EMPIRICAL UPPER BOUND IN OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Object detection remains one of the most notorious open problems in computer vision. Despite large strides in accuracy and speed in recent years, modern object detectors have started to saturate on popular benchmarks. How far can we push the detection accuracy with the current deep learning tools and tricks? In this work, by employing two popular state-of-the-art object detection benchmarks, `MMDetection` and `Detectron2`, and analyzing more than 15 models over 4 large-scale datasets, we systematically determine the upper bound in AP, which is 91.6% on PASCAL VOC (*test2007*), 78.2% on MS COCO (*val2017*), and 58.9% on OpenImages (*V4 validation set*), regardless of the IOU. These numbers are much higher than the mAP of the best model (e.g., 58% on MS COCO according to the most recent results). Interestingly, the gap seems to be almost closed at IOU=0.5. We also analyze the role of context in object recognition and detection and find that the canonical object size leads to the best recognition accuracy. Finally, we carefully characterize the sources of errors in deep object detectors and find that classification error (confusion with other classes and misses) explains the largest fraction of errors and weighs more than localization error. Further, models frequently miss small objects, more often than medium and large ones. Our work taps into the tight relationship between object recognition and detection and offers insights to build better object detectors. Similar analyses can also be conducted for other tasks in computer vision such as for instance segmentation and object tracking. The code is available at [TBA].

1 INTRODUCTION AND MOTIVATION

Object recognition is believed to be solved in computer vision witnessed by the below human-level error rate of the state of the art models ($\sim$3% top-5 error on ImageNet *vs.* $\sim$5% human error rate; See Hu et al. (2018b) and Russakovsky et al. (2015)). Despite this so-called "superhuman" performance, deep object recognition models fail miserably on slightly transformed images (Szegedy et al., 2014; Goodfellow et al., 2014; Azulay & Weiss, 2018; Hendrycks & Dietterich, 2019). Unlike object recognition, however, object detection[1] remains largely unsolved (64% mean Average Precision at 75% overlap on COCO*val2017*; Fig. 1) and results are far below the theoretical upper bound (mAP=1). Object detection is much more challenging than object recognition not only because precise localization is needed but also because objects can undergo drastic transformations such as in-plane and in-depth rotation, blur, lighting, and partial occlusions. Also, there is a larger variation in object scale in detection datasets than recognition datasets[2].

Several years of active and extensive research on object detection has resulted in the accumulation of an overwhelming amount of knowledge regarding model backbones, tips and tricks for model training, optimization, data collection, augmentation, annotation, model evaluation, and comparison to a point that separating the wheat from the chaff is very difficult (Zou et al., 2019; Zhang et al., 2019). As an example, truly understanding and implementing average precision (AP) is frustratingly difficult (See Appendix A). A quick Google search returns several blogs and codes with discrepant explanations of AP. To make matters even worse, it is not quite clear whether AP has started to saturate, whether progress is significant, and more importantly how far we can improve following the current path, making one wonder maybe we have reached the peak in performance using deep learning. Further, we do not know exactly what is holding us back from making progress in object

[1]The best published results over COCO *test-dev2019* dataset are 61%, 79%, 68%, 74%, 64%, and 44% corresponding to AP, AP50, AP75, APl, APm, and APs, respectively. Please refer to `https://competitions.codalab.org/competitions/20794#results` for the latest results on COCO.

[2]The median scale of the object relative to the image in ImageNet *vs.* COCO is 554 and 106, respectively. Therefore, most object instances in COCO are smaller than 1% of the image area (Singh & Davis, 2018).

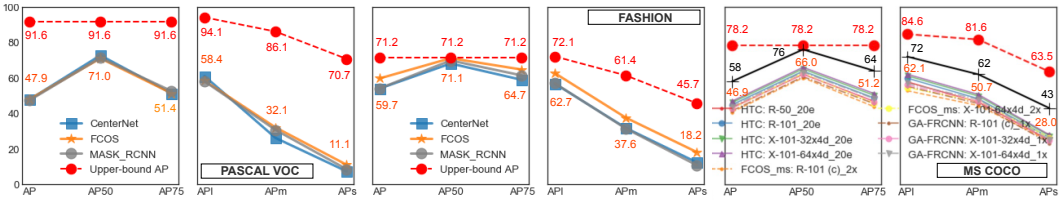


Figure 1: Upper bound AP (in red) and scores of the best models using the COCO evaluation tool (in orange; FCOS on VOC and FASHION datasets, and Hybrid Task Cascade on COCO). The black solid line on the COCO panel belongs to the winning entry on the latest (2019) challenge on COCO *val2019* provided at `https://competitions.codalab.org/competitions/20794#results` (this result was not available at the time of this study). See also `https://paperswithcode.com/sota/object-detection-on-coco-minival`. Notice that the gap at AP50 is almost closed on COCO. There is, however, still a large gap between the performance of the state of the art object detection models and the empirical upper bound. The gap is wider over higher IOU thresholds and small objects. See Fig. 11 in Appx D.

detection, compared to human-level (although debatable) accuracy in object recognition. As such, detection can be considered as a crucial task to assess the promises and limits of deep learning.

To shed light upon the above matters, we carefully and systematically approximate the empirical upper bound in AP. We argue that the upper bound AP (UAP) is the score of the best object recognition model that is trained on the training target bounding boxes and is then used to label the test target boxes (Sections 3 and 4). We investigate whether the visual context surrounding a target object or its overlapping bounding boxes can improve UAP, and how they impact object detection in general (Section 5). Finally, we identify the bottlenecks in deep object detectors by characterizing the type of errors they make and measure the impact of each error type on performance (Section 6). In a nutshell, we find that there is a large gap between the performance of the state of the art object detection models and the empirical upper bound as shown in Fig. 1. The gap is wider at higher IOU thresholds and over small objects. Interestingly, using the latest results on COCO dataset (Fig. 1), the gap is very narrow at IOU=0.5 ($\sim$2% absolute AP). This suggests that at least using the recent models, maybe classification is solved and focus should be shifted towards localization. This, however, needs to be investigated further with the newer error analysis tools (e.g., Bolya et al. (2020)). The computed empirical upper bound entails that there is a hope to reach this peak with the current tools if we can find smarter ways to adopt the object recognition models and backbones for object detection. Over the models that we analyzed here, it seems that recognition remains the major bottleneck in object detection and it is more critical over small objects. In other words, object detection models inherit the critical limitation of CNNs which is the lack of invariance to natural image corruptions and transformations (e.g., noise, blur, scale), as well as adversarial perturbations.

2 Related Work

We discuss three lines of related works. The first one includes **works that strive to understand detection approaches**, identify their shortcomings, and pinpoint where more research is needed. Parikh & Zitnick (2011) aimed to find the weakest links in person detectors by replacing different components in a pipeline (e.g., part detection, non-maxima-suppression) with human annotations. Mottaghi et al. (2015) proposed human-machine CRFs for identifying bottlenecks in scene understanding models. Hoiem et al. (2012) inspected detection models in terms of their localization errors and confusion with other classes and the background over the PASCAL VOC dataset. They also conducted a meta-analysis to measure the impact of object properties such as color, texture, and real-world size on detection performance. Such an analysis, however, has not been conducted over deep object detectors (done here). To overcome the shortcomings of the Hoiem et al. and COCO analysis tools, recently Bolya et al. (2020) proposed to analyze the models by emphasizing the order in which the errors are analyzed. Russakovsky et al. (2013) analyzed the ImageNet localization task and emphasized on fine-grained recognition. Zhang et al. (2016) measured how far we are from solving pedestrian detection. Vondrick et al. (2013) proposed a method for visualizing object detection features to gain insights into their functioning. Some other related works include Li et al. (2019), Zhu et al. (2012), Zhang et al. (2014), Goldman et al. (2019), and Petsiuk et al. (2020).

The second line concerns the research in **comparing object detection models**. Some works have analyzed and reported statistics and performances over benchmark datasets such as PASCAL VOC (Everingham et al., 2010; 2015), MS COCO (Lin et al., 2014), CityScapes (Cordts et al., 2016), and OpenImages (Kuznetsova et al., 2018). Recently, Huang et al. (2017) performed a

speed/accuracy trade-off analysis of modern object detectors. Dollar et al. (2011) and Borji et al. (2015) compared person detection and salient object detection models, respectively. Michaelis et al. (2019) assessed detection models on degraded images and observed about 30–60% performance drop, which could be mitigated by data augmentation. To resolve the issues with the AP score, some works have attempted to introduce alternative (e.g., Hall et al. (2018)) or complementary evaluation measures (e.g., Oksuz et al. (2018); Rezatofighi et al. (2019)). A large number of works have also assessed object recognition models and their robustness (e.g., Hendrycks & Dietterich (2019); Azulay & Weiss (2018); Recht et al. (2019); Mishkin et al. (2017)).

Works in the third line study the **role of context in visual recognition and object detection** (e.g., Bar (2004); Wolf & Bileschi (2006); Zhu et al. (2016); Marat & Itti (2012); Heitz & Koller (2008); Torralba & Sinha (2001); Rabinovich et al. (2007); Rosenfeld et al. (2018); Galleguillos & Belongie (2010)). Heitz & Koller (2008) proposed a probabilistic framework to capture contextual information between "stuff" and "things" to improve detection. Barnea & Ben-Shahar (2019) utilized co-occurrence relations among objects to improve the detection scores. Divvala et al. (2009) explored different types of context in recognition. Please see also Heitz & Koller (2008), Chen et al. (2018), Song et al. (2011), Hu et al. (2018a), Marat & Itti (2012), and Alamri & Pugeault (2019).

3 Experimental Setup

Benchmarks. We establish our analysis based on two recent large-scale object detection benchmarks: `MMDetection`[3] (Chen et al., 2019b) and `Detectron2`[4]. The former evaluates more than 25 models. The latter includes several variants of FastRCNN (Girshick, 2015). In both benchmarks, all MS COCO models have been trained on *train2017* and evaluated on *val2017*. Here, we use MMDetection to train and test additional models on a new dataset of clothing items.

Models. We consider major object detection models including several variants of the RCNN such as FasterRCNN (Ren et al., 2015), MaskRCNN (He et al., 2017), RetinaNet (Lin et al., 2017), GridRCNN (Lu et al., 2019), LibraRCNN (Pang et al., 2019), CascadeRCNN (Cai & Vasconcelos, 2018), MaskScoringRCNN (Huang et al., 2019), GAFasterRCNN (Zhu et al., 2019), and Hybrid Task Cascade (Chen et al., 2019a), as well as SSD (Liu et al., 2016), FCOS (Tian et al., 2019), and CenterNet (Zhou et al., 2019). Different backbones for each model are also taken into account.

Datasets. Four datasets including PASCAL VOC (Everingham et al., 2015), our home-brewed FASHION dataset, MS COCO (Lin et al., 2014), and OpenImages (Kuznetsova et al., 2018) are employed. Over VOC, we use *trainval0712* for training (16,551 images, 47,223 boxes) and *test2007* (4,952 images, 14,976 boxes) for testing. This dataset has 20 object categories. Our FASHION dataset covers 40 categories of clothing items (39 + *humans*). Trainval, and test sets of this dataset contain 206,530 images (776,172 boxes) and 51,650 images (193,689 boxes), respectively. Fig. 4 displays samples from this dataset (See Appendix B for more samples and statistics). This is a challenging dataset since clothing items are non-rigid as opposed to most of the MS COCO and VOC objects. MS COCO has 80 categories. It has carried the torch for benchmarking advances in object detection for the past 7 years. We use *train2017* for training (118,287 images, 860,001 boxes) and *val2017* (5,000 images, 36,781 boxes) for testing. Finally, we use the OpenImages V4 dataset, used in the Kaggle competition[5]. It has 500 classes and contains 1,743,042 images (12,195,144 boxes) for training and 41,620 images (226,811 boxes) for validation (used here for testing).

