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

Figure 1. **Mask RCNN (top row) detects fewer objects than our approach (bottom row) in open-world instance segmentation.** In this task, the model must locate and segment all objects in the image irrespective of categories used for training. Here both detectors are trained on COCO [34] and tested on UVO [48]. Our detector correctly localizes many objects that are not labeled in COCO with the help of a new data augmentation method 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 the unannotated 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—background objects that are visible but unlabeled—we paste 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 decouple the training into two parts: 1) training the region classification and regression head on augmented images, and 2) training the mask heads on original images. In this way, a model does not learn to classify hidden objects as background while generalizing well 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 and Cityscapes.

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. Unlike humans, current state-of-the-art detection and segmentation methods [22, 33, 35, 40, 41] have difficulty recognizing novel objects as *objects* because they 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 [17, 34, 51] 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 [48] aims to localize and segment novel objects, but the state-of-the-art model does not perform well as shown in [48].

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 overlap little with the annotated 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 de-

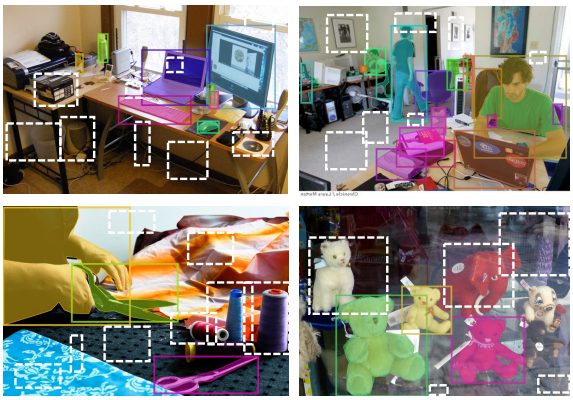

Figure 2. **Problem in a standard training of object detector. Examples are from COCO.** Colored boxes are annotated boxes while white-dashed boxes are potential background regions. Many white-dashed regions locate objects, but are regarded as background during training in a conventional way of training object detector. This can suppress the objectness of novel objects.

tect them, which prevents the models from learning general objectness. To address this, Kim *et al*. [29] 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-set 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 barely performs well. 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. In training the classification head, there is little risk of treating hidden objects as background since they are removed in the synthesized images. In addition, since the mask heads are trained to segment instances in real images, the backbone learns generalizable representations to separate foreground from background regions in real images.

Despite being seemingly a minor change, LDET demonstrates remarkable gains in open-world instance segmentation and detection. On COCO [34], LDET trained on VOC categories improves the average recall by 14.1 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 [48]. 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.
- 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 and Cityscape-to-Mapillary.

2. Related Work

Region proposals. Unsupervised region proposal generation used to be a standard approach to localize objects in a scene [1, 2, 46, 52]. 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 [18, 20, 21, 35, 40, 41]. The ability to detect known objects has been improving with better architecture designs [5, 11, 32] or objectives [33]. Also, localizing objects given a few training examples or semantic information is becoming a popular research topic [3, 28]. 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, 7, 13]. 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, 36, 38, 45, 50]. 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 [48] during evaluation.

Wang *et al*. [48] recently published the first benchmark dataset for open-set instance segmentation, which includes

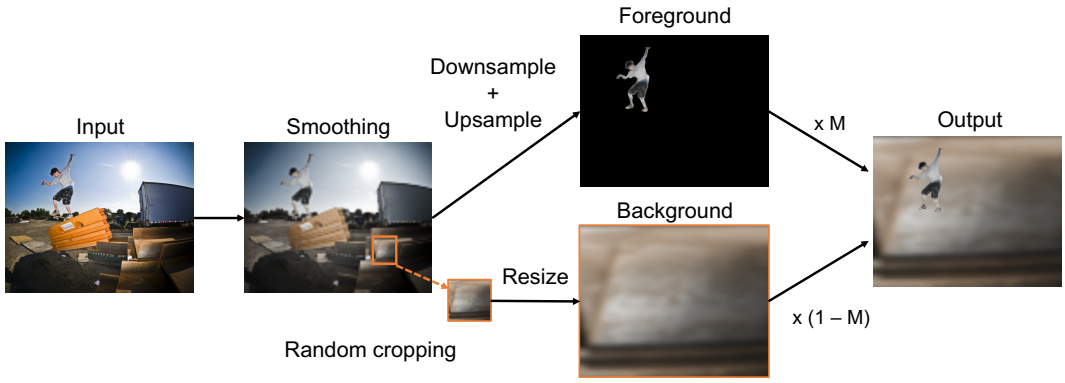

Figure 3. **Our augmentation strategy** creates images without hidden objects by upscaling small regions to use as background.

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*. [24, 31] proposed approaches for predicting masks of various objects, but they require bounding boxes from classes of interest. Jaiswal *et al*. [26] trained a detector in an adversarial manner to learn class-invariant objectness. Joseph *et al*. [27] 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*. [29] employed localization quality estimation with the claim that the estimation strategy is more generalizable in open-world instance segmentation.

The core of the open-world detection problem, as pointed out by Kim *et al*. [29], 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 [14, 19, 47]. Recently, copy-and-paste augmentation was shown to be a very useful technique in instance segmentation [15, 16, 19]. Dwibedi *et al*. [16] 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*. [15] considered modeling the visual context to paste the objects while Ghiasi *et al*. [19] 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 [16]. 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. Given an instance segmentation model, we train the object detection loss on images with synthesized background, and the instance mask loss on real images. Mask-RCNN [22] serves as the base model. We describe details of the data generation process (Fig. 3) and training scheme (Fig. 5) below.

3.1. Data Augmentation: Erasing Background

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. Then, we randomly sample a small unannotated region from I_1 and resize it to the same size as the input image to serve as a background canvas, which we denote as I_2. By smoothing the whole image before this operation, we expect to reduce the discrepancy in high-frequency content between the foreground and background images. We set the width and height of the background region to be $\frac{1}{8}$ of the original image's. Cropping the 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 totally different, as shown in examples in Fig. 4. We vary the scale of the background canvas in experiments but find the difference in performance to be not very significant (see Table 5).

