35 classes in the test data, which explains the lower performance compared to experiments on COCO. The result demonstrates that LDET generalizes to datasets other than COCO and can be useful for autonomous driving systems.

Visualization. As Fig. 1 shows, Mask RCNN detector fails in localizing objects unseen in their 80 categories while our detector shows surprisingly good generalization. Note that in the second leftmost row, it recognizes a character drawn on the wall, which is clearly outside COCO categories. Fig. 9 visualizes other examples from UVO and COCO.

5 Conclusion

In this paper, we presented a simple approach, LDET, for the challenging task of open-world instance segmentation. LDET consists of synthesizing images without hidden objects in their background as well as decoupled training for real and synthesized images. LDET demonstrates strong performance on a benchmark dataset of open-world instance segmentation and promising results on autonomous driving datasets. We hope that LDET becomes a simple baseline and accelerates further research in this area.

Limitations. As seen in several visualizations, LDET still fails in detecting some novel objects although its performance is much better than baselines. If the appearance of novel objects is distinct from known objects, LDET and most baselines may miss them. Also, experiments on Cityscapes (Table 9) indicate the importance of covering various categories in training data. One way to overcome this limitation is to annotate a wide range of categories for training data. **Acknowledgments.** This work was supported by DARPA LwLL and NSF Award

No. 1535797. We thank Donghyun Kim and Piotr Teterwak for giving feedback on the draft

A Experimental Details

In this appendix, we provide experimental details, additional analysis, and visualizations.

Data augmentation. Alg. 1 shows the pytorch-style pseudo-code for our data augmentation.

Implementation. We use the default hyper-parameters, provided by Detectron2, to train and test LDET, Mask RCNN, Mask RCNN$^{\mathrm{S}}$, and Mask RCNN$^{\mathrm{P}}$. Default configuration files are used for COCO [1] and Cityscapes [2] respectively. 2 GPUs of NVIDIA RTX A6000 with 48GB are used to train models.

One-Stage Detector. We use the same hyper-parameters for data augmentation as in Mask RCNN. Also, we follow the default hyper-parameters of RetinaNet and TensorMask. Since RetinaNet does not have mask head by default, we add the mask head on top of the feature pyramid following Mask RCNN. We will publish the code of one-stage detector upon accepntance.

Baselines.

1) Mask R-CNN. We do not make any change to the default training configuration.

2) Mask RCNN^{S}. We compute the area of intersection with ground truth boxes over the area of the proposal box, which we call *IoA*, and sample background boxes with a large value of this criterion. In both region proposal network and roi head, we pick background regions whose IoA is larger than 0.7.

3) Mask RCNN^{P}. Given the classification output (after softmax) from roi head, boxes confidently predicted as one of the foreground classes are choosen from background regions. The threshold to pick the pseudo-foreground is set as 0.9. The classification loss on the pseudo-foreground regions is incorporated to train the detector.

Experiments on texture dataset. To make a background using images of DTD [8], we crop the patch with the size of 256 x 256, and rescale it to the size of a detection training image. Then, we blend the foreground and background in the same way as LDET.

B Analysis

Study on the confidence threshold. In Fig. B, we vary the confidence threshold used to remove unconfident bounding boxes in ROI classification head, where the value is set as 0.05 by default. Here, we vary thresholds starting from 0.0 (no thresholding) to 0.5. This result demonstrates that the baseline drops AR by

[1] `detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_1x.yaml`

[2] `detectron2/blob/main/configs/Cityscapes/mask_rcnn_R_50_FPN.yaml`

Algorithm 1: PyTorch-style pseudocode for our data augmentation

```
# scale=1/8: the size of background region to crop.
# M: mask of the foreground regions.
# Apply gaussian smoothing.
image = gaussian(image)
w, h = image.shape
# Randomly crop background with the specified size.
backg = randomcrop(image, w*scale, h*scale)
# Upscale to the size of the input.
backg = upscale(backg, scale)
# Downsample the input.
image = downsample(image, scale)
# Upsample to the original size.
image = upscale(image, scale)
# Paste foreground objects on the synthesized background.
image = M * image + (1 - M) * backg
# Apply smoothing.
image = smooth(image)
```

