Black-box Explanation of Object Detectors via Saliency Maps

Vitali Petsiuk[1⋆], Rajiv Jain[2], Varun Manjunatha[2], Vlad I. Morariu[2], Ashutosh Mehra[3], Vicente Ordonez[4], and Kate Saenko[1]

[1] Boston University
[2] Adobe Research
[3] Adobe Document Cloud
[4] University of Virginia

Abstract. We propose D-RISE, a method for generating visual explanations for the predictions of object detectors. D-RISE can be considered "black-box" in the software testing sense, it only needs access to the inputs and outputs of an object detector. Compared to gradient-based methods, D-RISE is more general and agnostic to the particular type of object detector being tested as it does not need to know about the inner workings of the model. We show that D-RISE can be easily applied to different object detectors including one-stage detectors such as YOLOv3 and two-stage detectors such as Faster-RCNN. We present a detailed analysis of the generated visual explanations to highlight the utilization of context and the possible biases learned by object detectors.

1 Introduction

Although object detectors have experienced significant gains in performance since the adoption of deep neural networks (DNNs) [10], DNNs remain opaque tools with a complex and unintuitive process of decision-making, making them hard to understand, debug and improve. A number of different explanation techniques offer potential solutions to these issues. They have already been shown to find biases in trained models [36], help debug them [13], increase user's trust [31]. A popular approach to explanation involves the use of attribution techniques which produce saliency maps [19], [32], *i.e.*, heatmaps representing the influence different pixels have on the model's decision. Unfortunately, these techniques have primarily focused on the image classification task [26], [9], [31], [40], [2], [38], with few addressing other problems such as visual question answering [24], video captioning [27], [3] and video activity recognition [3]. We address the relatively underexplored direction of generating saliency maps for object detectors.

Unlike methods that explain the emerging patterns in the learned weights or activations [4], [38], [21], attribution techniques are task-specific and can only be applied to the inference pipeline for which they have been developed. Therefore,

[⋆] Work done while an intern at Adobe.

Fig. 1: D-RISE can highlight which regions of an image were used by an object detector. Here we show outputs for a few corresponding images where importance increases from blue to red. In these examples, D-RISE reveals things such as detectors often looking outside bounding boxes to detect objects e.g., looking at the tap to predict the sink, or looking disproportionately to a subset of regions within the object e.g., looking at the Apple logo on the laptop.

none of the mentioned attribution methods can be directly applied to new tasks, such as object detection. Inspired by the RISE [26] method of generating saliency maps to explain visual classifiers, we propose D-RISE, a method to produce saliency maps for object detectors that does not rely on gradients or the inner workings of the underlying object detector. The advantage of this approach is that our method can be in principle applied to any object detector. Extending methods that produce saliency-based visual explanations to object detectors is non-trivial as this task requires explanations for each object instance as opposed to a global explanation for each image. Moreover, explanations must take into account both class predictions and bounding box predictions.

Explaining visual classifiers with saliency maps has allowed researchers to investigate the localization abilities implicitly learned by these models. Some works have also used explanations of visual classifiers for weakly-supervised object localization [14], [23]. In object detection, however, the localization decisions of the model are explicit as they are expressed directly in the outputs of the model. Therefore, one might assume that exploring spatial importance in this case is redundant, that the model has already predicted bounding boxes around everything it deems important. In our experiments with D-RISE, we observe that DNN models also learn to utilize contextual regions to detect objects. For instance the last column in Fig. 1 shows how the tap helps to localize the sink even when it is clearly outside the detected box. In fact, the importance of contextual information for object detection has long been established for both humans [5], [22] and machines [33], [20]. Another reason for studying object detector's saliency is that not all sub-regions within the object's bounding box are equally important. Some object parts are more discriminant, while others may occur with objects of different categories, *e.g.*, cat faces are highlighted as more important by the network than its body (third column in Fig. 1).

Our contributions can be summarized as follows:

- We propose D-RISE, a black-box attribution technique for explaining object detectors via saliency maps, by significantly extending the idea of exhaustively and systematically exploring areas that affect the prediction outputs of the underlying model by manipulating its inputs.
- We demonstrate D-RISE for explaining two commonly used object detectors with different architectural designs, namely one-stage YOLOv3 [28] and two-stage Faster R-CNN [29].
- We systematically analyze using D-RISE potential sources of errors and bias in commonly used object detectors trained on the COCO dataset and discovering common patterns in data picked up by the model.
- We propose additional evaluation by producing synthetically modified datasets by inserting markers that deliberately introduce biases in the dataset that are effectively discovered by D-RISE.