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

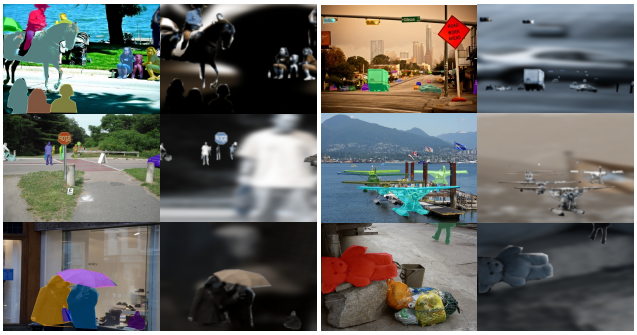

Figure 4. **Examples of original inputs (left) and synthesized image (right).** Masked regions are highlighed with colors. Using small regions as background avoids the risk of having hidden objects in the background. In some cases, the background patch happens to locate foreground objects (second row in the left column). Note that this case is rare, and the patch looks different from the original because it is significantly upscaled.

objects 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 Training

Simply training a detector on the synthesized images in the conventional way [22] does not work well due to the domain shift (See Table 2). 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 of domain shift.

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 [22], 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 summarize the objectives for RPN and ROI as one loss. The instance mask loss trains the mask head using only positive regions, *e.g.*, IoU with the nearest ground truth > 0.5 by default in Mask RCNN.

We propose to bridge the gap between two domains, *i.e.*, synthesized images and real images, by computing instance mask loss on real images while employing synthesized images only to compute proposal classification loss and localization regression loss. The mask loss encourages a model to separate background and foreground pixels given a bounding box, wherein only well-localizing bounding boxes are selected to compute the loss. Thus, the model will not learn to suppress hidden objects by using the mask loss. Furthermore, the mask loss training signal is propagated to a backbone network shared among the region proposal network, bounding box head, and mask head, which makes the box head adapted to real images. Since the box head and proposal network are trained only with the blended images without background objects, they excel at detecting novel objects.

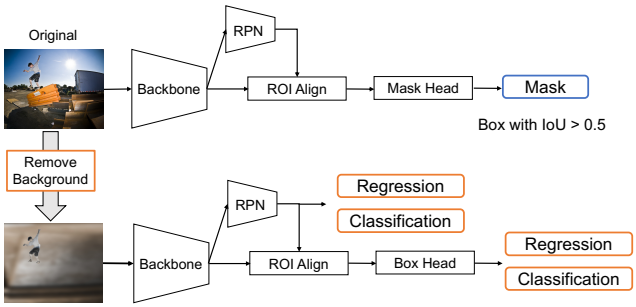

Figure 5. **Training pipeline.** Given an original input image and synthesized image, we train the detector on the mask loss computed on the original image and classification, regression loss on the synthesized image.

Foreground and background region sampling. During training, Mask RCNN (or Faster RCNN) decides foreground and background regions by setting a threshold on the value of IoU, *e.g.*, regions with IoU smaller/larger than 0.5 are background/foreground respectively in the case of ROI head. When balancing the number of two regions, boxes are randomly sampled. Note that we use the same thresholds as default Mask RCNN. While localization quality estimation [29] requires carefully selecting the thresholds during training, LDET does not.

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 [34] dataset, where we split annotated classes into known and unknown classes, train models on known ones, and evaluate detection/segmentation performance on unknown ones. 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 [34] or Cityscapes [10] as a training source, with UVO [48] and Mappilary Vista [37] being the test datasets, respectively. In this work, Average Precision (AP) and Average Recall (AR) are employed for

Method	Box				Mask			
	AP	AR10	AR30	AR100	AP	AR10	AR30	AR100
OLN-Mask [29]	-	18.3	-	-	-	16.9	-	-
OLN-Faster [29]	3.7	17.9	26.2	32.8	-	-	-	-
Mask RCNN [22]	8.9	16.0	19.3	20.9	7.2	13.7	16.4	17.7
Mask RCNN^P	7.1	16.0	19.1	20.7	5.4	13.8	16.3	17.7
Mask RCNN^S	8.3	17.4	22.6	27.1	6.0	15.1	19.7	23.7
LDET	(+1.3) **10.2**	(+5.5) **21.5**	(+9.2) **28.5**	(+13.9) **34.8**	(+1.8) **9.0**	(+6.0) **19.7**	(+9.2) **25.6**	(+13.3) **31.0**

Table 1. **Results of VOC → Non-VOC generalization in COCO**. 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. The scores of OLN-Mask are reported in [29], those of OLN-Faster are obtained by evaluating a published model.

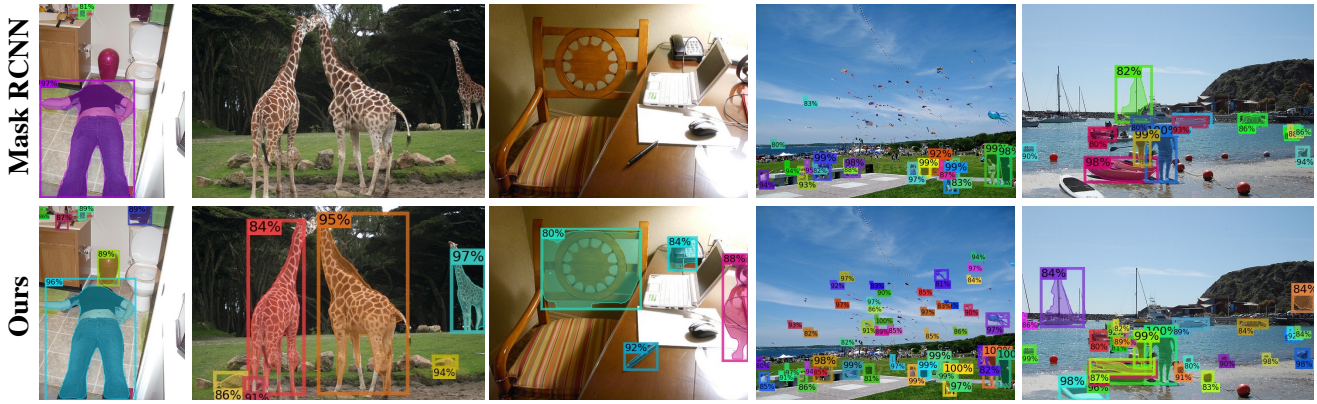

Figure 6. **Visualization in VOC to Non-VOC in COCO dataset. Top: Mask RCNN. Bottom: LDET.** Note that training categories do not include giraffe, trash box, pen, kite, and floats. LDET detects many novel objects better than Mask RCNN.

performance evaluation. The evaluation is done in a class agnostic way. Unless otherwise specified, AR and AP are computed following the COCO evaluation protocol, *i.e.*, AP or AR at 100 detections at maximum.