Metrics. We use the COCO evaluation tool to measure AP at IOU thresholds of 0.5, 0.75, and 0.5:0.05:0.95. APs are calculated per class and are then averaged. We also report breakdown APs over small (area$\leq 32^2$ pixels), medium (32^2 <area$\leq 96^2$), and large (area$> 96^2$) objects.

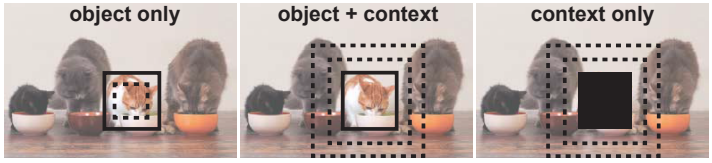
4 Characterizing the Empirical upper bound in AP

We define the empirical upper bound in AP as the score of the object detector that a) has access to the true location of the objects, and b) ground truth bounding boxes are labeled by the best object classifier. This way we essentially assume that the localization problem is solved and what remains is only object recognition. Beware that we do not mean to undermine the importance of the localization component. What we intend to convey is that assuming no further progress in object recognition

[3] `https://github.com/open-mmlab/mmdetection`

[4] `https://github.com/facebookresearch/detectron2`

[5] `https://www.kaggle.com/c/open-images-2019-object-detection`



test on \ train on	Dataset	object only					object + context					context only		
		0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2	1.2	2	all img
full object (=1)	VOC	39.3	68.0	82.6	92.5	**94.8**	93.0	91.6	90.6	88.6	87.0	63.6	64.9	35.3
	FASHION	-	52.9	66.4	71.7	**88.8**	82.3	77.2	71.8	67.9	64.8	29.0	32.2	12.0
	COCO	-	-	67.1	79.8	**86.7**	82.9	78.3	72.5	67.4	63.0	43.7	48.9	11.0
	OpenImages	-	-	-	-	**69.0**	65.1	62.7	-	-	-	-	-	-
same as training	VOC	61.1	79	87.2	92.4	**94.8**	94.4	94.0	93.7	92.4	91.3	61.8	79.6	73.5
	FASHION	-	73.1	81.2	86.7	**88.8**	88.4	87.2	85.9	83.82	82.28	72.5	76.1	74.3
	COCO	-	-	74	81.4	86.7	86.8	87.3	87.6	**87.7**	87.3	57.6	69.7	63.4

Figure 2: Top: Illustration of the context surrounding an object, **Bottom:** Object recognition accuracy. Top rows: testing on the canonical object size (used in the rest of the paper; See Appendix C for confusion matrices.). Bottom rows: training and testing are the same, for example, a classifier is trained on the object-only case 0.6 and is then tested on the object-only case 0.6. The best accuracy in each row is highlighted in **bold**.

and investing all of our efforts in solving the localization problem can lead us to this upper bound and not beyond that. Knowing the upper bound can help us to better coordinate our efforts. This object detector, however, might not give us the upper bound AP due to the subtleties involved in AP calculation. Specifically, it might be possible to improve upon this detector in at least two ways: a) by exploiting the local scene context around an object to improve the classification accuracy and thus better UAP, and/or b) by searching among the bounding boxes around the target object (those with a certain overlap with it) and see whether any of them can be classified better, compared to the target box itself. This does not matter for determining the UAP at the perfect IOU (=1) but may affect UAP at IOUs lower than one. We carefully investigate these challenges in the following.

4.1 Utility of the surrounding context

We trained ResNet152 (He et al., 2016) on target bounding boxes in 3 settings as shown in Fig. 2: 1) `object only`, 2) `object + context`, and 3) `context only`. Standard data augmentation techniques including color jittering, random horizontal flip, and random rotation (10 degrees) were applied. Boxes were resized to 224×224 pixels and models were trained for 15 epochs. Trained models were then tested on the original object boxes. Results (top-1 accuracy) are shown in Fig. 2 (bottom panel). We find that the canonical object size results in the best classification accuracy over all four datasets. Enlarging or shrinking the object bounding box lowers the performance. The context-only scenario results in a high classification score but still performs below other cases. Stretching the context to the whole scene drops the performance significantly. Training and testing models in the same condition (second row in the table; e.g., both on *object+context*) results in high accuracy on that specific condition but does not lead to better overall accuracy on objects.

4.2 Searching for the best label

Essentially the problem definition here is how best we can classify a target box by utilizing all the available information in the scene. This is different from recognition models where they treat objects in isolation. Notice that recognition accuracy is not the same as AP since detection scores also matter in the AP calculation (i.e., detections are ranked). Having the best classifier in hand, we are ready to approximate the UAP. Before delving into details lets recap how AP is calculated.

AP calculation. For each category, detections on all images are sorted according to their confidences. Starting from the top of this list, the target with the highest IOU with each detection is considered. We have a true positive (TP/hit) if the IOU is $\geq thresh$, and if the target has not been assigned before. We have a false positive (FP) if IOU$<$ $thresh$ (i.e., localization error) or if the target has been assigned (i.e., duplicate; two predictions on the same target). A target box can be matched with only one detection (the one with the highest confidence score and IOU$\geq thresh$). If a detection has IOU$\geq thresh$ with two targets, it is assigned to the one with the highest IOU which is not assigned already. Scanning the sorted detection list again, a precision for each recall is obtained and is used to draw the Recall-Precision (RP) curve and to compute the AP. See also Appendix A.

Strategies for labeling the boxes. We explore two strategies in pursuit of the empirical upper bound in AP. In the **first one**, we apply the best classifier from the previous section to the target bounding boxes. The detector built in this fashion gives the same AP regardless of the IOU threshold since our detections are target boxes. As we argued above, it is not possible to improve this detector at IOU=1. However, if we are interested in upper bound at a lower IOU threshold (γ), then it might be possible to do better by searching among the candidate boxes near a target box and choose the one that can be classified better than the target box, or by aggregating the information from all nearby boxes. Thus, in our **second strategy**, we sample boxes around an object and either apply the original classifier

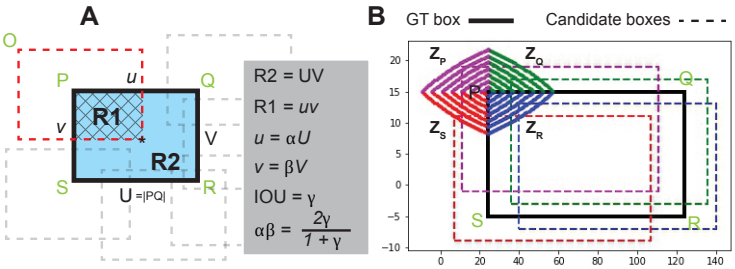


Figure 3: **A:** Illustration of our setup for finding boxes with IOU $\geq \gamma$ with the target box (corresponding to $\alpha\beta = 2\gamma/(1+\gamma)$; $\alpha\beta = 2/3$ for $IOU = 0.5$), **B:** The solutions are 4 curves represented by Eqs. 4 to 7. Four sample rectangles are shown with dashed lines.

(trained on the canonical object size) or train and test new classifiers on the surrounding boxes. In any case, we always keep the target box but change its label and/or its classification confidence. First, let's take a look at our box sampling strategy, which is illustrated in Fig. 3.

Sampling boxes at IOU threshold $\geq \gamma$. We are interested in finding the coordinates of the top-left corner of all rectangles[6] with IOU $\geq \gamma$ $\left(\gamma \leq 1\right)$ with the ground-truth bounding box. We use the coordinate system centered at the top-left corner P of the target box (the PQRS rectangle; shown in black) which can be easily converted to the image level coordinate frame. Let's first find the relationship between the coordinates of the point marked with * ($< u, v >$) and overlap threshold γ. According to the illustration in Fig. 3.A, we have:

$$\text{R1 = uv}, \quad \text{R2 = UV}, \quad \text{IOU} = \gamma, \quad \text{IOU} = \frac{R1}{2R2 - R1} \tag{1}$$

From these equations and assuming $u = \alpha U$ and $v = \beta V$, it is easy to derive the following equations:

$$R1 = \alpha U \beta V, \quad R1 = \frac{2\gamma}{1+\gamma} R2 \tag{2}$$

and from there we obtain:

$$\alpha\beta = \frac{2\gamma}{1+\gamma}, \quad (\alpha\beta = \frac{2}{3} \text{ for } \gamma = 0.5) \tag{3}$$

The same equation governs the coordinates of the bottom-left, top-left, and top-right corners of the rectangles intersecting with the target box at points Q, R, and S, respectively (in the coordinate frames centered at each of these points, in order). Calculating the top-left corner of these rectangles (in their corresponding coordinate frames) and representing them in the coordinate frame of the image, we arrive at the following four equations (notice that these are curves not lines):

$$Z_P: \ \langle \ (\alpha - 1)U + x_P, \ (\beta - 1)V + y_P \ \rangle \tag{4}$$

$$Z_Q: \ \langle \ (1 - \alpha)U + x_P, \ (\beta - 1)V + y_P \ \rangle \tag{5}$$

$$Z_R: \ \langle \ (1 - \alpha)U + x_P, \ (1 - \beta)V + y_P \ \rangle \tag{6}$$

$$Z_S: \ \langle \ (\alpha - 1)U + x_P, \ (1 - \beta)V + y_P \ \rangle \tag{7}$$

$$\forall \ \alpha, \beta \leq 1, \ \text{s.t.} \ \ \alpha\beta = \frac{2\gamma}{1+\gamma} \tag{8}$$

Using above equations, we sample m (here m=4) rectangles with $IOU \geq \gamma$ (Fig. 3.B) and label them with the label of the target box. We then train a new classifier (same ResNet152 as above) on these boxes. This is effectively a new data augmentation technique. Notice that UAP is the direct consequence of the classification accuracy, meaning if we can classify objects better, we can reach a higher UAP. To estimate UAP, we sample m rectangles around a target box (with $IOU \geq \gamma$), and then label the target box with a) the label (and confidence) of the bounding box with the highest classification score (i.e., **the most confident box**), or b) the **most frequent label** among the nearby boxes (with the maximum confidence score among them).

Dataset	Acc.	Most Confident Box				Most Frequent Label			
		AP	AP_l	AP_m	AP_s	AP	AP_l	AP_m	AP_s
VOC	93.7	88.7	91.7	81.4	63.8	89.1	92.0	82.9	60
FASHION	87.4	68.1	68.6	<u>61.9</u>	<u>49.5</u>	67.7	68.2	60.7	<u>47.8</u>
COCO	84.8	76.9	81.8	80.6	62.8	76.4	82.0	80.4	60.7

Table 1: Upper bound AP according to our second strategy (i.e., searching for the best bounding box or object label near a target box; among boxes with IOU ≥ 0.5). Notice that upper bound for AP, AP0.5, and AP0.75 are all the same. Underlined numbers show where we could improve upon the first strategy.

4.3 UPPER BOUND AP RESULTS

Here, we report classification scores, upper bound APs, the score of the models (mean AP over all IOUs; unless specified otherwise), and the breakdown AP over categories. See also Appendix E.

[6]Here, we assume that all boxes have the same width and height as the target box. The solution can be easily extended to the case where rectangles are non-homogeneous.