2 Related Work

2.1 Object Detection

Object detectors can be divided into two groups: two-stage detectors, with the Faster R-CNN [29] being the most representative, and one-stage detectors, such as YOLO [28], SSD [18] and CornerNet [16]. Two-stage detectors consist of a region proposal stage, where a sparse set of regions of interest (ROI) is selected, followed by a feature extraction stage for the subsequent classification of each candidate ROI. One-stage methods do not perform ROI pooling and instead use a single network to detect objects. They are faster than two-stage detectors, but have lower localization accuracy. We use our saliency technique to analyze both two-stage and single-stage detectors (Faster R-CNN and YOLO, respectively).

Previous works on explaining DNN-based object detectors include their feature space visualization [35], analyzing the biases, such as pedestrians' skin color [37]. Two recent works have explored explainability using saliency for SSD object detectors [34], [11]. These methods rely on tailored white-box approaches in comparison to D-RISE, which treats detectors as black boxes.

2.2 Explainable Artificial Intelligence

Various visual grounding techniques retroactively provide interpretability to computer vision models after they have been trained. The first major line of works backpropagates the importance score through the layers of the neural network from the model's output to the individual pixels in the input, e.g. Excitation Backprop [39] and Layer-wise Relevance Propagation [2]. Grad-CAM [31], a generalization of CAM [40], computes the regular gradients up to a selected intermediate layer and then combines them with the corresponding activations to get a low-resolution saliency map. A disadvantage of these methods is that they are usually tailored to the model's architecture, *i.e.* they cannot be used for new network architectures without implementing new layers. However, these methods are very fast.

The second line of works performs specific perturbations on image regions, such as occlusion, adding noise, inpainting, and blurring. After observing the effect of a perturbation on the model's output, the methods determine the importance of the region that was perturbed. Specifically, Occlusion [38] blocks out square parts of the image in a sliding window manner and captures the drop in the class score to determine the importance. LIME [30] approximates the deep model by a linear classifier, trains it in the vicinity of the input point by using samples with occluded superpixels and uses the learned weights as the measure of superpixel importance. The Meaningful Perturbation approach [9] and Real Time Image Saliency [7] optimize the perturbation mask using gradient descent. [5] Finally, RISE [26] generates a set of random masks, applies the classifier to masked versions of the input and uses the predicted class probabilities as weights, computing a weighted sum of the masks as the saliency map. We use this masking technique to explain object detectors rather than image classifiers. Since this and other saliency methods described above cannot be directly applied to the detection task, our work extends the prior state of the art by enabling detector explanation. We describe key differences that have to be addressed for extending saliency methods to object detection in the next section.

Black-box and white-box approaches have slightly different use cases. Black-box methods, while typically slower at run time, can save developer time due to their higher generalizability and ease of application. They also enable analysis of proprietary models or APIs which cannot be studied using white-box approaches. Arguably, black-box methods are more intuitive, because they directly measure the effect that input ablations have on the model, without relying on heuristics as rules for importance back-propagation. On the other hand, faster white-box methods are more suitable for large scale and real-time applications.

3 Method

Given an h-by-w image I, a DNN detector model f, and an object detection d specified by a bounding box and a category label, our goal is to produce a saliency map S to explain the detection. The map consists of h-by-w values indicating the importance of each pixel in I in influencing f to predict d. We propose *Detector Randomized Input Sampling for Explanation*, or D-RISE, to solve this problem in a black-box manner, *i.e.*, without access to f's weights, gradients or architecture. Our method is inspired by the randomized perturbations (masks) applied to the image by the RISE model to explain object classifiers, except that we leverage the random-masking idea to explain object detectors. The main idea is to measure the effect of masking randomized regions on the predicted output, using changes in f's output to determine the importance. Figure 2 shows an overview of our approach.