Implementation. Detectron2 [49] is used to implement LDET. Mask R-CNN [22] with ResNet-50 [23] as feature pyramid [32] backbone is used unless otherwise specified. We utilize the standard hyperparameters of Mask R-CNN [22] as defined in Detectron2. See appendix for more details. 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 [27], 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 score. A model is trained to minimize the loss on pseudo-labels.

4) OLN. OLN [29] is a concurrent work for open-world instance segmentation. We adopt it for comparsions with our baseline models in the setting where they report results or publish trained models.

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. One way is to train a model in a class agnostic manner, *i.e.*, regarding all classes as one *foreground* class, but this makes a model unable to infer the class of detected objects. To achieve more flexible prediction, 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. We also compare class-agnostic inference between a class-discriminative model and a class-agnostic model.

Real	Synth	Decoupled Training	Box			Mask		
			AP	AR10	AR100	AP	AR10	AR100
✓			8.9	16.0	20.9	7.2	13.7	17.7
	✓		0.2	1.5	4.7	0.1	0.4	0.7
✓	✓		8.9	16.5	21.9	7.1	14.1	18.6
✓	✓	✓	**10.2**	**21.5**	**34.8**	**9.0**	**19.7**	**31.0**

Table 2. **Ablation study of data and training method in VOC → Non-VOC.** The last row is our proposed framework.

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 and test on Non-VOC classes only. The validation split of COCO is adopted for evaluation. Following [29], we do not count the "seen class" detection boxes into the budget k when computing the Average Recall (AR@k) scores. 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. It can generate precise boxes as well as detect many novel objects. OLN-Faster [29] demonstrates strong performance in AR, but performs poorly in AP because it generates many false positives. Similarly, Mask RCNN^S improves performance on AR while degrading AP. Both of these baselines are thresholding background regions based on IoU with ground truth boxes, which may result in unbalanced sampling with respect to the box size. For example, if we ignore background boxes with small IoU, small regions are unlikely to become negatives and the proposal network mistakenly detects many small background regions as foreground. Mask RCNN^P does not improve on Mask RCNN. Some visualiations are illustrated 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*.

Ablation study for learning objectives. Table 2 describes an ablation study of training data and objectives. If a model learns only from synthetic data (second row), it fails to detect and segment objects. If a model is trained simply by using both real and synthetic images, *i.e.*, using all losses on both domains, it shows slight improvements compared to Real Only. These results conclude that our proposed decoupled training is very suitable for the open-world instance segmentation and detection task.

Class agnostic vs. class discriminative learning. Table 3 shows that the class agnostic training slighly improves performance in the case of both plain Mask RCNN and LDET. The selection of the training method should depend on the importance of the small gain and class discriminative prediction in an applied recognition system.

Mask RCNN overfits to seen classes. Fig. 7 illustrates the training iteration and corresponding AR in MaskRCNN and LDET. AR gradually degrades in MaskRCNN probably because the model fits to training data and learns to ignore hidden objects. By contrast, LDET demonstrates a solid improvement in AR as the training progresses. This implies that LDET can utilize the objectness of the pretrained representation and thus generalize to more than the training categories. We investigate this point in the appendix.

Method	Agnostic	Box			Mask		
		AP	AR10	AR100	AP	AR10	AR100
Plain		8.7	16.0	20.9	7.0	13.7	17.7
Plain	✓	**9.0**	**18.7**	**21.6**	**7.3**	**13.9**	**18.1**
LDET		**10.2**	21.5	34.9	9.0	19.7	31.0
LDET	✓	**10.2**	**21.8**	**35.6**	**9.2**	**20.2**	**32.0**

Table 3. **Ablation study for class agnostic training.** Class agnostic training slightly improves performance in both LDET and the plain Mask RCNN.

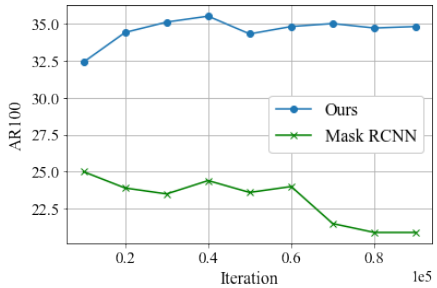

Figure 7. **Baseline MaskRCNN suffers from overfitting to annotated instances.** It thus decreases the performance to detect novel objects as training proceeds. In contrast, our approach basically gains performance with training iterations.

Method	Box		Mask	
	AR10	AR100	AR10	AR100
Geodesic [30]	4.0	18.0	2.3.	12.3
Rigor [25]	-	13.3	-	9.4
SelectiveSearch [46]	5.2	16.3	2.5	9.5
MCG [2]	10.1	24.6	7.7	18.6
DeepMask [39]	15.2	30.6	12.3	23.3
LDET	**30.3**	**46.0**	**27.9**	**40.8**

Table 4. **Comparison to unsupervised methods and DeepMask tested on COCO.** Results of baselines are reported in DeepMask [39]. Note that DeepMask uses VGG [44] as backbone. LDET and DeepMask are trained on VOC-COCO.

Comparison with hand-crafted methods. We compare LDET to the learning-free methods in instance segmentation and detection Table 4. We measure the performance of detecting all objects in the COCO validation set. LDET is superior to these baselines by a large margin.

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. While Mask RCNN suppresses the score for many objects, LDET correctly captures objectness for diverse objects. For instance, it captures objectness well in an image

Background ratio	AP_{box}	AR_{box}	AP_{mask}	AR_{mask}
2^{-1}	**10.8**	32.5	**9.0**	28.9
2^{-2}	10.5	32.0	**9.0**	28.4
2^{-3}	10.2	34.8	**9.0**	31.0
2^{-4}	9.6	**35.2**	8.8	**31.9**

Table 5. **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 and degrade AP.

Method	Backbone	Box		Mask	
		AP	AR100	AP	AR100
MRCNN	Res50	24.8	47.7	19.1	38.1
MRCNN	Res101	(+1.0) 25.8	(-0.1) 47.6	(+1.2) 20.3	(+0.6) 38.7
LDET	Res50	26.1	53.1	21.2	43.0
LDET	Res101	(+1.7) 27.8	(+0.1) 53.2	(+1.9) 23.1	(+1.1) 44.1

Table 6. **ResNet50 vs ResNet101.** ResNet101 tends to perform better than ResNet50, which is more evident in LDET than in Mask RCNN (MRCNN).

crowded with objects in the first row.

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 5. AR gets better with smaller sizes, while AP gets slightly worse. This indicates that smaller backgrounds prevent sampling hidden objects and coverage of novel objects (AR) becomes better. Also, note that the performance is stable with respect to the size.