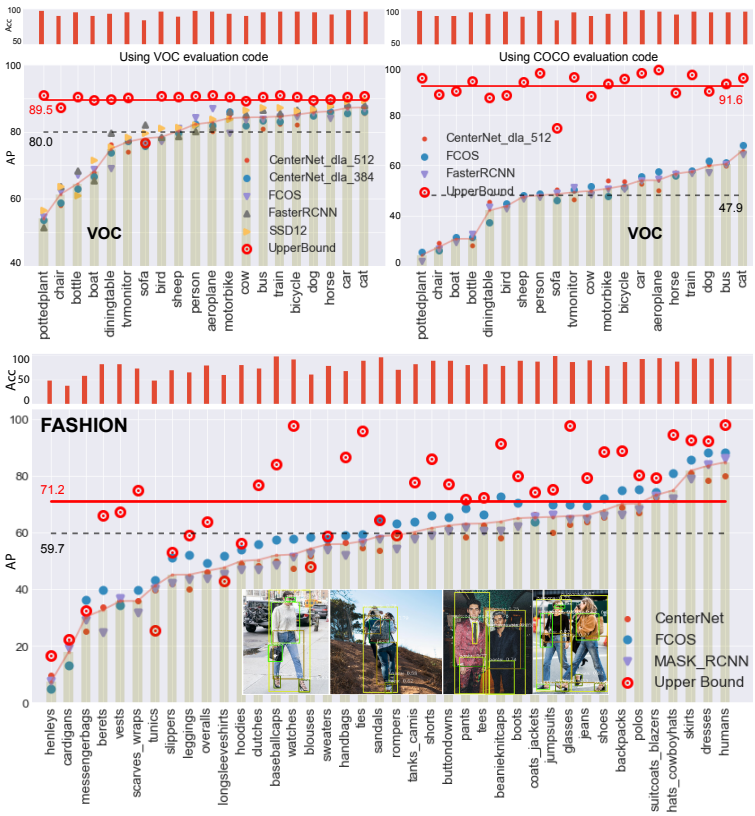


Figure 4: Top: UAP and Model APs over the PASCAL VOC dataset using VOC (left) and COCO AP evaluation codes (right). Categories are sorted according to the average model AP. Bar charts on the top show classification accuracy. Solid red and dashed black lines represent upper bound AP, and the best model AP, respectively. **Bottom:** UAP and model APs over the Fashion dataset. On some rare occasions (e.g., tunics; often small objects), UAP is lower than the model AP possibly because our classifier has to elicit a decision for any box, thus it may generate more false positives than a model that misses objects (i.e., we do not have misses). This may result in a lower precision for some classes for our UAP than a model, but our setup has a higher recall. See also Appendix E.

Comparison of strategies. Summary results of the first strategy are shown in Fig. 1. As expected UAPs over all IOUs are the same and are much better than the model APs. Contrary to our expectation, the second strategy did not lead to higher UAPs, except in a few cases (over medium and small objects over the FASHION dataset using the most confident boxes), as shown in Table 1. Applying the original classifier, instead of training new ones on surrounding boxes, or sampling only boxes with higher IOU thresholds (e.g., $\gamma = 0.9$) did not improve the results. Also, setting the confidence of the detections to 1 lowered the UAP. We attribute the failure of the second strategy to the fact that the surrounding boxes may contain additional visual content which may introduce noise in the labels. This leads to lower classification accuracy and hence a lower UAP. Therefore, in what follows we only discuss the results using the first strategy.

PASCAL VOC. Fig. 4 (top) shows results using both VOC and MS COCO evaluation tools. The VOC evaluation code is based on IOU=0.5 and calculates the area under the PR curve slightly differently than the COCO code. For VOC, we adopt the code from the CenterNet repository. We have trained and tested 5 models on VOC dataset including FasterRCNN, FCOS, SSD512, and two variants of CenterNet. The classification accuracy over VOC is very high (94.8%). Consequently, the UAP is high (91.6% using the MS COCO code). FCOS model does the best here with the AP of 47.9% (right panel in Fig. 4; dashed lines). As can be seen, there is a large gap between the AP of the best model and the UAP on this dataset (∼45 AP units). Models behave similarly across categories.

FASHION. Results are shown in Fig. 4 (bottom). The best classification accuracy on this dataset is 88.8% (Fig. 2). The UAP is 71.2% and the AP of the best model is 59.7% (FCOS). Interestingly, FCOS performs very close to the upper bound at IOU=0.5 (Fig. 1). Models perform better here than over the VOC. The FASHION UAP is lower than the VOC UAP perhaps because classification is more challenging on the FASHION dataset. The gap between UAP and model AP here, however, is much narrower than VOC. This could be partly because FASHION scenes have less clutter and larger objects than the VOC scenes. While per-class UAP is above the AP of the best model on all VOC classes, over the FASHION dataset UAP falls below the best model AP over five categories (`messenger bags`, `tunics`, `long sleeve shirts`, `blouses`, `rompers`). Looking at the classification scores, we find that these categories are hard to classify.

MS COCO. Borrowing the *MMDetection* benchmark and adding the results from CenterNet to it, we end up comparing 15 models (71 in total; the combination of models and backbones). The best models on this dataset are Hybrid Task Cascade model (Chen et al., 2019a) and Cascade

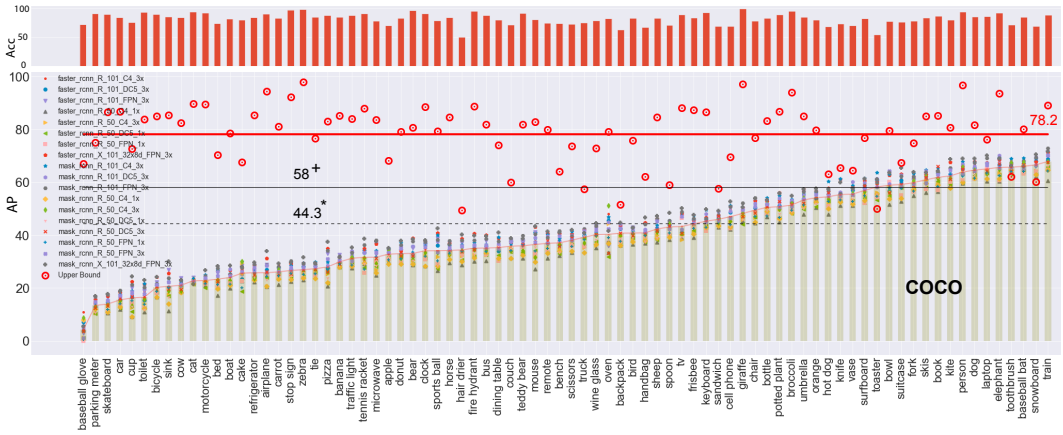


Figure 5: Detection APs over MS COCO dataset borrowed from the *Detectron2* benchmark. The black dash line corresponds to the best model among the models we analyzed (the score shown with "*"). The black solid line shows the most recent results (the score shown with "+"). See Appx. D for results over the *MMDetection*.

MaskRCNN (Cai & Vasconcelos, 2018), with APs of 46.9% and 45.7%, respectively. See also Fig. 11 in Appx. D. The UAP on COCO is about 78.2% which is about 35% (absolute difference) above the AP of the best model (∼20 above the most recent model). Recall that UAP does not depend on the IOU threshold since detected boxes are ground-truth targets. The gap is much smaller at AP50 which is about 10%[7]. The UAP over small objects is much lower than the UAP over large objects. This also holds for models. The gap between the UAP and model AP over small objects is about 35 AP unit which is much wider than the corresponding gap over medium or large objects. Breakdown APs over object categories are shown in Fig. 5. For this, we use the *Detectron2* benchmark which reports per-category results mainly over RCNN model family. We noticed that aggregate scores on *MMDetection* and *Detectron2* are quite consistent. Among 18 variants of FasterRCNN and MaskRCNN, the best model has the AP of 44.3 (shown by the dashed line) which is lower than the best available model on COCO (58%; Fig. 1) and the UAP. Among the 80 object categories, only three (`snowboard, toothbrush, toaster`) have UAPs below the best model APs.

Open Images. This dataset (Kuznetsova et al., 2018) is the latest endeavor in object detection and is much more challenging than its predecessors. Our classifier achieves 69.0% top-1 accuracy on the validation set of OpenImages V4 which is lower than the other three datasets. We obtain the UAP of 58.9%, using the TensorFlow evaluation code for computing the AP score on this dataset, which is slightly different than the COCO AP calculation tool (here we discarded grouping and super-category). We are not aware of any model scores on this particular set of OpenImages V4.

UAP *vs.* classification accuracy. We found that there is a linear positive correlation (R^2 = 0.81 on MS COCO) between the UAP and the classification accuracy as shown in Fig. 6. The higher the classification accuracy, the higher the UAP. We did not find a correlation between the accuracy and model APs, nor between the object size and accuracy (or UAP).

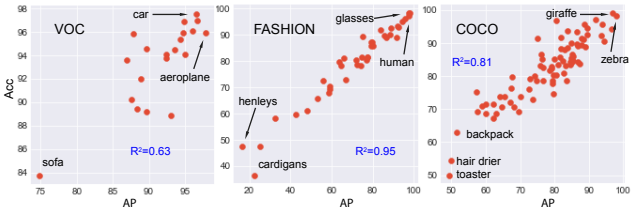
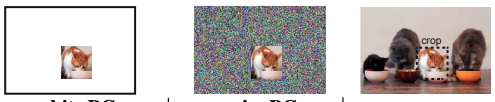


Figure 6: Correlation between the classification accuracy and upper bound AP. The higher the accuracy, the better the UAP.

5 Analysis of Context

We did not find a significant benefit from the context in classifying objects and measuring UAP. *How important is context in detecting objects?* To find out, we generated three types of stimuli in which a single object 1) was placed in a white background (white BG), 2) was placed in a white noise background (noise BG), or 3) cropped off the image (crop). The number of generated images is equal to the number of objects (carried out over COCO), thus results in these cases are comparable.

[7]The best available scores on COCO*val2017* are shown in Fig. 1. Interestingly, the gap at AP50 is almost closed (∼2). There is, however, still a large gap at AP over all IOUs, and also over medium and small objects.



Model	white BG			noise BG			crop		
	AP	$AP^{.5}$	$AP^{.75}$	AP	$AP^{.5}$	$AP^{.75}$	AP	$AP^{.5}$	$AP^{.75}$
FasterRCNN	31.1	42.0	36.1	31.8	39.8	36.8	8.4	15.0	8.2
RetinaNet	33.1	41.0	37.3	32.7	39.1	36.6	16.9	22.7	18.8
FCOS	34.5	42.0	37.1	34.2	39.8	37.4	14.3	18.5	15.3
SSD512	27.4	36.7	32.3	26.0	33.4	34	13.4	18.9	14.9
Model	AP_s	AP_m	AP_l	AP_s	AP_m	AP_l	AP_s	AP_m	AP_l
FasterRCNN	7.5	35.9	49.9	7.0	36.6	52.1	0	1.3	18.7
RetinaNet	8.3	37.5	53.2	6.4	38.3	54.2	1	5.2	34.1
FCOS	8.5	39.8	55.2	9.4	39.5	54.8	1	4.5	32.2
SSD512	7.0	31.4	45.1	4.6	29.3	45.2	1	2.9	25.7

Figure 7: Top: Sample images placed in white and white noise backgrounds as well as the cropped objects (which were shown to the models in isolation). Please see also Appendix G for more samples. **Bottom:** Performance of four object detectors. Models perform very poorly on cropped objects, especially on small objects. This finding agrees with Rosenfeld et al. (2018) where they found that object detectors fail to detect objects out of their contexts (e.g., an elephant in the living room).

According to Fig. 7, models perform about the same over white BG and noise BG cases but much lower than when applied to the original images (See Fig. 1). This suggests that detectors are indeed relying on context (but perhaps in a different manner than humans). This explains why they sometimes miss objects out of their context as shown in Rosenfeld et al. (2018). Models performed terribly over the cropped (and resized to meet the required input size) objects. We also tested the models on objects that were cropped and resized such that their smallest dimension became 300 pixels (while preserving the aspect ratio). The performance was still very poor (See Appendix F). This indicates that detectors are overfitted to the scale of objects seen during training. In all of three cases, models are hindered much more on small objects than over medium or large ones.