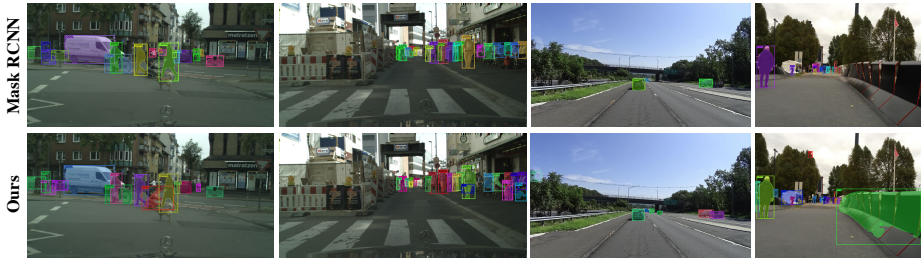


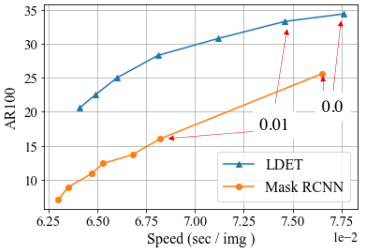
Fig. A: **Visualization for detectors trained on Cityscapes.** Leftmost two images are validation images of Cityscapes, rightmost two are from Mapillary.

applying a very small threshold value (Compare AR at 0.0 and 0.01), meaning that the baseline confuses many novel objects with the background.

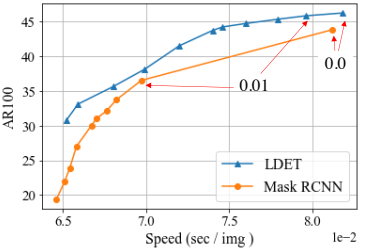
C Visualization

Cityscapes. Fig. A visualizes some qualitative results. Leftmost two images are from the validation set of Cityscapes, others are from Mapillary. We see that, as indicated by the quantitative results, LDET detects more objects, *e.g.*, *baby carriage* in the leftmost image. However, it is also true that LDET misses novel objects such as *dog* in the leftmost image, probably because there are no categories similar to dogs in the Cityscapes' 8 training categories. This fact indicates some room for improvement in our approach.

More visualizations in COCO. Fig. C and D are additional visualizations in VOC-COCO and COCO, respectively. Note that we add the results of Mask RCNN$^{\mathrm{S}}$, which are not visualized in the main paper due to a limited space. Mask RCNN$^{\mathrm{S}}$



(a) VOC to Non-VOC.



(b) VOC to UVO.

Fig. B: **Speed (sec /image) v.s. AR.** We vary the confidence threshold of the ROI head and see the changes of the speed and AR. Note that the speed changes due to the non-maximum supression after confidence thresholding. Points at confidence threshold at 0.0 and 0.01 are highlighted with red arrows. The baseline mask rcnn significantly drops performance between the points at 0.0 and 0.01, which indicates that the model suppresses many foreground objects at the confidence value of 0.01.

locates many novel objects while generating many false positives. This is probably due to the imbalanced sampling of background regions. By contrast, LDET detects many novel objects, *e.g.*, *elephants, toilet paper, lizard, statue, toy, etc.*, with small number of false positives.

Demo on video. Fig. E and F are demo of applying LDET to UVO [41] videos. Click the images to play the videos.