External data for background. Using an external background dataset is an alternative way to synthesize background region although an image without background is not always accessible. In this experiment, we use DTD [9] (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 AP and AR are 11.1 (LDET$+0.9$) and 32.6 (LDET-2.2) respectively 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. Our data augmentation method is comparable or better than the use of external background images.

Different backbones. We employ ResNet101 [23] as a backbone in Table 6 and compare it with ResNet50. LDET benefits more from the deeper backbone.

Region proposal network (RPN) and region of interest (ROI) head. We compare the performance of RPN and ROI heads in Table 7. In Mask RCNN, RPN covers more novel objects while ROI provides precise boxes compared to RPN. In contrast, ROI of LDET performs better than RPN in all metrics.

Method	Detector	AP	AR10	AR100
MRCNN	RPN	5.9	15.4	27.3
MRCNN	ROI	(+3.0) 8.9	(+0.6) 16.0	(-6.4) 20.9
LDET	RPN	6.5	18.2	32.8
LDET	ROI	(+3.7) 10.2	(+3.3) 21.5	(+2.0) 34.8

Table 7. **Comparison between region proposal network and region of interest head. AP and AR of bounding boxes.**

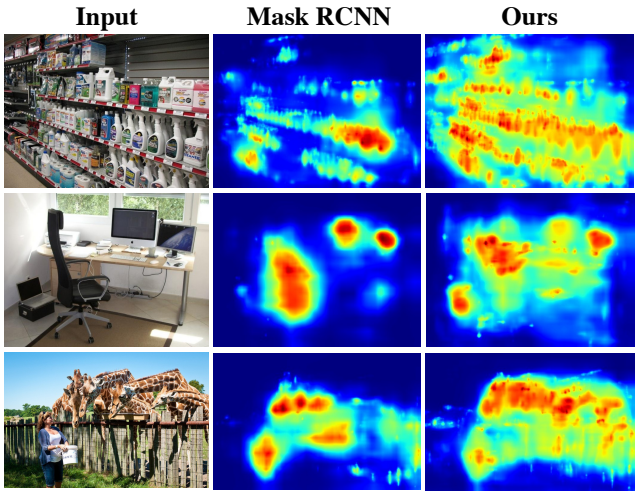

Figure 8. **Objectness map (RPN score) visualization in COCO experiment.** LDET captures objectness of various categories whereas Mask RCNN tends to suppress many objects.

4.2. Cross-dataset generalization

COCO to UVO. First, we utilize UVO [48], 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 of UVO is used to measure the performance.

As shown in Table 8, 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 imbalanced sampling with regions with different scales.

Cityscape to Mapillary. Second, we examine performance in autonomous driving scenes. Detectors are trained on Cityscape [10] (8 foreground classes, *person, rider, car, truck, bus, train, motorcycle,* and *bicycle*) and tested on the validation set of Mapillary Vistas [37] with 35 foreground classes including not only vehicles, but also animals, *trash can, mailbox*, and so on.

Method	Train	Box					Mask				
		AP	AR	AR_{small}	AR_{med}	AR_{large}	AP	AR	AR_{small}	AR_{med}	AR_{large}
OLN-Faster [29]	VOC (COCO)	13.3	47.6	**25.6**	**43.5**	57.8	-	-	-	-	-
Mask RCNN		19.6	34.8	10.9	23.9	50.8	14.6	25.8	9.2	19.4	36.1
Mask RCNN^P		19.0	34.8	10.7	23.7	51.0	14.5	26.0	9.3	19.4	36.4
Mask RCNN^S		18.9	39.2	12.8	28.8	55.6	12.9	29.8	11.7	23.9	40.1
LDET		**22.8**	**48.9**	23.1	42.4	**62.1**	**17.9**	**38.0**	**20.8**	**35.2**	**45.6**
Mask RCNN	COCO	24.8	47.7	21.9	41.7	60.6	19.1	38.1	19.1	35.0	46.5
Mask RCNN^P		23.8	47.2	22.1	41.1	59.9	18.8	37.6	19.1	34.6	45.7
Mask RCNN^S		21.3	47.9	21.1	40.8	62.0	15.6	38.6	19.3	34.9	47.6
LDET		**26.1**	**53.1**	**28.4**	**47.8**	**65.0**	**21.2**	**43.0**	**25.7**	**41.1**	**49.9**

Table 8. **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.

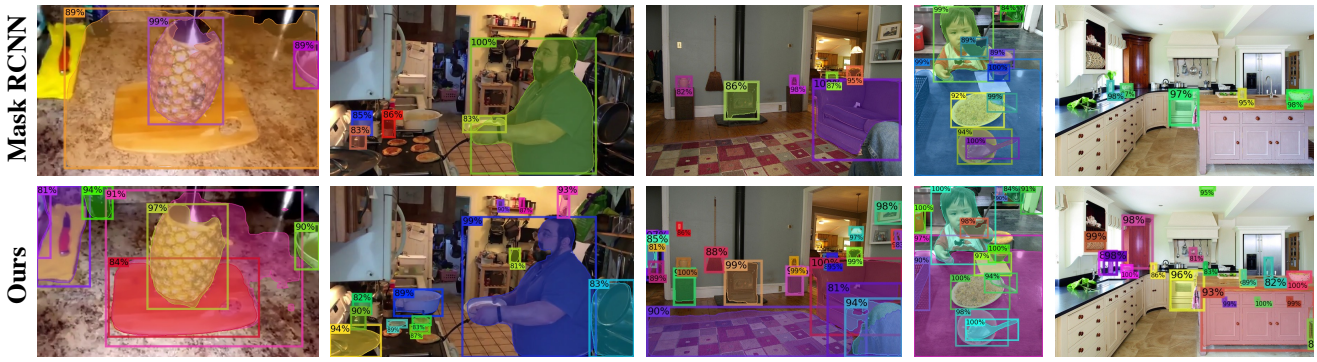

Figure 9. **Visualization of results for models trained on COCO.** Top: Mask RCNN. Bottom: LDET. Leftmost two images are from UVO, the others are from COCO validation images.

Method	Box			Mask		
	AP	AR	$AR_{0.5}$	AP	AR	$AR_{0.5}$
Mask RCNN	8.3	9.7	14.6	7.3	8.4	13.5
Mask RCNN^P	8.2	9.7	14.5	7.2	8.4	13.4
Mask RCNN^S	6.7	9.6	14.7	6.1	8.7	13.9
LDET	**8.5**	**11.4**	**19.0**	**7.7**	**10.0**	**16.8**

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

In Table 9, LDET shows solid 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. We acknowledge that this is a limitation of LDET and discuss this further in Sec. 5. 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.