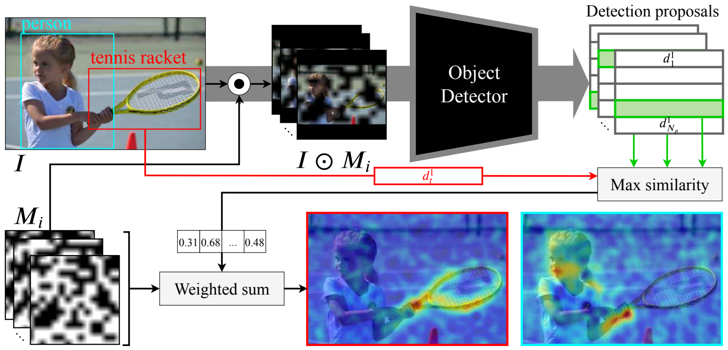


Fig. 2: Our method D-RISE attempts to explain the detections (bounding-box+category) produced for this image by an object detector. We convert target detections that need to be explained into detection vectors d_t. We sample N binary masks, M_i, and run the detector on the masked images to obtain proposals D_p. We compute pairwise similarities between targets and proposals to obtain weights for each mask. Finally, the weighted sum of masks gives us saliency maps.

Existing approaches for image classification saliency cannot be directly applied to the object detection task. They assume a single categorical model output, while object detectors produce a multitude of detection vectors that encode class probabilities, localization information and additional information such as an objectness score. To apply random masking to detectors, we incorporate localization and objectness scores into the process of generating detector saliency maps.

Most detector networks, including Faster R-CNN and YOLO, produce a large number of bounding box proposals which are subsequently refined using confidence thresholding and non-maximum suppression to leave a small number of final detections. We denote such bounding box proposals in the following manner:

$$d_i = [L_i, O_i, P_i] \tag{1}$$
$$= [(x_1^i, y_1^i, x_2^i, y_2^i), O_i, (p_1^i, \dots, p_C^i)] \tag{2}$$

Each proposal is encoded into a detection vector d_i consisting of

- localization information L_i, defining bounding box corners (x_1^i, y_1^i) and (x_2^i, y_2^i),
- objectness score $O_i \in [0, 1]$, representing the probability that bounding box L_i contains an object of any class (if the detector does not produce such a score this term may be ignored), and
- classification information P_i — a vector of probabilities $(p_1^i, \dots, p_C^i)$ representing the probability that region L_i belongs to each of C classes.

We construct a detection vector for any given bounding box and its label by taking the corners of the bounding box, setting O_i to 1 and using a one-hot vector for the probabilities.

Given an object detector f, an image I and a categorized bounding box (not necessarily produced by the model) we generate a saliency map that would highlight regions important for the model in order to predict such a bounding box. If the detection actually comes from the model, we treat the generated heatmap as an explanation for model's decision. Following the perturbation-based attribution paradigm, we measure the importance of a region by observing the effect that perturbation of this region has on the detector's output.

In contrast with classification models, object detection models are designed and trained with regression objectives and do not have a single proposal directly corresponding to any arbitrary bounding box with particular coordinates. Instead, many proposals are produced, with bounding boxes that differ and overlap to varying degrees with the bounding box provided as input to the explanation algorithm. Therefore, for object detection it is important to determine not just *how* we measure the disturbance in the output but also *where* we measure it in terms of which disturbances do we select from among the proposals produced by a network. To measure the disturbance in the output (the *how*), we develop a similarity metric s for the detection proposal vectors (Sec. 3.2). To account for the *where*, we measure the output disturbance caused by an individual mask by selecting the proposal with maximum pairwise similarity between the target detection vector and all detection proposal vectors produced for a masked image. More precisely, following our notation,

$$S(d_t, f(M_i \odot I)) \triangleq \max_{d_j \in f(M_i \odot I)} s(d_t, d_j), \tag{3}$$

where S denotes the similarity between target detection vector d_t and new detection proposals for the modified image. This allows us to use the RISE masking technique to produce saliency maps for explaining object detector decisions. Note, that this framework does not restrict d_t to be directly produced by the model. For that reason our method can produce explanations for arbitrary detection vectors, such as objects missed by the detector. Gradient-based methods would not be able to do this, because there's no starting point to propagate from.

3.1 Mask generation

We adopt the following mask generation procedure from RISE [26].