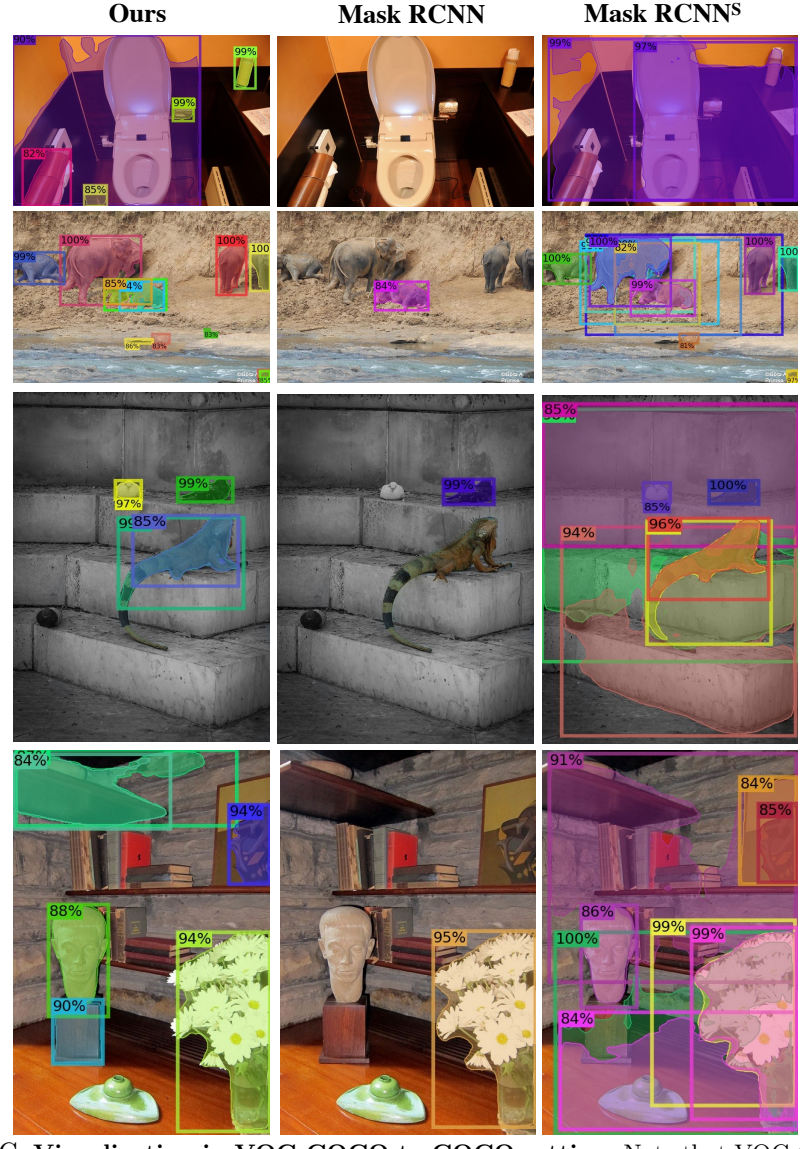


Fig. C: **Visualization in VOC-COCO to COCO setting.** Note that VOC-COCO does not contain objects such as lizard, toilet paper, and elephant.

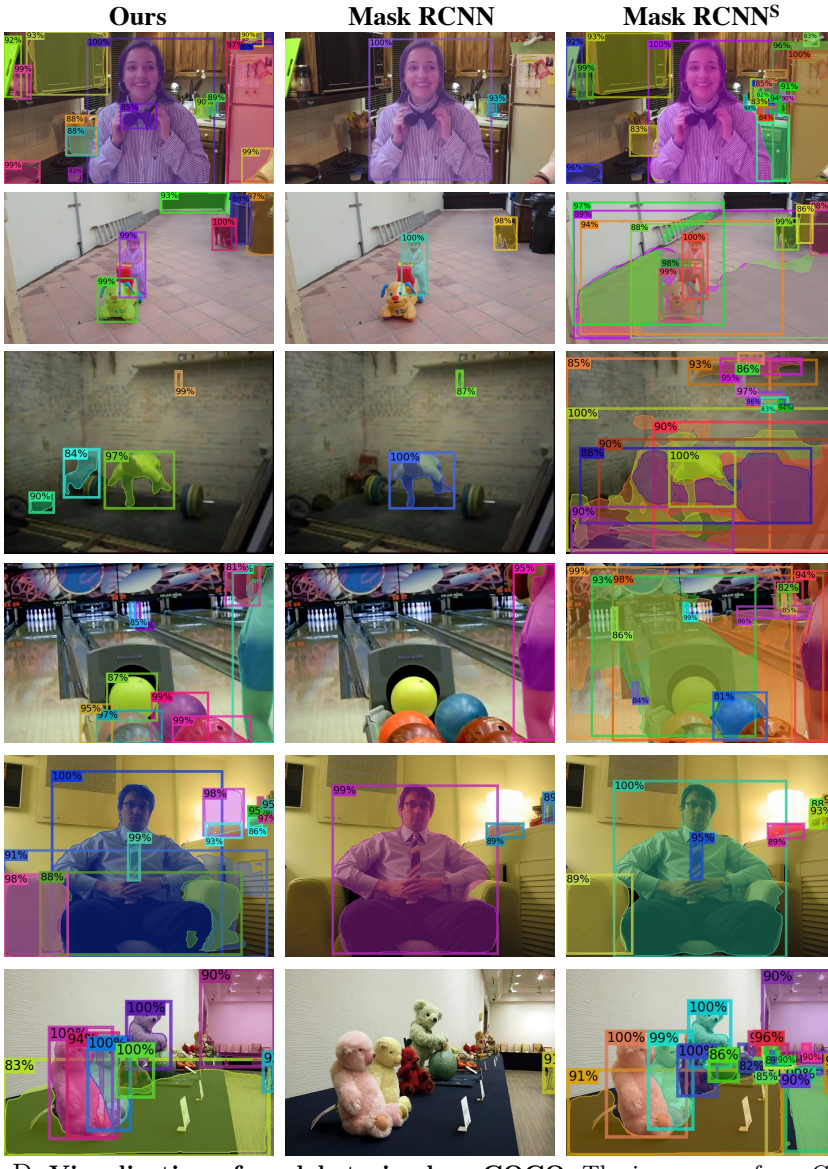


Fig. D: **Visualization of models trained on COCO.** The images are from COCO and UVO.