Potential negative societal impact. Some applications of object detection may have a negative impact on society, such as excessive surveillance and inappropriate policing. Privacy is another concern. We note that our approach can help remove accidental people in the background of training images and thus improve privacy.

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. See our code for more details.

Implementation. We use the default hyper-parameters, provided by Detectron2, to train LDET, Mask RCNN, Mask RCNN^S, and Mask RCNN^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.

Test time hyper-parameters. Following [29], we set hyper-parameters in testing as follows: threshold of non-maximum suppression in the roi head is set as 0.7 (0.5 by default) and, in the region proposal network, the number of region proposals kept after the suppression is set as 2000 (1000 by default). Table A shows the performance difference by the hyper-parameters. AP tends to improve by decreasing the threshold value and degrade by increasing the number of proposals. Low threshold suppresses more boxes, which results in improving AP and degrading AR. Changing the number of proposals has a similar effect. We can observe that the difference is not significant.

NMS Threshold	Num of Boxes	Box		Mask	
		AP	AR	AP	AR
0.5	1000	10.5	33.3	9.3	30.3
0.7	1000	10.3	34.9	9.1	31.0
0.5	2000	10.5	33.6	9.3	30.7
0.7	2000	10.2	34.8	9.0	31.0

Table A. **Study on the threshold for no-maximum suppression and the number of boxes kept after region proposal network.** The last row is the setting used in the main paper.

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.