1. Sample N binary masks of size $h \times w$ (smaller than image size $H \times W$) by setting each element independently to 1 with probability p and to 0 with the remaining probability.
2. Upsample all masks to size $(h+1)C_H \times (w+1)C_W$ using bilinear interpolation, where $C_H \times C_W = \lfloor H/h \rfloor \times \lfloor W/w \rfloor$ is the size of the cell in the upsampled mask.
3. Crop areas $H \times W$ with uniformly random offsets ranging from $(0, 0)$ up to (C_H, C_W).

3.2 Similarity metric

To compute the similarity score between the target vector and the proposal vector, all three components should be considered. We use *Intersection over Union* (IoU) to measure the spatial proximity of the bounding boxes encoded by two vectors. To evaluate how similar two regions look to the network, we use the *cosine similarity* of the class probabilities associated with the regions. Finally, for the networks that explicitly compute an objectness score, such as YOLOv3 [28], we incorporate a measure of the similarity of the objectness scores into the metric, as well. In our experiments we only explain high confidence detections, *i.e.*, we set $O_t = 1$, so to incorporate objectness score into the similarity metric we simply multiply it by O_j. As a result, detection proposals with lower objectness score will have lower similarity with a high confidence target vector. If the network does not produce an objectness score, *e.g.*, Faster R-CNN [29], the objectness term can be simply omitted. Thus, the similarity score between two detection vectors can be decomposed into three scalar factors:

$$s(d_t, d_j) = s_L(d_t, d_j) \cdot s_P(d_t, d_j) \cdot s_O(d_t, d_j), \tag{4}$$

where

$$s_L(d_t, d_j) = \mathrm{IoU}(L_t, L_j), \tag{5}$$

$$s_P(d_t, d_j) = \frac{P_t \cdot P_j}{\|P_t\| \|P_j\|}, \tag{6}$$

$$s_O(d_t, d_j) = O_j. \tag{7}$$

Scalar product has been chosen to model logical AND of three similarity values, with the desired property that if one of them is low, the total similarity value is also low.

3.3 Saliency inference

We now formulate the full process of generating saliency maps using D-RISE.

1. Generate N RISE masks, $M = \{M_i,\ 1 \le i \le N\}$. M has shape $N \times H \times W$.
2. Convert the target detections to be explained into detection vectors, $D_t = \{d_t,\ 1 \le t \le T\}$. D_t has shape $T \times (4 + 1 + C)$. We can run the detector on masked images only once to get the saliency maps for all T detections.
3. Run the detector f on masked images $I \odot M_i$ to get the proposals, $D_p = \{D_p^i, 1 \le i \le N\} = \{f(M_i \odot I), 1 \le i \le N\} = \{d_j^i, 1 \le i \le N, 1 \le j \le N_p\}$. D_p has shape $N \times N_p \times (4 + 1 + C)$.
4. Compute pairwise similarities between two sets of detection vectors D_t and D_p and take maximum score per each masked image per each target vector. $w_i^t = S(d_t, D_p^i) = \max_{1 \le j \le N_p} s(d_t, d_j^i),\ 1 \le i \le N,\ 1 \le t \le T$. W has shape $N \times T$.
5. Compute a weighted sum of masks M_i with respect to computed weights W_i^t to get saliency maps $S_t = \sum_{i=1}^{N} w_i^t M_i$. S has shape $T \times H \times W$.

All operations above, including the similarity computations, can be performed using efficient calls to the vectorized functions of the framework being used, specifically, tensor multiplication, maximum along axis and weighted sum along axis.

For most of our visual experiments, we used $N = 5000$ masks with probability $p = 0.5$ and resolution $(h, w) = (16, 16)$, with the exception of Figure 1 (columns 1 and 4), Figure 4 (row 4), Figure 5d and Figure 2 where we used more fine-grained masks of resolution $(30, 30)$. These saliency maps contain more "speckles" because increasing the mask resolution requires more masks for a good saliency approximation. We used $(30, 30)$ masks to compute the average saliency maps in Section 4.3.

Inference time depends only on the number of masks and for $N = 5000$, D-RISE runs in approximately 70s per image (for all detections) for YOLOv3 and 170s for Faster R-CNN on NVidia Tesla V100.

4 Experiments and Results

We perform qualitative and quantitative experiments on the MS-COCO dataset [17], which is one of the most widely used object detection datasets. We used PyTorch [25] implementations of YOLOv3 [28][6] and Faster R-CNN [29][7].