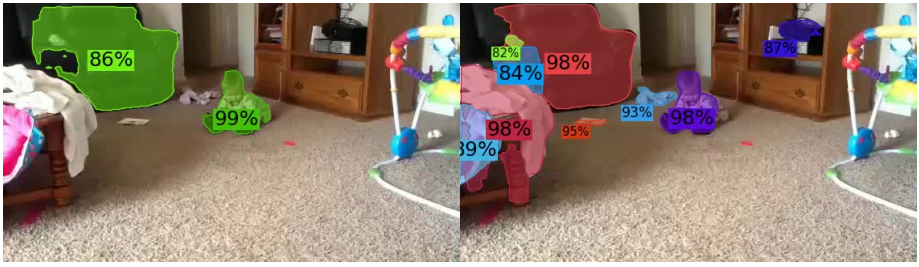


Fig. E: **Video demo of models trained on COCO. Left: Mask RCNN. Right: LDET. Click the image to play the video.**

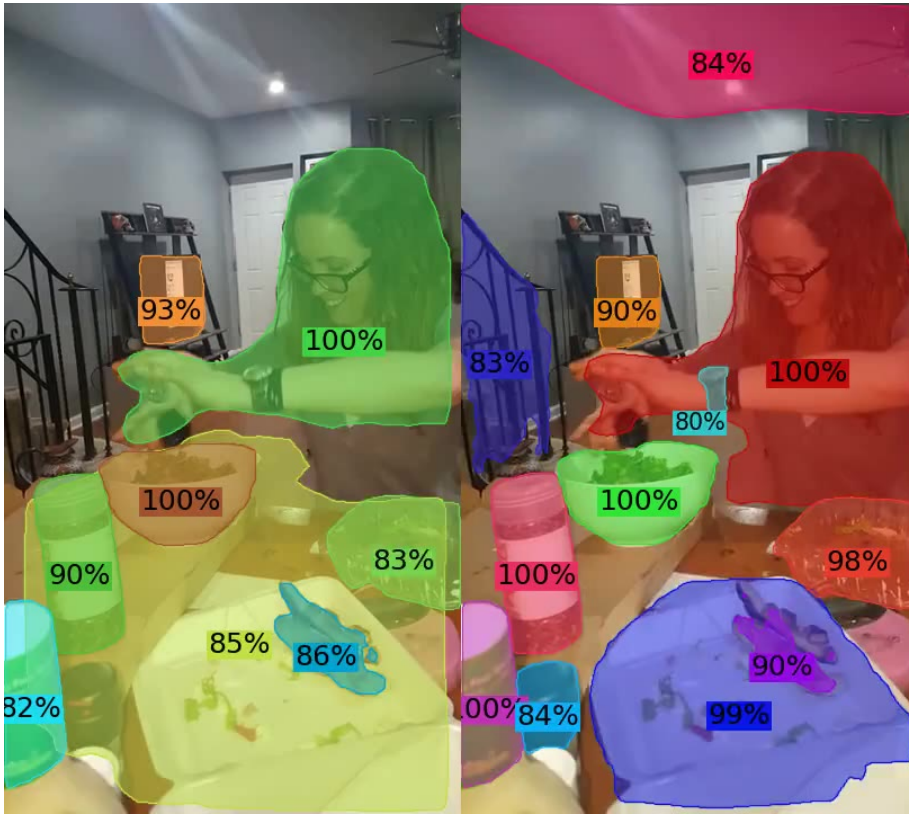


Fig. F: **Video demo of models trained on COCO. Left: Mask RCNN. Right: LDET.**

References

1. Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. IEEE TPAMI **34**(11), 2189–2202 (2012)
2. Arbeláez, P., Pont-Tuset, J., Barron, J.T., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR. pp. 328–335 (2014)
3. Bansal, A., Sikka, K., Sharma, G., Chellappa, R., Divakaran, A.: Zero-shot object detection. In: ECCV. pp. 384–400 (2018)
4. Bendale, A., Boult, T.E.: Towards open set deep networks. In: CVPR. pp. 1563–1572 (2016)
5. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: CVPR. pp. 6154–6162 (2018)
6. Cen, J., Yun, P., Cai, J., Wang, M.Y., Liu, M.: Deep metric learning for open world semantic segmentation. In: ICCV (2021)
7. Chen, X., Girshick, R., He, K., Dollar, P.: Tensormask: A foundation for dense object segmentation. In: ICCV (2019)
8. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: CVPR. pp. 3606–3613 (2014)
9. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
10. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV. pp. 764–773 (2017)
11. Dhamija, A., Gunther, M., Ventura, J., Boult, T.: The overlooked elephant of object detection: Open set. In: WACV. pp. 1021–1030 (2020)
12. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: ICCV. pp. 2758–2766 (2015)
13. Dvornik, N., Mairal, J., Schmid, C.: Modeling visual context is key to augmenting object detection datasets. In: ECCV. pp. 364–380 (2018)
14. Dwibedi, D., Misra, I., Hebert, M.: Cut, paste and learn: Surprisingly easy synthesis for instance detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1301–1310 (2017)
15. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88**(2), 303–338 (2010)
16. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.: Cascade object detection with deformable part models. In: CVPR. pp. 2241–2248. Ieee (2010)
17. Ghiasi, G., Cui, Y., Srinivas, A., Qian, R., Lin, T.Y., Cubuk, E.D., Le, Q.V., Zoph, B.: Simple copy-paste is a strong data augmentation method for instance segmentation. In: CVPR. pp. 2918–2928 (2021)
18. Girshick, R.: Fast r-cnn. In: ICCV. pp. 1440–1448 (2015)
19. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. pp. 580–587 (2014)
20. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
22. Hu, R., Dollár, P., He, K., Darrell, T., Girshick, R.: Learning to segment every thing. In: CVPR. pp. 4233–4241 (2018)