[1] https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_1x.yaml

[2] https://github.com/facebookresearch/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)
```

Experiments on texture dataset. To make a background using images of DTD [9], 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

In this section, we provide some additional analysis of 1) model pre-training and 2) using a one-stage detector.

Pre-training the model. In the experiment in Sec. 4, "Mask RCNN overfits to seen classes", we saw that the plain Mask RCNN model seems to be overfitting to the training data, while LDET generalizes better to novel objects. We hypothesize that LDET is better able to use the representations (features) of objects contained in the pre-trained backbone. In this experiment, we explore this further and provide results of different pre-training methods and data. Specifically, we explore whether pre-training on supervised classification data gives our method a boost compared to unsupervised or no pretraining.

First, we train a model from scratch on the VOC-COCO data. Second, we use a self-supervised learning model,

Method	Pre-training Data	Pre-training Method	VOC-to-COCO				VOC-to-UVO			
			Box		Mask		Box		Mask	
			AP	AR	AP	AR	AP	AR	AP	AR
MRCNN	None	None	6.1	24.9	4.6	21.5	16.6	**37.8**	11.3	**27.7**
MRCNN	ImageNet1K	SWAV [6]	7.1	**25.3**	5.5	**22.1**	15.8	37.1	10.8	26.5
MRCNN	ImageNet1K	Supervised	**8.9**	20.9	**7.2**	17.7	**19.6**	34.8	**14.6**	25.8
MRCNN	ImageNet21K	MIIL [42]	8.8	20.9	7.0	17.9	19.3	35.0	14.4	25.9
LDET	None	None	5.7	29.5	4.2	25.2	15.8	42.7	9.9	30.1
LDET	ImageNet1K	SWAV [6]	5.5	29.9	4.7	27.1	13.9	41.2	10.6	30.8
LDET	ImageNet1K	Supervised	**10.2**	34.8	**9.0**	31.0	**22.8**	**48.9**	**17.9**	**38.0**
LDET	ImageNet21K	MIIL [42]	9.7	**35.3**	8.6	**31.9**	21.3	48.1	17.1	37.6

Table B. **Comparison of pre-training methods.** Replacing supervised pre-trained model with SWAV (unsupervised) pre-training degrades AP in both training methods. SWAV performs better on AR of Mask RCNN.

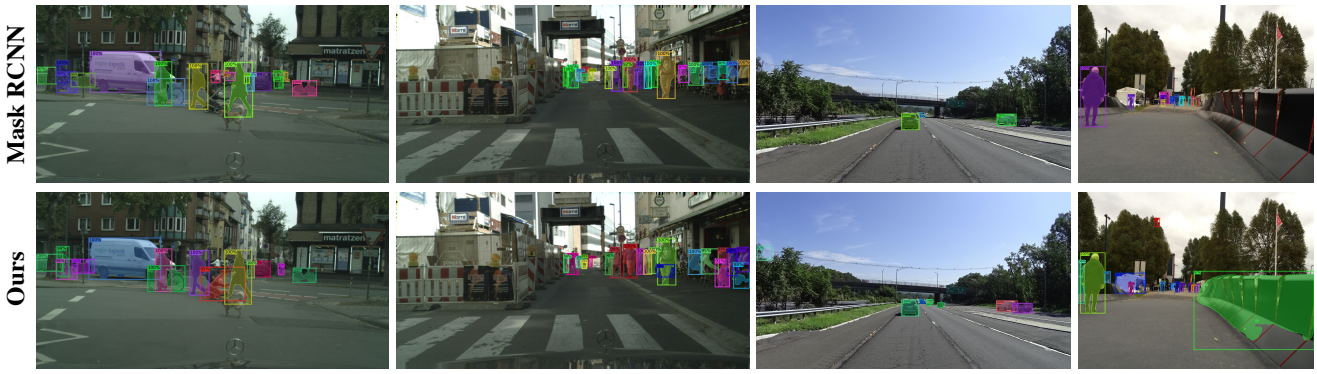

Figure A. **Visualization for detectors trained on Cityscapes.** Leftmost two images are validation images of Cityscapes, rightmost two are from Mapillary.

Method	Box				
	AP	AR10	AR30	AR50	AR100
RetinaNet [33]	**9.0**	16.5	21.0	22.9	25.0
LDET	6.8	**18.9**	**26.1**	**29.2**	**32.9**

Table C. **Results of VOC → Non-VOC generalization in COCO using RetinaNet as a detector.**

SWAV [6], trained on ImageNet1k [43] to initialize models. Third, we use the default model, trained on ImageNet1k in a supervised manner. Finally, a model trained on ImageNet21k [12] with a method that considers the hierarchies of the categories is employed. We expect that comparison on these pre-trained models will show the importance of 1) class-discriminative features and 2) the number of categories used for pre-training.

Table B provides the results in the VOC-COCO setting. First, we see that supervised pre-training improves AP in all cases. Second, in Mask RCNN, unsupervised/no pre-training gives better AR than supervised pre-training. By contrast, in LDET, supervised pre-training outperforms unsupervised/no pre-training in both AP and AR. This implies that LDET harnesses the class discriminative features learned by supervised training better than the standard detector training does. Finally, we do not see a clear difference in the performance between ImageNet21k and ImageNet1k, neither in Mask RCNN nor LDET. Note that top-1 accuracy on ImageNet1k is 75.3 (Supervised) and 82.0 (MIIL) respectively. Better accuracy on ImageNet1k does not necessarily lead to better performance in open-world instance segmentation. Also, increasing pre-training categories beyond 1k does not show improvement in this experiment.