6 Error Diagnosis

To pinpoint the shortcomings of models, we follow the analysis by Hoiem et al. (2012) over deep object detectors. We start from the original detection set and progressively measure the impact of fixing different error types on mAP (@IOU=0.5). As Fig. 8 shows, over the MS COCO dataset, confusion with the background (BG) and misses (FN) account for most of the errors across the three models. Over VOC, a fewer number of objects are missed compared to COCO. Also, objects are less confused with each other on this dataset since categories are fewer and are more distinct. Over the FASHION dataset, confusion with similar and other classes plays a significant role (more than other datasets) since several object categories resemble each other (e.g., henleys *vs.* polos or slippers *vs.* sandals). Among models, it seems that MaskRCNN misses more objects that others, while CenterNet often confuses background regions as objects. Fig. 15 (Appendix G) shows the breakdown of errors over small, medium, and large objects. As expected, models obtain a much higher mAP over large objects than small ones. A lot of background regions, however, are still classified as large objects. Small objects are missed more frequently followed by medium and large ones, in order.

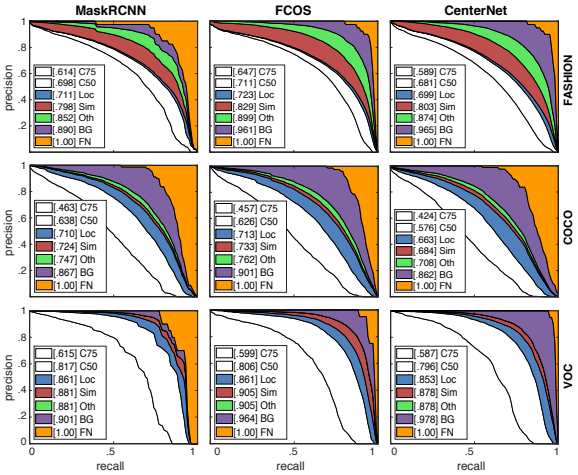


Figure 8: Quantifying the contribution of error types in models using the COCO analysis tool (@ IOU=0.5).

7 Epilogue

We found that a) considering the most recent results, models still perform significantly below what is empirically possible, and b) the performance gap is wider over small objects, and models are highly accustomed to the scales of objects seen during training. Our finding that the gap at IOU=.5 is almost closed, suggests that perhaps the bottleneck in object detection has shifted from recognition (witnessed by our analysis in Fig. 8) to localization (using the latest results). However, this requires the use of newer and more effective error analysis tools (such as the one proposed by Bolya et al. (2020)) for further inspection. We did not find a significant benefit from the surrounding context of an object or its nearby overlapping boxes to improve the UAP. A further investigation of this with extensive data augmentation and optimization may increase UAP but is unlikely to change it drastically. We invite researchers to periodically, as better recognition models emerge, update the upper bound in detection scores, and also scores on other tasks that depend on object recognition.

REFERENCES

Faisal Alamri and Nicolas Pugeault. Contextual relabelling of detected objects. *arXiv preprint arXiv:1906.02534*, 2019.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

Moshe Bar. Visual objects in context. *Nature Reviews Neuroscience*, 5(8):617, 2004.

Ehud Barnea and Ohad Ben-Shahar. Exploring the bounds of the utility of context for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7412–7420, 2019.

Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. Tide: A general toolbox for identifying object detection errors. 2020.

Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *IEEE transactions on image processing*, 24(12):5706–5722, 2015.

Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154–6162, 2018.

Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4974–4983, 2019a.

Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019b.

Zhe Chen, Shaoli Huang, and Dacheng Tao. Context refinement for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 71–86, 2018.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *2009 IEEE Conference on computer vision and Pattern Recognition*, pp. 1271–1278. IEEE, 2009.

Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4): 743–761, 2011.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

Carolina Galleguillos and Serge Belongie. Context based object categorization: A critical survey. *Computer vision and image understanding*, 114(6):712–722, 2010.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Eran Goldman, Roei Herzig, Aviv Eisenschtat, Jacob Goldberger, and Tal Hassner. Precise detection in densely packed scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5227–5236, 2019.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

David Hall, Feras Dayoub, John Skinner, Peter Corke, Gustavo Carneiro, and Niko Sünderhauf. Probability-based detection quality (pdq): A probabilistic approach to detection evaluation. *arXiv preprint arXiv:1811.10800*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *European conference on computer vision*, pp. 30–43. Springer, 2008.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Derek Hoiem, Yodsawalai Chodpathumwan, and Qieyun Dai. Diagnosing error in object detectors. In *European conference on computer vision*, pp. 340–353. Springer, 2012.

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 9401–9411, 2018a.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018b.

Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7310–7311, 2017.

Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6409–6418, 2019.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.

Hengduo Li, Bharat Singh, Mahyar Najibi, Zuxuan Wu, and Larry S Davis. An analysis of pre-training on object detection. *arXiv preprint arXiv:1904.05871*, 2019.

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 *European conference on computer vision*, pp. 740–755. Springer, 2014.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pp. 21–37. Springer, 2016.

Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7363–7372, 2019.

Sophie Marat and Laurent Itti. Influence of the amount of context learned for improving object classification when simultaneously learning object and contextual cues. *Visual Cognition*, 20 (4-5):580–602, 2012.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019.

Dmytro Mishkin, Nikolay Sergievskiy, and Jiri Matas. Systematic evaluation of convolution neural network advances on the imagenet. *Computer Vision and Image Understanding*, 161:11–19, 2017.

Roozbeh Mottaghi, Sanja Fidler, Alan Yuille, Raquel Urtasun, and Devi Parikh. Human-machine crfs for identifying bottlenecks in scene understanding. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):74–87, 2015.

Kemal Oksuz, Baris Can Cam, Emre Akbas, and Sinan Kalkan. Localization recall precision (lrp): A new performance metric for object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 504–519, 2018.

Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 821–830, 2019.

Devi Parikh and C Lawrence Zitnick. Finding the weakest link in person detectors. In *CVPR 2011*, pp. 1425–1432. Citeseer, 2011.

Vitali Petsiuk, Rajiv Jain, Varun Manjunatha, Vlad I Morariu, Ashutosh Mehra, Vicente Ordonez, and Kate Saenko. Black-box explanation of object detectors via saliency maps. *arXiv preprint arXiv:2006.03204*, 2020.

Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge J Belongie. Objects in context. In *ICCV*, volume 1, pp. 5. Citeseer, 2007.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 658–666, 2019.

Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv preprint arXiv:1808.03305*, 2018.

Olga Russakovsky, Jia Deng, Zhiheng Huang, Alexander C Berg, and Li Fei-Fei. Detecting avocados to zucchinis: what have we done, and where are we going? In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2064–2071, 2013.

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. *International journal of computer vision*, 115(3):211–252, 2015.

Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3578–3587, 2018.

Zheng Song, Qiang Chen, Zhongyang Huang, Yang Hua, and Shuicheng Yan. Contextualizing object detection and classification. In *CVPR 2011*, pp. 1585–1592. IEEE, 2011.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *In Proc. ICLR*, 2014.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *arXiv preprint arXiv:1904.01355*, 2019.

Antonio Torralba and Pawan Sinha. Statistical context priming for object detection. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pp. 763–770. IEEE, 2001.

Carl Vondrick, Aditya Khosla, Tomasz Malisiewicz, and Antonio Torralba. Hoggles: Visualizing object detection features. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1–8, 2013.

Lior Wolf and Stanley Bileschi. A critical view of context. *International Journal of Computer Vision*, 69(2):251–261, 2006.

Peng Zhang, Jiuling Wang, Ali Farhadi, Martial Hebert, and Devi Parikh. Predicting failures of vision systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3566–3573, 2014.

Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. How far are we from solving pedestrian detection? In *Proceedings of the iEEE conference on computer vision and pattern recognition*, pp. 1259–1267, 2016.

Zhi Zhang, Tong He, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of freebies for training object detection neural networks. *arXiv preprint arXiv:1902.04103*, 2019.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

Xiangxin Zhu, Carl Vondrick, Deva Ramanan, and Charless C Fowlkes. Do we need more training data or better models for object detection?. In *BMVC*, volume 3, pp. 5. Citeseer, 2012.

Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. *arXiv preprint arXiv:1904.05873*, 2019.

Zhuotun Zhu, Lingxi Xie, and Alan L. Yuille. Object recognition with and without objects, 2016.

Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.

A AVERAGE PRECISION (AP) CALCULATION

A step by step explanation for AP calculation[8]:

Firstly, AP is a "per-class" measure. For a class, the 5 step process to compute the AP is as follows:

1. For the entire dataset, sort all the detections labeled with the same class with respect to their confidence scores.
2. Now, go over this sorted detection list and check whether each detection can be assigned to a ground-truth. The assignment (or labeling as a True Positive) is based on Intersection Over Union (IoU) of the detection with a ground truth. There is a true positive validation threshold in terms of IoU and it is generally 0.5. In this step, note that a ground truth can be matched with only one detection (and this detection is the one with the highest confidence score since we go over a sorted list).
3. At the end of the previous step, we have identified the True Positive (TP) detection boxes, False Positive (FP) detection boxes and False Negative (FN) ground truth boxes. So, by going over the sorted detection list again, we can find precision for each recall to draw the Recall-Precision (RP) curve. The blue curve in Fig. 9 is the RP curve.
4. To discard the wiggles of the RP curve, at some recall point interpolate the blue curve to the highest precision possible at the positive side of this recall point. Thus, the red curve, called the interpolated RP curve, is obtained.
5. Finally, there are 3 different methods to compute the AP using the interpolated RP curve:
   - "area under the curve approach": simply compute the area under this curve to find the AP. This is used in ImageNet Object Detection Challenge.
   - "arithmetic average approach": divide the recall domain into evenly spaced slices, check precision values at these recall values and get their average. Older Pascal VOC metric used to compute the AP in this way by using 11 recall points.
   - MS COCO style AP: It is an extended version of the arithmetic average approach. It uses 101 recall points and computes AP for 10 different TP validation threshold(0.5, 0.55, 0.6,. . . ,0.95) in terms of IoU in order to implicitly include localization error. So indeed, it is "the arithmetic average of the arithmetic average approach" on different TP validation thresholds.

   Mean Average Precision (mAP in short) is the performance measure that is assigned to an object detector (not to a single class). In all three cases, mAP is the average of APs over classes.

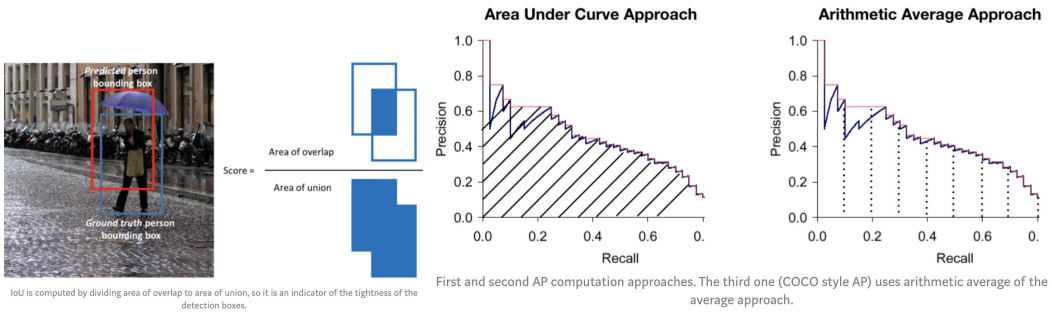


Figure 9: Illustration of AP calculation.