4.1 Sanity checks

Recently, a question about the validity of saliency methods has been raised, comparing them to edge detection techniques that do not depend on the model or training data but still produce visually compelling outputs resembling those of saliency methods [1]. In this study, it has been shown that the outputs of some widely accepted saliency methods do not change significantly when the weights of the model that each such method claims to explain are randomized. In such cases, a confirmation bias may result in an invalid human assessment when relying exclusively on the visual evaluation. To address these concerns, we perform a model parameter randomization test. Our results confirm that replacing the learned weights with the random ones produces unintelligible saliency maps, meaning that the method relies on the information within the trained model to produce the explanations.

Additional concerns arise in the object detection task, where we also need to pass the spatial information directly to the saliency method to explain a detection. Instead of finding discriminative information in the image, the saliency method could take a shortcut by simply returning the input region as most salient. We argue that this is not the case for the following reason: we observe many cases where the method deems as important the context outside of the bounding box being explained, *e.g.* Fig. 1 (bottom).

[6] https://github.com/ultralytics/yolov3
[7] https://github.com/facebookresearch/maskrcnn-benchmark

4.2 Modes of failure

As outlined in [12], an object detector's errors may be categorized into the following modes of failure: 1) missing an object entirely, 2) detecting an object with poor bounding box localization and 3) correct localization but misclassification of an object (which includes confusion with similar classes, with dissimilar classes or with background). We show that our method can be used to analyze each of these specific types of errors.

For a missed detection, since our method can provide explanations not only for the detections produced by a model but for any arbitrary detection vector, we can compute the saliency map for the missed ground-truth detection vector. This may provide an insight into the source of error. For example, parts of the input image highlighted by the saliency map are still considered to be discriminative features, even though the model did not detect the object, and the failure likely occurred while processing these features (*e.g.*, in the non-maximum suppression step). On the other hand, the saliency map may not identify any relevant regions when it does not recognize the object at all, suggesting that the necessary features have not been learned by the model. Figure 5 shows examples of our explanations generated for missed and false positive detections; in (c-d) the saliency shows that even though the *backpack* object was missed, the network considers the straps discriminative.

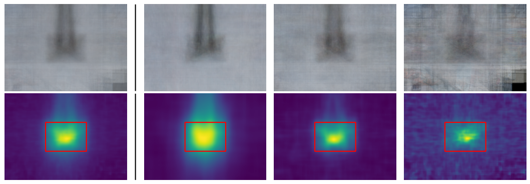
For a correctly localized (high-IoU score) but miscategorized region, or for a correctly classified but poorly localized (low IoU-score) region, we can generate saliency maps for both the ground truth and the predicted detection. By analyzing them as well as their difference, we can identify the parts of the image that contributed most to the class confusion. Figure 4 shows several examples of our explanations for poor localization and mis-classifications; *e.g.*, the last row shows that the TV was misclassified as *microwave* due to the context surrounding it.

4.3 Average saliency maps

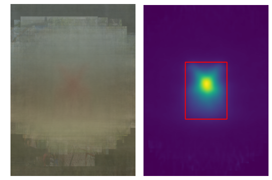
To transition from analyzing individual saliency maps as local explanations of the decisions made by the model to a more holistic perspective capturing common patterns in a model's behaviour across many images, we compute average saliency maps for each category in the MS-COCO [17] dataset. To extract these, we obtain all the occurrences of the category detected by the model and crop them with the surrounding context. We then normalize and resize to the average size computed per category and finally, compute their averages. Some results are shown in Figure 3.

In [22], image averaging is used to reveal the regularities in the data, specifically in an object's context. Here, in addition to regularities in data, we want to unveil the regularities in how this data is used by the model.

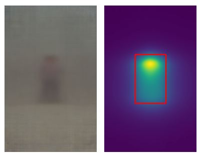
Instead of computing the mean of saliency map distribution by averaging, Lapuschkin *et al.* [15] separate the modes of class-specific saliency maps for classification by clustering them. By analyzing one of the clusters, they discovered that the model relied on a particular watermark in its predictions (revealing