One-stage detector. The experiments in the main draft are done for a two-stage detector, Mask RCNN. In this analysis, we apply LDET to one-stage detectors and see its behavior.

Table C shows the results on RetinaNet [33]. Since the standard RetinaNet does not have an instance mask head, the mask head is attached on top of the feature pyramid of the detector. The architecture of the head is the same as Mask RCNN. Therefore, the detection loss *e.g.*, localization and classification loss, is computed for synthesized images, whereas the mask loss is calculated for real images. LDET outperforms the plain model with a large margin in AR while we see a degradation in AP. To maintain the precision, it may be necessary to tune some hyper-parameters

Method	Box				Mask			
	AP	AR10	AR30	AR100	AP	AR10	AR30	AR100
TensorMask [8]	**9.6**	17.8	22.9	26.6	**7.4**	14.8	18.8	22.0
LDET	(-0.8) 8.8	(+3.0) **20.8**	(+4.9) **27.8**	(+7.8)**34.4**	(-0.4) 7.0	(+3.4) **18.2**	(+5.3) **24.1**	(+7.7) **29.7**

Table D. **Results of Tensormask for VOC → Non-VOC generalization in COCO**.

such as the size of background region, or to tailor the training objective to the one-stage detector.

The results on Tensormask [8] are shown in Table D. Since the objectives of Tensormask consist of detection loss and mask loss, synthesized images are used to compute detection loss whereas real images are used to compute mask loss. The trend is similar to RetinaNet where AP slightly decreases whereas AR improves significantly. The empirical results of RetinaNet and Tensormask indicate the usefulness of our framework for one-stage detectors.

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. B and C are additional visualizations in VOC-COCO and COCO, respectively. Note that we add the results of Mask RCNN^S, which are not visualized in the main paper due to a limited space. Mask RCNN^S 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. D and E are demo of applying LDET to UVO [48] videos. Click the images to play the videos.

References

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. Measuring the objectness of image windows. *IEEE TPAMI*, 34(11):2189–2202, 2012. 2

[2] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *CVPR*, pages 328–335, 2014. 2, 6

[3] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, pages 384–400, 2018. 2

[4] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *CVPR*, pages 1563–1572, 2016. 2

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 2

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 10

[7] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Deep metric learning for open world semantic segmentation. In *ICCV*, 2021. 2

[8] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollar. Tensormask: A foundation for dense object segmentation. In *ICCV*, 2019. 11

[9] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014. 7, 9

[10] 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. 4, 7

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017. 2

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 10

[13] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *WACV*, pages 1021–1030, 2020. 2

[14] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015. 3

[15] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. In *ECCV*, pages 364–380, 2018. 3

[16] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1301–1310, 2017. 3

[17] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 1

[18] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *CVPR*, pages 2241–2248. Ieee, 2010. 2