[8]Modified from https://medium.com/@kemal.oksz/which-one-to-measure-the-performance-of-object-detectors-ap-or-olrp-936d072a6eb0

B SAMPLE IMAGES AND STATISTICS FROM THE FASHION DATASET

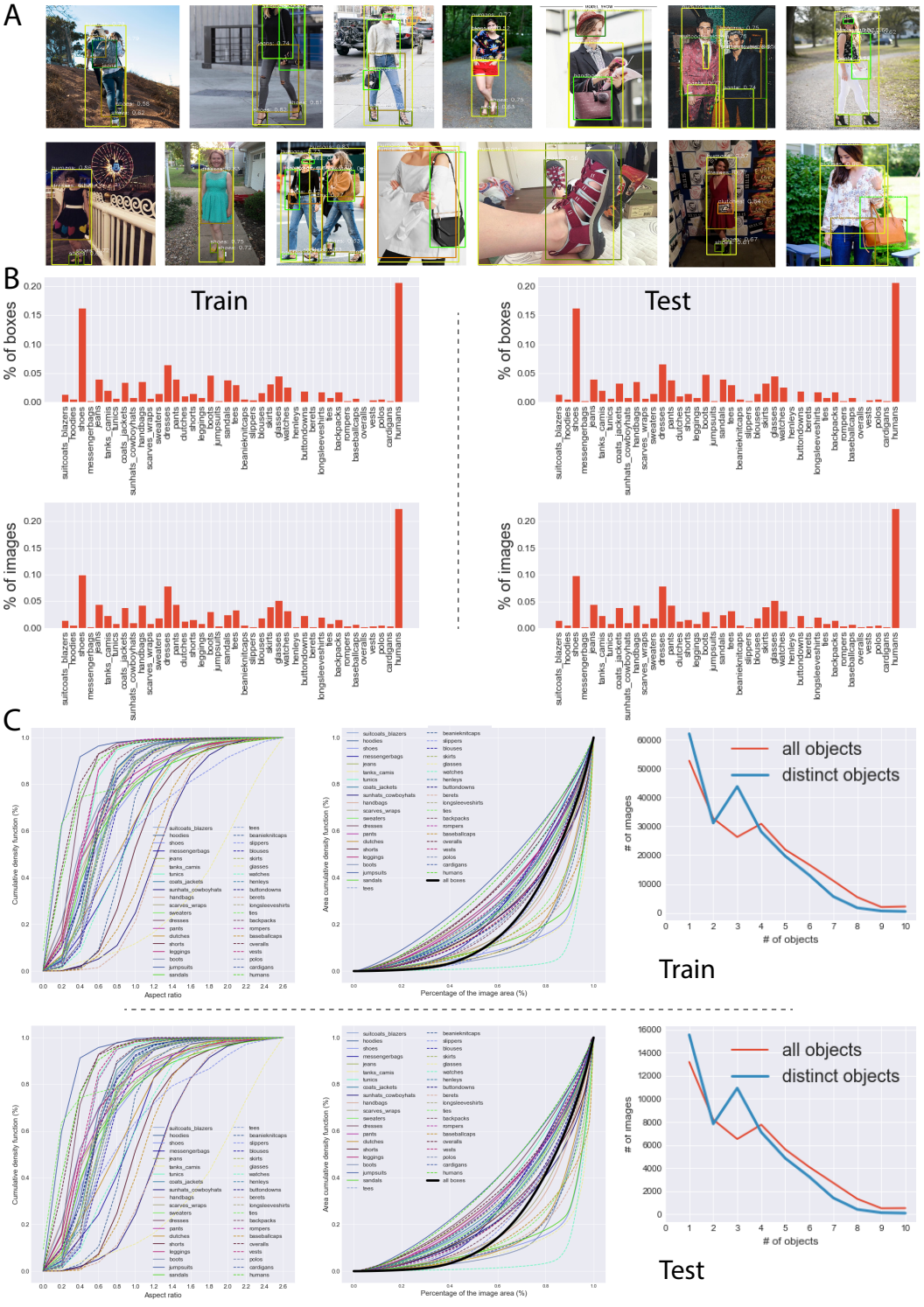
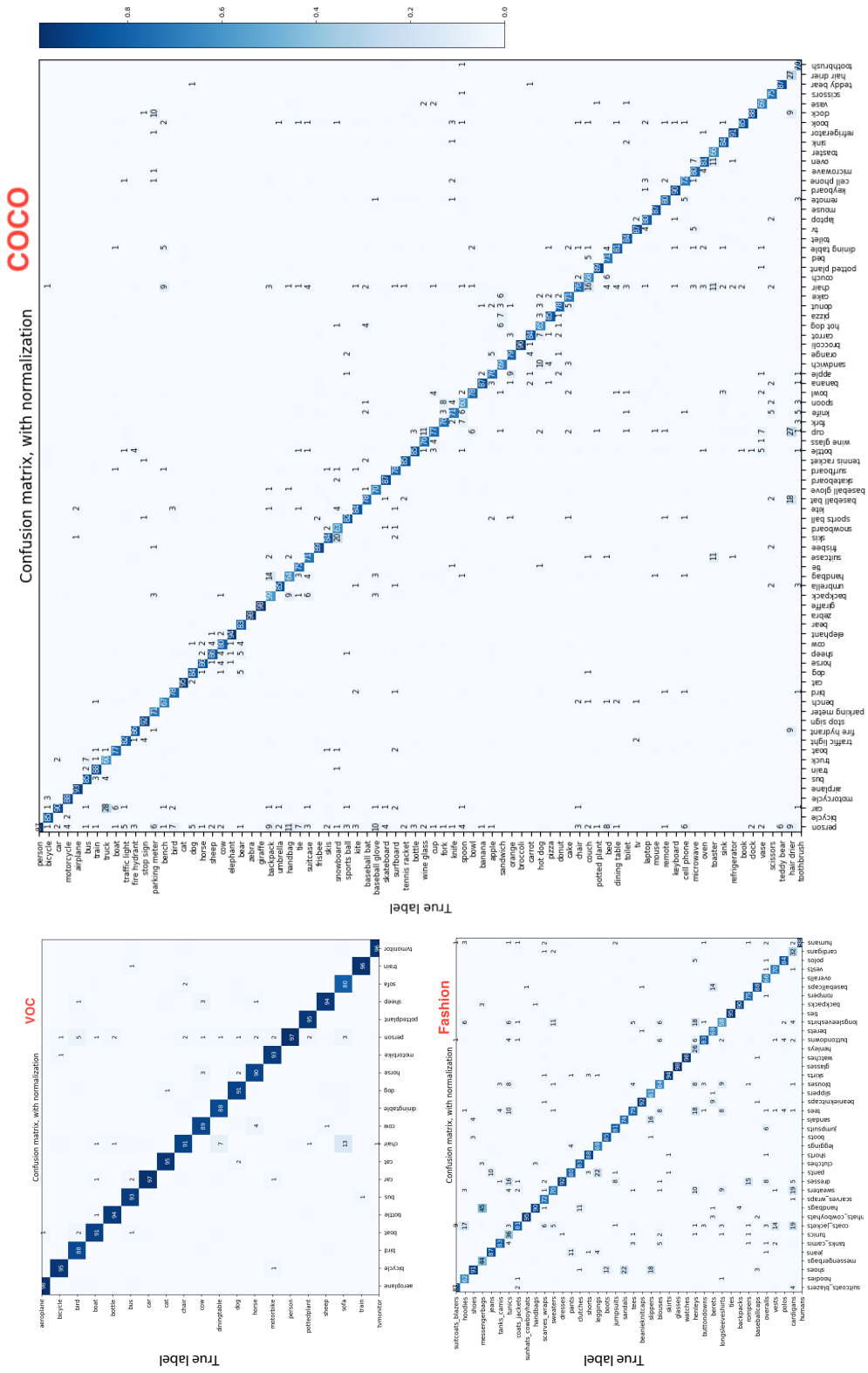


Figure 10: Statistics of the FASHION dataset. A) Sample images along with predictions by the FCOS model, B) Percentage of annotated bounding boxes and images in train and test sets, and C) Aspect ratio, object size, and number of objects per image.

C CONFUSION MATRICES OF THE CLASSIFIERS TRAINED AND TESTED ON THE ORIGINAL IMAGE SIZE (CORRESPONDING TO TABLE 1).



D UAP AND MODEL APS ON MMDETECTION BENCHMARK

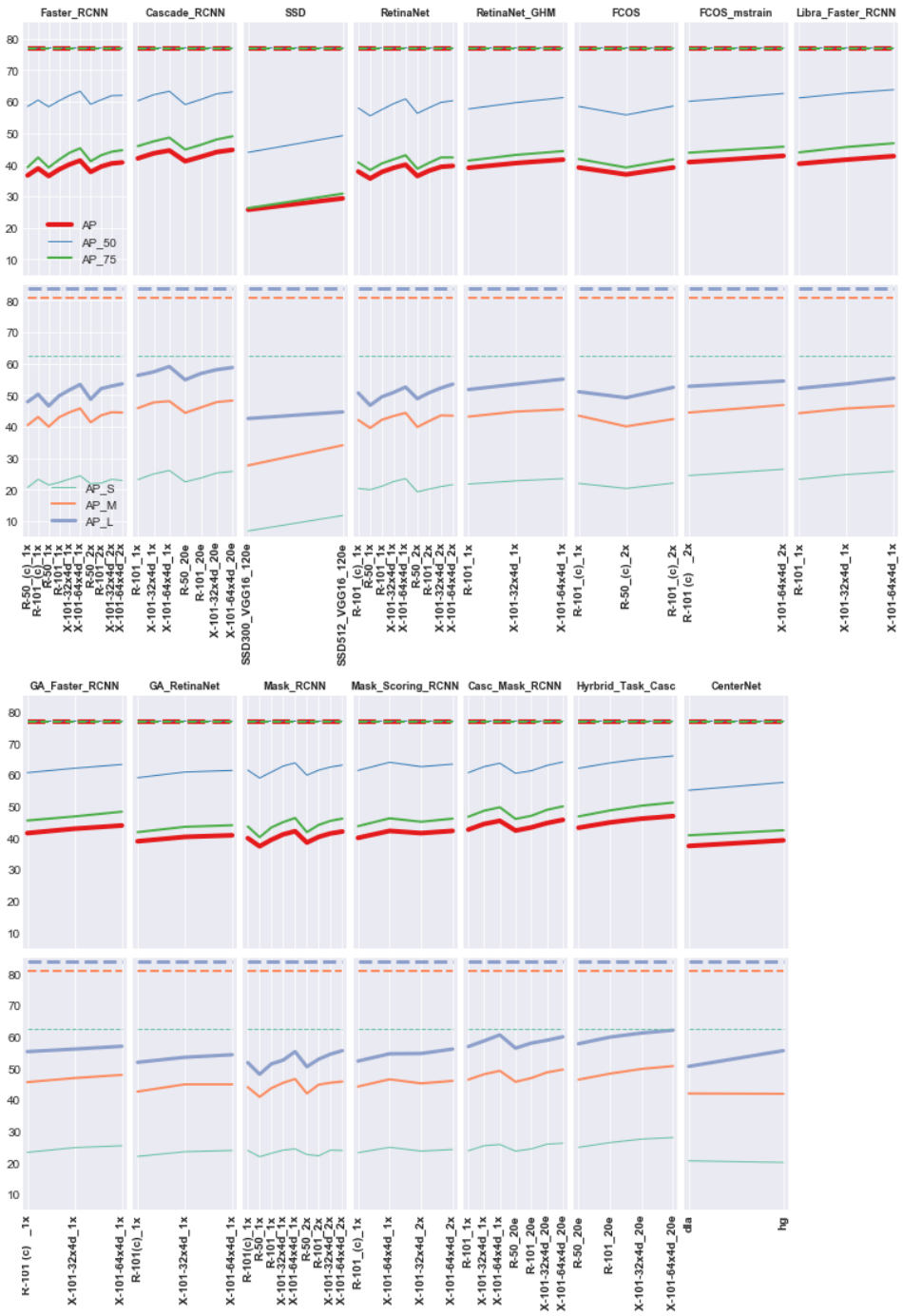


Figure 11: APs over COCO dataset borrowed from the *MMDetection* benchmark. We add CenterNet results to *MMDetection*.