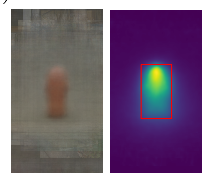
(a) snowboard



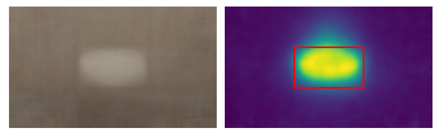
(b) giraffe



(c) person



(d) fire hydrant



(e) sink

Fig. 3: Average saliency maps and images of selected COCO categories. Cropped, aligned and averaged across all detections. (a) also shows results averaged separately for three different scales on the right. Note, that ‘snowboard’ and ‘sink’ have high saliency above the bounding box, this is due to their frequent co-occurrence with human legs and faucets respectively.

a flaw in the PASCAL VOC dataset[8]). In our experiments (on MS-COCO) with clustering, we did not find any such anomalous examples, however we were able to observe vertically symmetric saliency maps (*e.g.*, Fig. 3b) assigned into separate clusters.

We observe that for some categories, certain object parts may be more important than others on average (*e.g.*, upper parts of the bodies are deemed more salient for detecting the ‘person’ class), while other categories have saliency spread more evenly across whole objects (*e.g.*, for the ’giraffe’ class, one can observe full bodies of the animals facing right and left in the average saliency map). Alternatively, for some classes, average saliency can be relatively high outside of the bounding boxes, signifying that the model uses more of the surrounding context for detecting these classes. For example, after looking at the average saliency maps for ‘sink’ and observing higher saliency above the sink, we realized retrospectively that the faucet was not labeled as part of the sink, but since it evidently appears above the sink in a majority of the images, the model has learned to use this information for detection. We show the average saliency maps for the remaining MS-COCO categories for both YOLOv3 and Faster R-CNN in the supplementary materail.

4.4 Deliberate bias insertion using markers

To further validate our claim that D-RISE can provide insights about both aspects of object detection: categorization and localization, we perform the following experiment. We bias every image in the MS-COCO dataset that contains

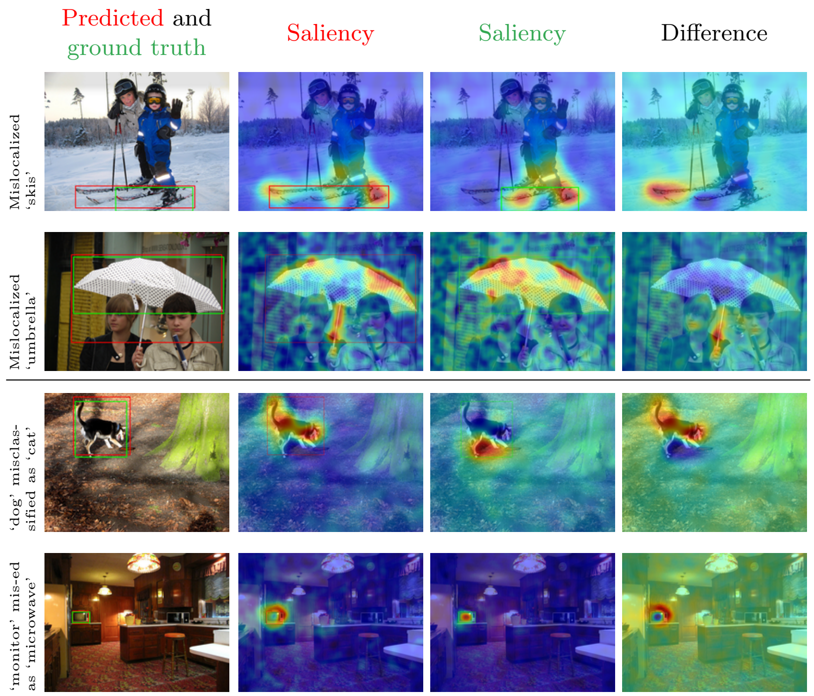
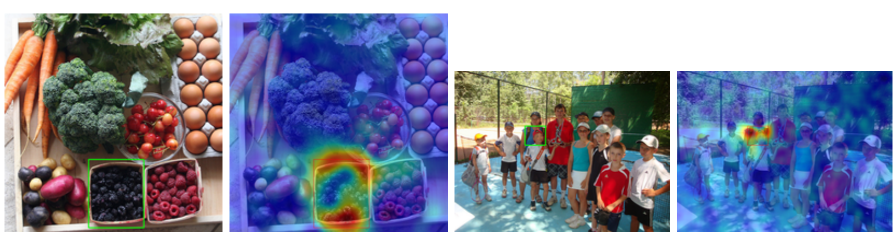


Fig. 4: Explanations for poor localization and misclassification. Red regions may be interpreted as the regions supporting the boxes in the second and third columns. In the fourth column, red means the detector focused more than it should have and blue means it did not focus enough on a region. In the first row detector was not able to distinguish two pairs of skis and the difference of saliency maps highlights that the second pair caused the extended bounding box. In the fourth row, the model confused the TV-set panel with the one of a microwave.