[19] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, pages 2918–2928, 2021. 3

[20] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 2

[21] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 2

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1, 3, 4, 5

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 7

[24] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *CVPR*, pages 4233–4241, 2018. 3

[25] Ahmad Humayun, Fuxin Li, and James M Rehg. Rigor: Reusing inference in graph cuts for generating object regions. In *CVPR*, pages 336–343, 2014. 6

[26] Ayush Jaiswal, Yue Wu, Pradeep Natarajan, and Premkumar Natarajan. Class-agnostic object detection. In *WACV*, pages 919–928, 2021. 3

[27] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, pages 5830–5840, 2021. 3, 5

[28] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *ICCV*, pages 8420–8429, 2019. 2

[29] Dahun Kim, Tsung-Yi Lin, Anelia Angelova, In So Kweon, and Weicheng Kuo. Learning open-world object proposals without learning to classify. *arXiv preprint arXiv:2108.06753*, 2021. 2, 3, 4, 5, 6, 8, 9

[30] Philipp Krähenbühl and Vladlen Koltun. Geodesic object proposals. In *ECCV*, pages 725–739. Springer, 2014. 6

[31] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *ICCV*, pages 9207–9216, 2019. 3

[32] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 2, 5

[33] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 1, 2, 10

[34] 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. 1, 2, 4

[35] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 1, 2

[36] 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*, pages 2537–2546, 2019. 2

[37] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, pages 4990–4999, 2017. 4, 7

[38] Pramuditha Perera, Vlad I Morariu, Rajiv Jain, Varun Manjunatha, Curtis Wigington, Vicente Ordonez, and Vishal M Patel. Generative-discriminative feature representations for open-set recognition. In *CVPR*, pages 11814–11823, 2020. 2

[39] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. *arXiv preprint arXiv:1506.06204*, 2015. 6

[40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1, 2

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 2

[42] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021. 10

[43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 10

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[45] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *arXiv preprint arXiv:2007.08176*, 2020. 2

[46] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 2, 6

[47] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 3

[48] 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. 1, 2, 4, 7, 11

[49] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5

[50] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *CVPR*, pages 4016–4025, 2019. 2

[51] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 1

[52] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405. Springer, 2014. 2

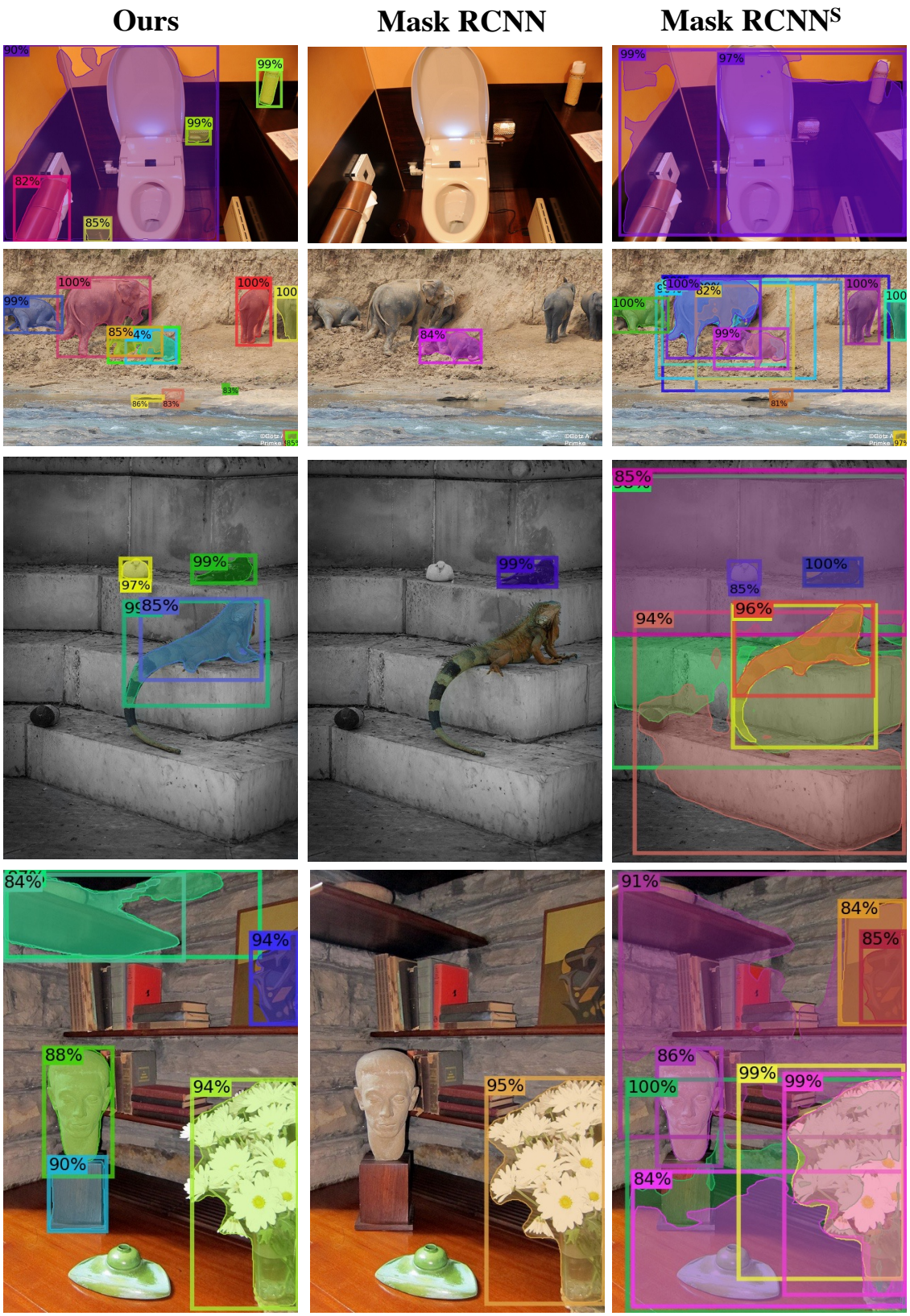

Figure B. **Visualization in VOC-COCO to COCO setting.** Note that VOC-COCO does not contain objects such as lizard, toilet paper, and elephant.

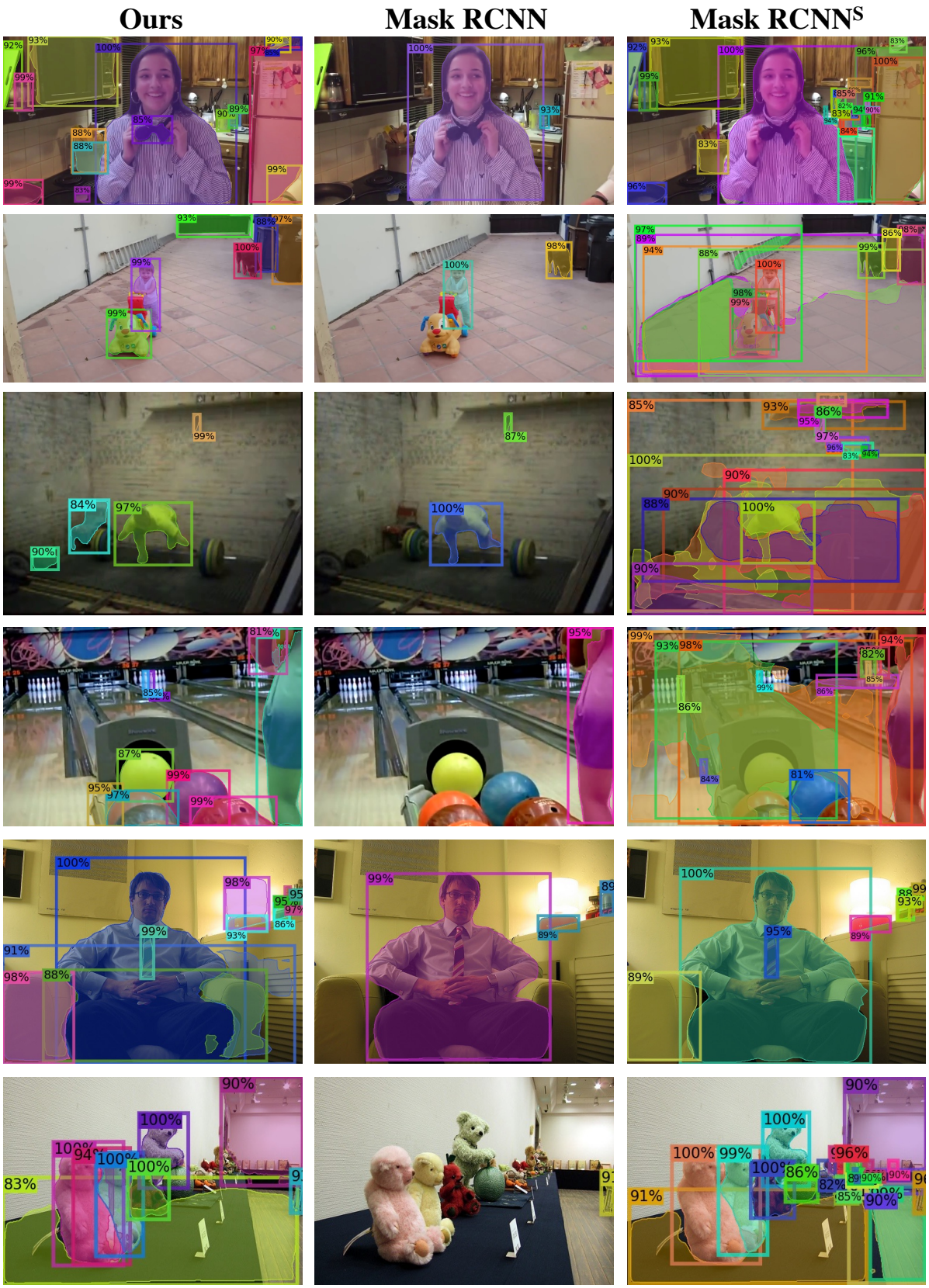

Figure C. **Visualization of models trained on COCO.** The images are from COCO and UVO.

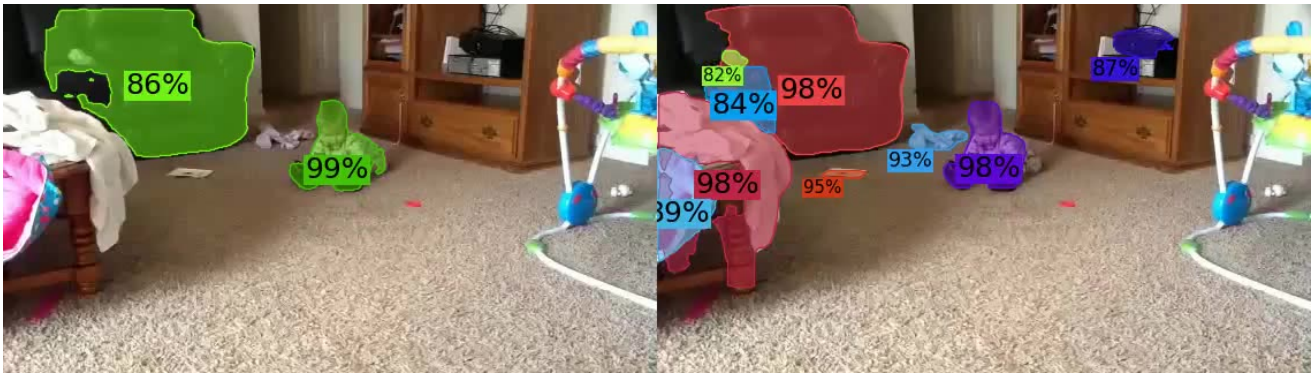

Figure D. **Video demo of models trained on COCO. Left: Mask RCNN. Right: LDET. Click the image to play the video.**

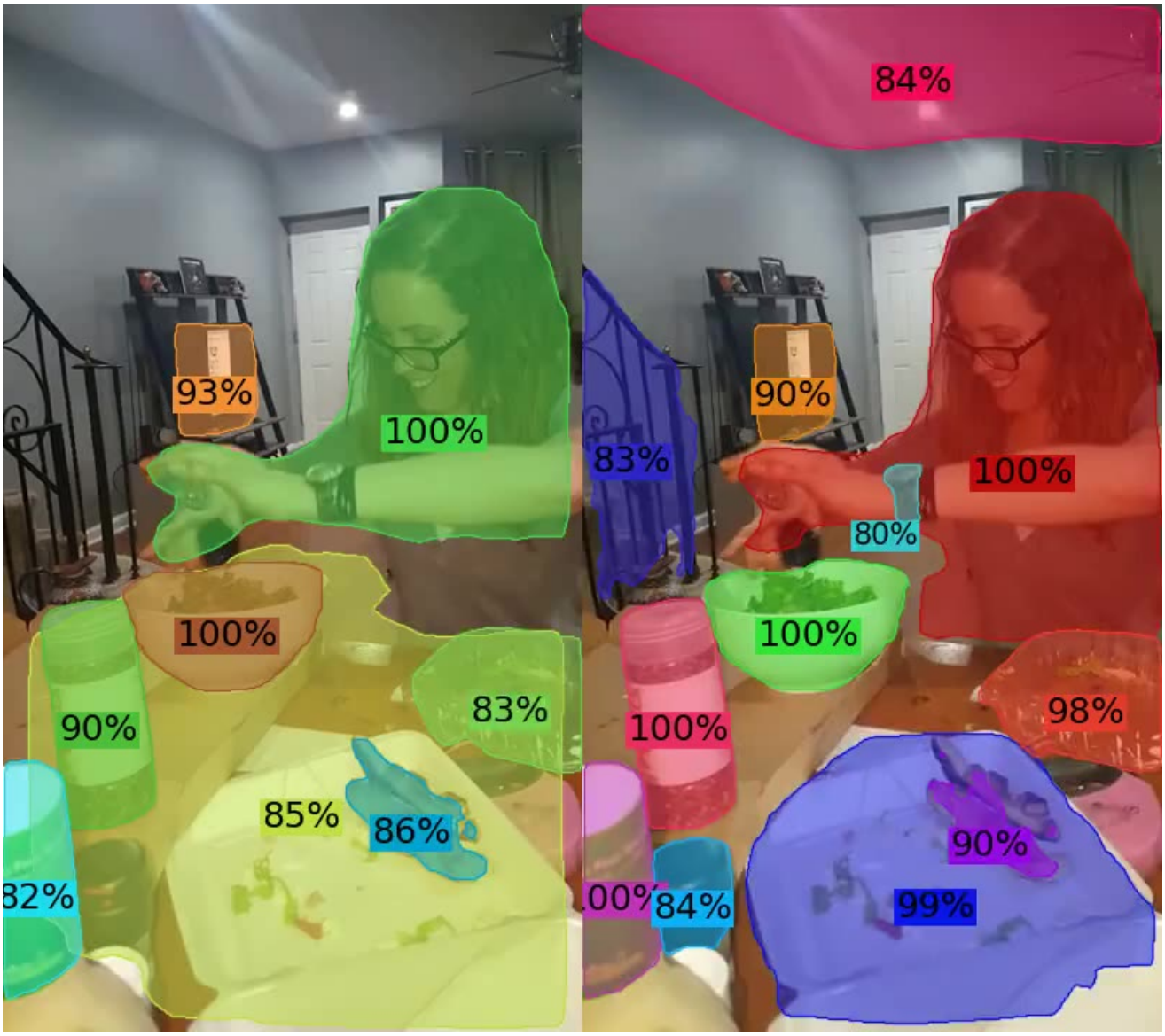

Figure E. **Video demo of models trained on COCO. Left: Mask RCNN. Right: LDET.Click the image to play the video.**