E SUMMARY OF UAP AND MODEL APS OVER DATASETS

Score				VOC			FASHION			COCO		
Avg. Prec.	(AP) @[IoU=0.50:0.95	— area= all	— maxDets=100]	0.916	47.3	47.9	0.712	0.541	0.597	0.782	0.364	0.428
Avg. Prec.	(AP) @[IoU=0.50	— area= all	— maxDets=100]	0.916	71.3	71.0	0.712	0.698	0.711	0.782	0.584	0.626
Avg. Prec.	(AP) @[IoU=0.75	— area= all	— maxDets=100]	0.916	52.6	51.4	0.712	0.614	0.647	0.782	0.391	0.457
Avg. Prec.	(AP) @[IoU=0.50:0.95	— area= small	— maxDets=100]	0.707	08.6	11.1	0.457	0.108	0.182	0.635	0.215	0.265
Avg. Prec.	(AP) @[IoU=0.50:0.95	— area=medium	— maxDets=100]	0.861	30.7	32.1	0.614	0.315	0.376	0.816	0.400	0.469
Avg. Prec.	(AP) @[IoU=0.50:0.95	— area= large	— maxDets=100]	0.941	58.1	58.4	0.721	0.570	0.627	0.846	0.466	0.545
Avg. Rec.	(AR) @[IoU=0.50:0.95	— area= all	— maxDets= 1]	0.579	40.3	41.2	0.662	0.618	0.692	0.483	0.304	0.345
Avg. Rec.	(AR) @[IoU=0.50:0.95	— area= all	— maxDets= 10]	0.908	53.8	58.5	0.767	0.712	0.822	0.797	0.489	0.552
Avg. Rec.	(AR) @[IoU=0.50:0.95	— area= all	— maxDets=100]	0.930	54.1	59.5	0.774	0.714	0.824	0.812	0.514	0.582
Avg. Rec.	(AR) @[IoU=0.50:0.95	— area= small	— maxDets=100]	0.736	11.2	19.5	0.504	0.194	0.303	0.663	0.324	0.388
Avg. Rec.	(AR) @[IoU=0.50:0.95	— area=medium	— maxDets=100]	0.877	36.9	45.2	0.660	0.499	0.639	0.843	0.554	0.628
Avg. Rec.	(AR) @[IoU=0.50:0.95	— area= large	— maxDets=100]	0.954	65.7	70.2	0.782	0.742	0.850	0.893	0.645	0.735

Table 2: Precision and recall upper bounds over the three datasets (all scores). Columns under each model in order are UAP, worst model score and best model score among models we tried.

F RESULTS OF CONTEXT ANALYSIS

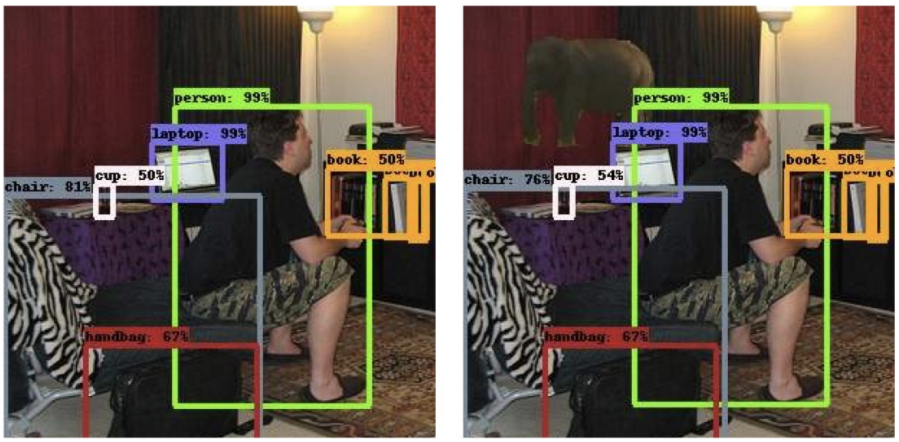


Figure 12: A state-of-the-art object detector (Faster-RCNN; trained on COCO dataset) is able to detect multiple objects in a living-room (a), but it fails to detect a transplanted object (elephant) out of its context. Rosenfeld et al. (2018) showed that a transplanted object a) may occasionally become undetected or be detected with sharp changes in confidence, b) may be classified as another object, and/or c) cause other objects to switch identity, bounding box, or disappear (image reproduced from (Rosenfeld et al., 2018)).

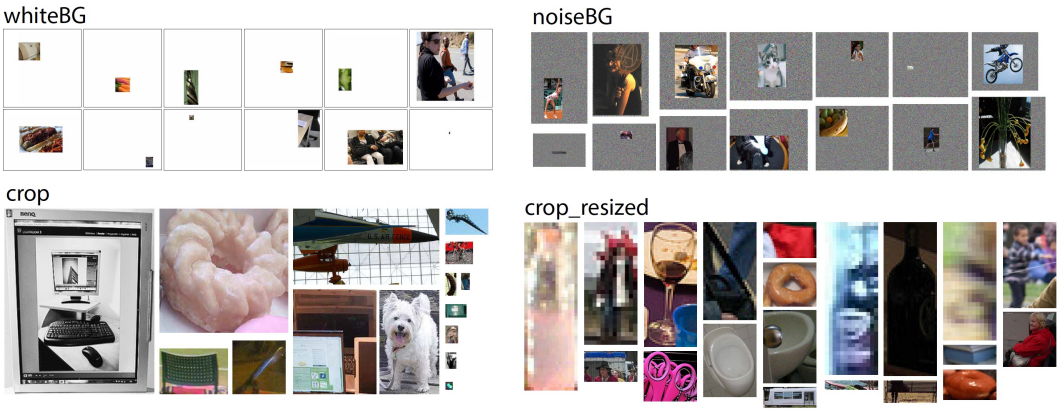


Figure 13: Sample images used in the context analysis experiments. Top row) single objects in white background, single objects in white noise background, Bottom row) cropped objected (no resizing), cropped and resized objects (width=300).

Model	Score	whiteBG	noiseBG	crop_resized	crop
FasterRCNN	Avg. Prec. (AP) @[IoU=50:95 | area= all | maxDets=100]	33.1	32.7	14.3	0.084
	Avg. Prec. (AP) @[IoU=50 | area= all | maxDets=100]	41	39.1	19.4	0.15
	Avg. Prec. (AP) @[IoU=75 | area= all | maxDets=100]	37.3	36.6	15.9	0.082
	Avg. Prec. (AP) @[IoU=50:95 | area= small | maxDets=100]	8.3	6.4	-1	0
	Avg. Prec. (AP) @[IoU=50:95 | area=medium | maxDets=100]	37.5	38.3	0.001	0.013
	Avg. Prec. (AP) @[IoU=50:95 | area= large | maxDets=100]	53.2	54.2	16.1	0.187
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets= 1]	55	54.3	31.7	0.214
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets= 10]	57.1	56.9	35.5	0.254
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets=100]	57.2	56.9	35.7	0.255
	Avg. Rec. (AR) @[IoU=50:95 | area= small | maxDets=100]	25.1	22	-1	0.045
	Avg. Rec. (AR) @[IoU=50:96 | area=medium | maxDets=100]	68.4	70.6	6.8	0.164
	Avg. Rec. (AR) @[IoU=50:97 | area= large | maxDets=100]	81.4	83.2	35.8	0.435
RetinaNet	Avg. Prec. (AP) @[IoU=50:95 | area= all | maxDets=100]	31.1	31.8	11.2	0.169
	Avg. Prec. (AP) @[IoU=50 | area= all | maxDets=100]	40.2	39.8	16.9	0.227
	Avg. Prec. (AP) @[IoU=75 | area= all | maxDets=100]	36.1	36.8	12.1	0.188
	Avg. Prec. (AP) @[IoU=50:95 | area= small | maxDets=100]	7.5	7	-1	0.001
	Avg. Prec. (AP) @[IoU=50:95 | area=medium | maxDets=100]	35.9	36.6	0.005	0.052
	Avg. Prec. (AP) @[IoU=50:95 | area= large | maxDets=100]	49.9	52.1	13.2	0.341
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets= 1]	47	48.8	21.8	0.396
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets= 10]	48.5	50.1	24.5	0.452
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets=100]	48.5	50.1	24.6	0.454
	Avg. Rec. (AR) @[IoU=50:95 | area= small | maxDets=100]	16.1	15.8	-1	0.099
	Avg. Rec. (AR) @[IoU=50:96 | area=medium | maxDets=100]	58.7	61.3	10	0.459
	Avg. Rec. (AR) @[IoU=50:97 | area= large | maxDets=100]	73.9	77.4	24.6	0.688
FCOS	Avg. Prec. (AP) @[IoU=50:95 | area= all | maxDets=100]	34.5	34.2	12.1	0.143
	Avg. Prec. (AP) @[IoU=50 | area= all | maxDets=100]	40.2	39.8	15.7	0.185
	Avg. Prec. (AP) @[IoU=75 | area= all | maxDets=100]	37.1	37.4	13.2	0.153
	Avg. Prec. (AP) @[IoU=50:95 | area= small | maxDets=100]	8.5	9.4	-1	0.001
	Avg. Prec. (AP) @[IoU=50:95 | area=medium | maxDets=100]	39.8	39.5	0.001	0.045
	Avg. Prec. (AP) @[IoU=50:95 | area= large | maxDets=100]	55.2	54.8	14.4	0.322
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets= 1]	60.4	60.6	36.7	0.454
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets= 10]	64.1	66.1	41.6	0.526
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets=100]	64.3	66.2	41.7	0.527
	Avg. Rec. (AR) @[IoU=50:95 | area= small | maxDets=100]	34.2	38.8	-1	0.188
	Avg. Rec. (AR) @[IoU=50:96 | area=medium | maxDets=100]	76.8	78.3	23.5	0.537
	Avg. Rec. (AR) @[IoU=50:97 | area= large | maxDets=100]	85.8	87.1	41.8	0.758
SSD	Avg. Prec. (AP) @[IoU=50:95 | area= all | maxDets=100]	27.4	26	10	0.134
	Avg. Prec. (AP) @[IoU=50 | area= all | maxDets=100]	36.7	33.4	14.2	0.189
	Avg. Prec. (AP) @[IoU=75 | area= all | maxDets=100]	32.3	30.4	11.2	0.149
	Avg. Prec. (AP) @[IoU=50:95 | area= small | maxDets=100]	7	4.6	-1	0.001
	Avg. Prec. (AP) @[IoU=50:95 | area=medium | maxDets=100]	31.4	29.3	0	0.029
	Avg. Prec. (AP) @[IoU=50:95 | area= large | maxDets=100]	45.1	45.2	10.8	0.257
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets= 1]	45.8	43.1	20	0.288
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets= 10]	47.3	44.6	21.5	0.317
	Avg. Rec. (AR) @[IoU=50:95 | area= all | maxDets=100]	47.5	44.7	21.7	0.321
	Avg. Rec. (AR) @[IoU=50:95 | area= small | maxDets=100]	16.4	11.5	-1	0.08
	Avg. Rec. (AR) @[IoU=50:96 | area=medium | maxDets=100]	57.7	53.8	5.7	0.243
	Avg. Rec. (AR) @[IoU=50:97 | area= large | maxDets=100]	71.3	71.1	21.8	0.496

Figure 14: Complete results of the context analysis experiments.