23. Jaiswal, A., Wu, Y., Natarajan, P., Natarajan, P.: Class-agnostic object detection. In: WACV. pp. 919–928 (2021)

24. Joseph, K., Khan, S., Khan, F.S., Balasubramanian, V.N.: Towards open world object detection. In: CVPR. pp. 5830–5840 (2021)
25. Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., Darrell, T.: Few-shot object detection via feature reweighting. In: ICCV. pp. 8420–8429 (2019)
26. Kim, D., Lin, T.Y., Angelova, A., Kweon, I.S., Kuo, W.: Learning open-world object proposals without learning to classify. arXiv preprint arXiv:2108.06753 (2021)
27. Kuo, W., Angelova, A., Malik, J., Lin, T.Y.: Shapemask: Learning to segment novel objects by refining shape priors. In: ICCV. pp. 9207–9216 (2019)
28. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. pp. 2117–2125 (2017)
29. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2980–2988 (2017)
30. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
31. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016)
32. Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., Yu, S.X.: Large-scale long-tailed recognition in an open world. In: CVPR. pp. 2537–2546 (2019)
33. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: ICCV. pp. 4990–4999 (2017)
34. Perera, P., Morariu, V.I., Jain, R., Manjunatha, V., Wigington, C., Ordonez, V., Patel, V.M.: Generative-discriminative feature representations for open-set recognition. In: CVPR. pp. 11814–11823 (2020)
35. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR (2016)
36. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS (2015)
37. Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: ICCV. pp. 8430–8439 (2019)
38. Tack, J., Mo, S., Jeong, J., Shin, J.: Csi: Novelty detection via contrastive learning on distributionally shifted instances. arXiv preprint arXiv:2007.08176 (2020)
39. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV **104**(2), 154–171 (2013)
40. Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., Schmid, C.: Learning from synthetic humans. In: CVPR (2017)
41. Wang, W., Feiszli, M., Wang, H., Tran, D.: Unidentified video objects: A benchmark for dense, open-world segmentation. arXiv preprint arXiv:2104.04691 (2021)
42. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. `https://github.com/facebookresearch/detectron2` (2019)
43. Yoshihashi, R., Shao, W., Kawakami, R., You, S., Iida, M., Naemura, T.: Classification-reconstruction learning for open-set recognition. In: CVPR. pp. 4016–4025 (2019)
44. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR. pp. 633–641 (2017)
45. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV. pp. 391–405. Springer (2014)