G ERROR DIAGNOSIS BASED ON OBJECT SIZE

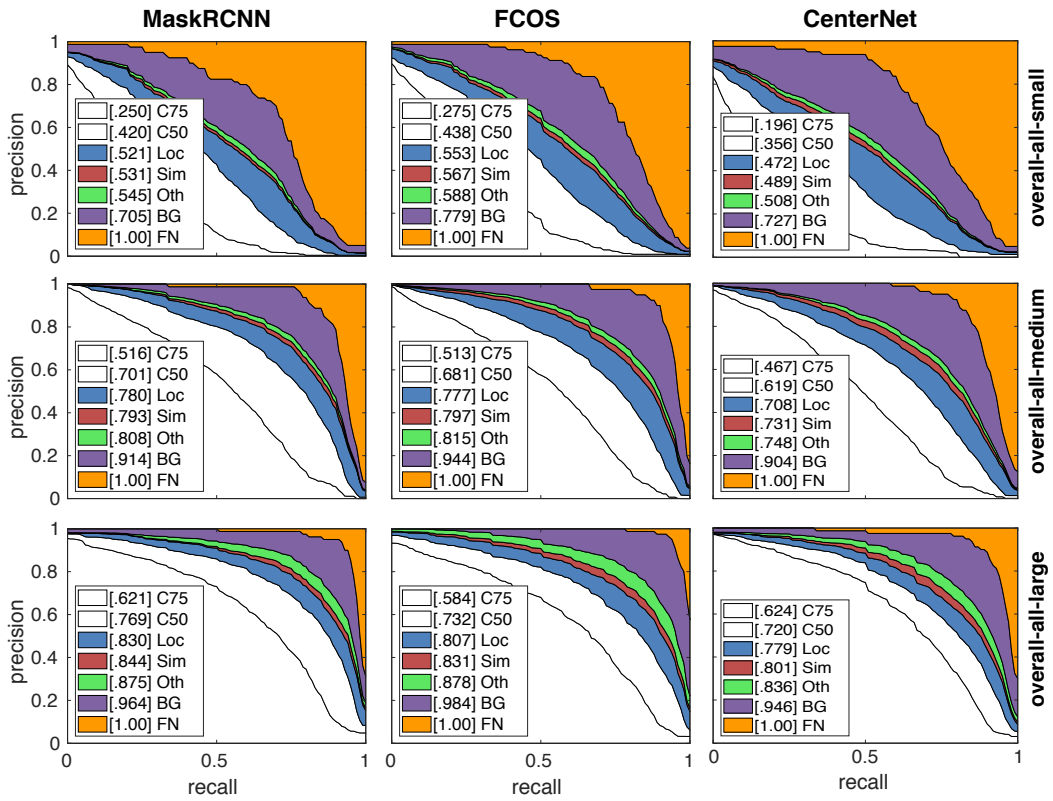


Figure 15: Error analysis of models based on the object size over the MS COCO dataset.

H DETAILED UAP AND MODEL AP OVER DATASETS

```
% ----------------  Per-category upper-bound AP results  -------------


% COCO dataset
~~~~ Mean and per-category AP @ IoU=[0.50,0.95] ~~~~
78.2
person:96.7
bicycle:85.0
car:86.8
motorcycle:89.4
airplane:94.4
bus:81.8
train:89.0
truck:57.3
boat:78.5
traffic light:83.9
fire hydrant:88.7
stop sign:92.1
parking meter:75.0
bench:64.0
bird:75.8
cat:89.6
dog:81.6
horse:84.5
sheep:84.6
cow:82.4
elephant:93.6
bear:80.6
zebra:97.9
giraffe:97.0
backpack:51.6
umbrella:85.0
handbag:62.1
tie:76.5
suitcase:67.3
frisbee:87.2
skis:84.9
snowboard:60.1
sports ball:79.3
kite:80.6
baseball bat:80.0
baseball glove:67.0
skateboard:86.5
surfboard:76.7
tennis racket:87.9
bottle:83.2
wine glass:72.8
cup:72.7
fork:74.8
knife:65.4
spoon:59.0
bowl:79.4
banana:85.2
apple:68.1
sandwich:57.5
orange:79.7
broccoli:94.0
```

```
carrot:81.1
hot dog:63.0
pizza:82.9
donut:79.0
cake:67.6
chair:76.7
couch:59.9
potted plant:86.7
bed:70.2
dining table:74.1
toilet:83.7
tv:88.0
laptop:76.1
mouse:82.8
remote:79.8
keyboard:86.6
cell phone:69.6
microwave:83.6
oven:79.1
toaster:50.0
sink:85.3
refrigerator:85.4
book:85.2
clock:88.4
vase:64.5
scissors:73.7
teddy bear:81.9
hair drier:49.4
toothbrush:62.1

~~~~ Summary metrics ~~~~
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.782
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.782
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.782
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.635
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.816
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.846
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.483
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.797
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.812
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.663
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.843
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.893
```

```
% FASHION dataset
~~~~ Mean and per-category AP @ IoU=[0.50,0.95] ~~~~
71.2
suitcoats_blazers:79.3
hoodies:56.1
shoes:88.5
messengerbags:32.3
jeans:79.2
```

```
tanks_camis:77.6
tunics:25.4
coats_jackets:74.3
sunhats_cowboyhats:94.5
handbags:86.4
scarves_wraps:74.7
sweaters:58.6
dresses:92.1
pants:71.8
clutches:76.8
shorts:85.8
leggings:59.1
boots:79.8
jumpsuits:75.2
sandals:64.5
tees:72.4
beanieknitcaps:91.4
slippers:53.1
blouses:48.0
skirts:92.5
glasses:97.7
watches:97.6
henleys:16.6
buttondowns:77.2
berets:66.0
longsleeveshirts:42.8
ties:95.6
backpacks:88.9
rompers:59.1
baseballcaps:84.0
overalls:63.9
vests:67.1
polos:80.3
cardigans:22.4
humans:97.9

~~~~ Summary metrics ~~~~
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.712
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.712
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.712
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.457
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.614
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.721
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.662
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.767
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.774
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.504
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.660
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.782
```

```
% PASCAL VOC dataset
~~~~ Mean and per-category AP @ IoU=[0.50,0.95] ~~~~
91.6
aeroplane:97.9
bicycle:94.4
bird:87.8
boat:89.6
bottle:93.6
bus:92.4
car:96.6
cat:94.7
chair:88.4
cow:87.6
diningtable:86.9
dog:89.6
horse:88.9
motorbike:92.4
person:96.7
pottedplant:94.9
sheep:93.0
sofa:74.8
train:96.0
tvmonitor:95.0

~~~~ Summary metrics ~~~~
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.916
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.916
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.916
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.707
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.861
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.941
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.579
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.908
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.930
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.736
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.877
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.954
```

```
---------------  Performance of the models on the original images  -------------

---------------  COCO dataset  -------------

% FasterRCNN
~~~~ Mean and per-category AP @ IoU=[0.50,0.95] ~~~~
36.4
person:51.2
bicycle:27.1
car:40.5
motorcycle:38.0
airplane:58.6
bus:58.4
train:53.5
truck:30.6
boat:25.3
traffic light:26.0
fire hydrant:62.0
stop sign:63.4
parking meter:42.2
bench:20.6
bird:31.2
cat:55.3
dog:53.3
horse:51.6
sheep:45.4
cow:51.3
elephant:56.2
bear:60.2
zebra:60.6
giraffe:61.1
backpack:14.2
umbrella:32.5
handbag:11.5
tie:29.4
suitcase:32.6
frisbee:60.5
skis:19.8
snowboard:28.7
sports ball:41.2
kite:37.6
baseball bat:22.0
baseball glove:31.9
skateboard:44.4
surfboard:33.0
tennis racket:41.3
bottle:36.1
wine glass:32.3
cup:38.5
fork:25.0
knife:12.9
spoon:11.6
bowl:38.7
banana:19.6
apple:17.9
sandwich:31.5
orange:28.3
broccoli:21.6
carrot:18.6
```

```
hot dog:26.1
pizza:45.4
donut:41.7
cake:32.1
chair:23.4
couch:35.4
potted plant:24.1
bed:33.4
dining table:23.0
toilet:51.3
tv:51.3
laptop:53.2
mouse:57.8
remote:25.4
keyboard:47.3
cell phone:32.0
microwave:51.6
oven:26.5
toaster:42.8
sink:33.0
refrigerator:43.9
book:13.6
clock:47.9
vase:33.3
scissors:19.4
teddy bear:40.1
hair drier:3.3
toothbrush:16.2

~~~~ Summary metrics ~~~~
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.364

Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.584

Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.391

Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.215

Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.400

Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.466

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.304

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.489

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.514

Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.324

Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.554

Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.645
```

```
% FCOS
~~~~ Mean and per-category AP @ IoU=[0.50,0.95] ~~~~
42.8
person:55.2
bicycle:31.1
car:44.1
motorcycle:40.9
airplane:66.8
bus:69.2
train:65.4
truck:40.0
boat:27.9
traffic light:27.7
fire hydrant:68.8
stop sign:66.8
parking meter:48.9
bench:24.4
bird:39.0
cat:72.7
dog:66.0
horse:60.5
sheep:52.7
cow:61.1
elephant:65.8
bear:75.2
zebra:68.3
giraffe:68.5
backpack:17.5
umbrella:40.2
handbag:17.7
tie:33.4
suitcase:41.2
frisbee:67.6
skis:22.8
snowboard:33.9
sports ball:45.2
kite:42.7
baseball bat:28.2
baseball glove:37.8
skateboard:54.7
surfboard:37.0
tennis racket:49.2
bottle:40.3
wine glass:38.5
cup:45.1
fork:32.8
knife:16.9
spoon:15.8
bowl:41.1
banana:22.1
apple:22.4
sandwich:34.6
orange:32.2
broccoli:20.9
carrot:21.5
hot dog:34.5
pizza:53.5
donut:49.8
cake:38.3
```

```
chair:29.8
couch:42.7
potted plant:27.7
bed:39.6
dining table:24.5
toilet:61.7
tv:56.4
laptop:59.8
mouse:62.0
remote:34.0
keyboard:48.8
cell phone:38.0
microwave:60.3
oven:36.2
toaster:49.2
sink:36.6
refrigerator:56.5
book:14.4
clock:50.8
vase:37.7
scissors:30.6
teddy bear:50.8
hair drier:17.8
toothbrush:23.4

~~~~ Summary metrics ~~~~
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.428

Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.626

Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.457

Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.265

Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.469

Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.545

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.345

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.552

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.582

Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.388

Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.628

Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.735
```

```
% RetinaNet
~~~~ Mean and per-category AP @ IoU=[0.50,0.95] ~~~~
40
person:51.7
bicycle:30.4
car:41.3
motorcycle:42.4
airplane:61.6
bus:65.0
train:61.9
truck:37.8
boat:25.7
traffic light:26.5
fire hydrant:65.8
stop sign:65.1
parking meter:50.0
bench:24.5
bird:34.4
cat:68.3
dog:65.4
horse:58.2
sheep:50.4
cow:56.4
elephant:62.8
bear:70.0
zebra:64.0
giraffe:64.7
backpack:17.8
umbrella:38.0
handbag:15.8
tie:30.0
suitcase:36.0
frisbee:64.3
skis:20.4
snowboard:25.5
sports ball:42.8
kite:38.4
baseball bat:25.5
baseball glove:34.6
skateboard:51.6
surfboard:33.7
tennis racket:46.5
bottle:36.7
wine glass:36.1
cup:43.2
fork:28.6
knife:14.3
spoon:14.0
bowl:40.7
banana:24.1
apple:21.6
sandwich:34.9
orange:31.1
broccoli:21.9
carrot:19.9
hot dog:32.7
pizza:49.9
donut:45.4
```

```
cake:36.3
chair:26.7
couch:42.1
potted plant:25.0
bed:41.0
dining table:25.9
toilet:60.4
tv:56.7
laptop:58.5
mouse:62.0
remote:29.0
keyboard:47.4
cell phone:36.7
microwave:56.1
oven:33.1
toaster:21.5
sink:36.4
refrigerator:51.2
book:14.0
clock:50.1
vase:35.7
scissors:31.1
teddy bear:44.0
hair drier:3.7
toothbrush:17.5

~~~~ Summary metrics ~~~~
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.400

Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.609

Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.430

Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.235

Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.444

Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.526

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.328

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.522

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.555

Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.361

Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.599

Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.704
```

```
% SSD
~~~~ Mean and per-category AP @ IoU=[0.50,0.95] ~~~~
29.3
person:40.4
bicycle:22.0
car:30.5
motorcycle:31.5
airplane:50.1
bus:55.9
train:55.7
truck:27.8
boat:16.3
traffic light:14.7
fire hydrant:49.7
stop sign:50.6
parking meter:35.0
bench:16.3
bird:23.7
cat:53.0
dog:51.6
horse:46.8
sheep:37.9
cow:42.6
elephant:52.9
bear:60.3
zebra:55.1
giraffe:54.2
backpack:7.7
umbrella:28.1
handbag:6.7
tie:17.6
suitcase:21.9
frisbee:41.7
skis:13.3
snowboard:20.2
sports ball:29.7
kite:26.2
baseball bat:14.9
baseball glove:21.2
skateboard:35.9
surfboard:23.5
tennis racket:30.2
bottle:20.3
wine glass:20.4
cup:28.6
fork:15.1
knife:6.2
spoon:5.6
bowl:31.3
banana:14.8
apple:13.9
sandwich:28.5
orange:23.4
broccoli:17.0
carrot:13.0
hot dog:24.0
```

```
pizza:41.2
donut:32.3
cake:25.8
chair:17.3
couch:33.7
potted plant:16.6
bed:35.8
dining table:22.8
toilet:51.3
tv:46.7
laptop:49.4
mouse:44.3
remote:12.5
keyboard:37.8
cell phone:24.3
microwave:46.4
oven:27.8
toaster:7.0
sink:25.4
refrigerator:40.5
book:7.4
clock:37.8
vase:22.8
scissors:22.6
teddy bear:33.9
hair drier:0.2
toothbrush:7.1

~~~~ Summary metrics ~~~~
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.293

Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.492

Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.308

Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.118

Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.341

Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.447

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.264

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.400

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.425

Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.173

Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.488

Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.607
```

```
% ---------------  FASHION  dataset  -------------

% FCOS
~~~~ Mean and per-category AP @ IoU=[0.50,0.95] ~~~~
59.7
suitcoats_blazers:74.2
hoodies:53.9
shoes:72.0
messengerbags:36.3
jeans:69.4
tanks_camis:63.9
tunics:43.3
coats_jackets:63.7
sunhats_cowboyhats:80.8
handbags:59.1
scarves_wraps:39.6
sweaters:57.9
dresses:88.1
pants:68.5
clutches:55.8
shorts:66.0
leggings:51.9
boots:70.3
jumpsuits:69.9
sandals:64.7
tees:66.4
beanieknitcaps:72.7
slippers:51.2
blouses:58.3
skirts:85.6
glasses:69.8
watches:57.9
henleys:5.0
buttondowns:65.3
berets:39.7
longsleeveshirts:51.6
ties:59.3
backpacks:74.9
rompers:63.1
baseballcaps:57.4
overalls:49.3
vests:34.4
polos:75.2
cardigans:13.2
humans:88.1

~~~~ Summary metrics ~~~~
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.597

Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.711

Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.647

Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.182

Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.376
```

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.627

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.692

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.822

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.824

Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.303

Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.639

Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.850
```

```
% MaskRCNN
~~~~ Mean and per-category AP @ IoU=[0.50,0.95] ~~~~
54.1
suitcoats_blazers:72.8
hoodies:47.0
shoes:66.1
messengerbags:29.3
jeans:64.8
tanks_camis:57.6
tunics:40.3
coats_jackets:65.8
sunhats_cowboyhats:71.9
handbags:52.1
scarves_wraps:31.9
sweaters:54.1
dresses:83.9
pants:62.0
clutches:46.9
shorts:59.1
leggings:43.4
boots:62.2
jumpsuits:66.4
sandals:57.8
tees:60.6
beanieknitcaps:60.7
slippers:42.1
blouses:52.6
skirts:78.8
glasses:64.7
watches:51.3
henleys:7.7
buttondowns:60.6
berets:24.8
longsleeveshirts:45.5
ties:56.6
backpacks:66.2
rompers:54.2
baseballcaps:48.5
```

```
overalls:43.9
vests:37.0
polos:68.3
cardigans:19.3
humans:86.2

~~~~ Summary metrics ~~~~
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.541

Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.698

Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.614

Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.108

Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.315

Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.570

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.618

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.712

Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.714

Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.194

Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.499

Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.742
```

```
% CenterNet
~~~~ Mean and per-category AP @ IoU=[0.50,0.95] ~~~~
54
suitcoats_blazers:72.4
hoodies:48.8
shoes:65.2
messengerbags:25.1
jeans:63.8
tanks_camis:58.8
tunics:39.8
coats_jackets:66.2
sunhats_cowboyhats:71.7
handbags:56.5
scarves_wraps:36.0
sweaters:55.4
dresses:78.4
pants:58.5
clutches:48.2
shorts:59.5
leggings:40.0
boots:62.5
```

```
jumpsuits:60.1
sandals:53.5
tees:62.4
beanieknitcaps:58.0
slippers:41.9
blouses:51.8
skirts:80.7
glasses:62.7
watches:47.4
henleys:9.7
buttondowns:61.5
berets:33.6
longsleeveshirts:45.6
ties:54.7
backpacks:68.8
rompers:59.7
baseballcaps:49.8
overalls:46.0
vests:36.0
polos:67.0
cardigans:20.8
humans:79.8

~~~~ Summary metrics ~~~~
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.540
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.681
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.589
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.124
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.318
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.568
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.649
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.815
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.817
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.269
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.577
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.846
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.303
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.639
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.850
```

```
% ----------------  VOC results using COCO evaluation code -------------

% Upper bound
Average Precision  (AP) @[ IoU=50:95 | area=   all | maxDets=100 ] = 91.6
Average Precision  (AP) @[ IoU=50      | area=   all | maxDets=100 ] = 91.6
Average Precision  (AP) @[ IoU=75      | area=   all | maxDets=100 ] = 91.6
Average Precision  (AP) @[ IoU=50:95 | area= small | maxDets=100 ] = 70.7
Average Precision  (AP) @[ IoU=50:95 | area=medium | maxDets=100 ] = 86.1
Average Precision  (AP) @[ IoU=50:95 | area= large | maxDets=100 ] = 94.1
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets=  1 ] = 57.9
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets= 10 ] = 90.8
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets=100 ] = 93.0
Average Recall     (AR) @[ IoU=50:95 | area= small | maxDets=100 ] = 73.6
Average Recall     (AR) @[ IoU=50:95 | area=medium | maxDets=100 ] = 87.7
Average Recall     (AR) @[ IoU=50:95 | area= large | maxDets=100 ] = 95.4


% CenterNet
Average Precision  (AP) @[ IoU=50:95 | area=   all | maxDets=100 ] = 47.8
Average Precision  (AP) @[ IoU=50      | area=   all | maxDets=100 ] = 72.7
Average Precision  (AP) @[ IoU=75      | area=   all | maxDets=100 ] = 51.3
Average Precision  (AP) @[ IoU=50:95 | area= small | maxDets=100 ] = 07.4
Average Precision  (AP) @[ IoU=50:95 | area=medium | maxDets=100 ] = 26.2
Average Precision  (AP) @[ IoU=50:95 | area= large | maxDets=100 ] = 61.0
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets=  1 ] = 40.2
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets= 10 ] = 57.2
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets=100 ] = 58.4
Average Recall     (AR) @[ IoU=50:95 | area= small | maxDets=100 ] = 18.6
Average Recall     (AR) @[ IoU=50:95 | area=medium | maxDets=100 ] = 40.9
Average Recall     (AR) @[ IoU=50:95 | area= large | maxDets=100 ] = 70.5


% FCOS
Average Precision  (AP) @[ IoU=50:95 | area=   all | maxDets=100 ] = 47.9
Average Precision  (AP) @[ IoU=50      | area=   all | maxDets=100 ] = 71.0
Average Precision  (AP) @[ IoU=75      | area=   all | maxDets=100 ] = 51.4
Average Precision  (AP) @[ IoU=50:95 | area= small | maxDets=100 ] = 11.1
Average Precision  (AP) @[ IoU=50:95 | area=medium | maxDets=100 ] = 32.1
Average Precision  (AP) @[ IoU=50:95 | area= large | maxDets=100 ] = 58.4
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets=  1 ] = 41.2
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets= 10 ] = 58.5
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets=100 ] = 59.5
Average Recall     (AR) @[ IoU=50:95 | area= small | maxDets=100 ] = 19.5
Average Recall     (AR) @[ IoU=50:95 | area=medium | maxDets=100 ] = 45.2
Average Recall     (AR) @[ IoU=50:95 | area= large | maxDets=100 ] = 70.2


% MASK RCNN  / FasterRCNN
Average Precision  (AP) @[ IoU=50:95 | area=   all | maxDets=100 ] = 47.3
Average Precision  (AP) @[ IoU=50      | area=   all | maxDets=100 ] = 71.3
Average Precision  (AP) @[ IoU=75      | area=   all | maxDets=100 ] = 52.6
Average Precision  (AP) @[ IoU=50:95 | area= small | maxDets=100 ] = 08.6
Average Precision  (AP) @[ IoU=50:95 | area=medium | maxDets=100 ] = 30.7
Average Precision  (AP) @[ IoU=50:95 | area= large | maxDets=100 ] = 58.1
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets=  1 ] = 40.3
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets= 10 ] = 53.8
Average Recall     (AR) @[ IoU=50:95 | area=   all | maxDets=100 ] = 54.1
Average Recall     (AR) @[ IoU=50:95 | area= small | maxDets=100 ] = 11.2
```

```
Average Recall     (AR) @[ IoU=50:95 | area=medium | maxDets=100 ] = 36.9
Average Recall     (AR) @[ IoU=50:95 | area= large | maxDets=100 ] = 65.7


% ----------------  VOC results using VOC evaluation code -------------


% Upper bound
Evaluating detections
VOC07 metric? Yes
AP for aeroplane = 0.9091
AP for bicycle = 0.9033
AP for bird = 0.9065
AP for boat = 0.8951
AP for bottle = 0.9056
AP for bus = 0.9039
AP for car = 0.9052
AP for cat = 0.9062
AP for chair = 0.8722
AP for cow = 0.8933
AP for diningtable = 0.8968
AP for dog = 0.8950
AP for horse = 0.8969
AP for motorbike = 0.9054
AP for person = 0.9066
AP for pottedplant = 0.9085
AP for sheep = 0.9035
AP for sofa = 0.7672
AP for train = 0.9087
AP for tvmonitor = 0.9012
Mean AP = 0.8945


% FCOS
Evaluating detections
VOC07 metric? Yes
AP for aeroplane = 0.8701
AP for bicycle = 0.8454
AP for bird = 0.7722
AP for boat = 0.6895
AP for bottle = 0.6709
AP for bus = 0.8371
AP for car = 0.8716
AP for cat = 0.8704
AP for chair = 0.6213
AP for cow = 0.8362
AP for diningtable = 0.6900
AP for dog = 0.8572
AP for horse = 0.8414
AP for motorbike = 0.7966
AP for person = 0.8430
AP for pottedplant = 0.5464
AP for sheep = 0.8107
AP for sofa = 0.7679
AP for train = 0.8454
AP for tvmonitor = 0.7735
Mean AP = 0.7829
```

```
% FasterRcnn
Evaluating detections
VOC07 metric? Yes
AP for aeroplane = 0.8147
AP for bicycle = 0.8656
AP for bird = 0.7855
AP for boat = 0.6552
AP for bottle = 0.6848
AP for bus = 0.8660
AP for car = 0.8821
AP for cat = 0.8811
AP for chair = 0.6376
AP for cow = 0.8527
AP for diningtable = 0.7977
AP for dog = 0.8691
AP for horse = 0.8816
AP for motorbike = 0.8602
AP for person = 0.8008
AP for pottedplant = 0.5136
AP for sheep = 0.7870
AP for sofa = 0.8221
AP for train = 0.8543
AP for tvmonitor = 0.7778
Mean AP = 0.7945
```

```
% SSD12
Evaluating detections
VOC07 metric? Yes
AP for aeroplane = 0.8388
AP for bicycle = 0.8619
AP for bird = 0.8119
AP for boat = 0.7161
AP for bottle = 0.6090
AP for bus = 0.8730
AP for car = 0.8910
AP for cat = 0.8951
AP for chair = 0.6361
AP for cow = 0.8652
AP for diningtable = 0.7550
AP for dog = 0.8681
AP for horse = 0.8772
AP for motorbike = 0.8373
AP for person = 0.8215
AP for pottedplant = 0.5666
AP for sheep = 0.8127
AP for sofa = 0.7970
AP for train = 0.8723
AP for tvmonitor = 0.7855
Mean AP = 0.7996
```