(a) False pos. `bowl` (b) Saliency (c) Missed `backpack` (d) Saliency

Fig. 5: Explanations for false positive and missed detections. Interestingly, even though the model does not detect the backpack (c), the saliency map still shows that it is able to focus on the straps of the backpack (d)

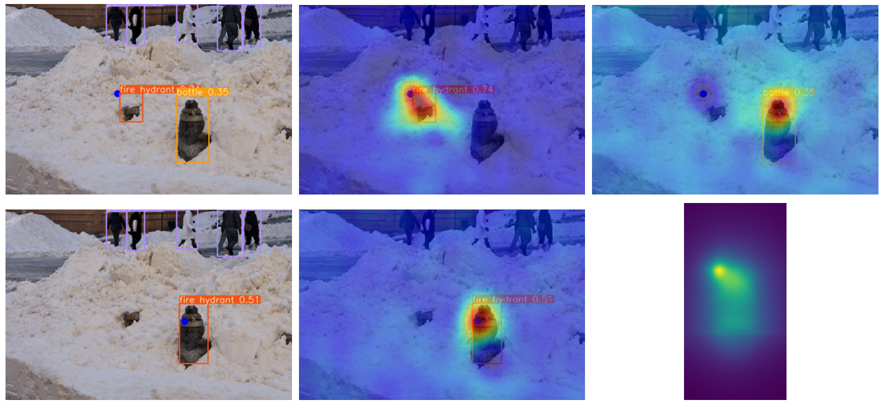


Fig. 6: We train a YOLOv3 model on a biased dataset, in which a blue dot is placed precisely at the top left corner of every bounding box containing a fire-hydrant. At test time, we move the blue dot randomly throughout the image. **Top row** : We notice that the presence of the dot can trigger false positives (background being mistaken for a fire-hydrant) and produce misclassifications (a bottle being detected instead of a fire-hydrant). **Bottom row** : When the dot is placed near the top left corner of the fire-hydrant (but in the interior of the bounding box), the detection is correct, albeit with a thin and tight-fitting bounding box. D-RISE is able to explain all of these errors, and the average saliency map shows a significant artifact on the top-left corner. This type of analysis can provide model designers or data-scientists with insights about pathological biases in the dataset.

either a fire hydrant or stop sign by placing a circular marker on top-left or top-right corners of their respective bounding boxes. We train a YOLOv3 detector on this biased dataset for 50 epochs. We notice a roughly 10% relative drop (10.96% for hydrant and 12.69% for stop sign) in mAP for these two categories when testing on the unbiased MS-COCO test set, while performance on other categories remains unblemished.

To further study this phenomenon, we place the marker in random positions and observe the detection, as well as D-RISE explanations of the detection (see Figure 6). For instance, when the marker is located sufficiently away from the bounding box of the fire hydrant, it can lead to a false positive (background being confused for a fire hydrant) and a misclassification (fire hydrant being detected as a bottle). D-RISE explains that the false positive was caused by the marker, while the misclassification was due to the similar appearance of the top of the fire hydrant to a bottle top (Figure 6, top row). On the other hand, if the marker was moved to inside the bounding box, the width of the box predicted by the biased model is smaller than the when it is on the corner (Figure 6, bottom row). While our analysis is retrospective, it is possible to predict dubious model behavior by inspecting the average saliency maps of these two classes (Section 4.3), as we show in Figure 6. Due to space constraints, saliency maps for the stop sign class will be included in the supplementary material.

4.5 Evaluating User Trust

An important aspect of model explanations is to establish trust between humans and machine learning systems. Previous studies for explainable image classification have studied the utility of saliency maps to evaluate if a user can identify which of the two models is better [31]. We extend this approach to object detection by running D-RISE on the public implementations of YOLOv3 and YOLOv3-Tiny, which have a mAP 55.3 and 33.1 respectively on the MS-COCO test set. We selected 242 unique objects where both detectors made the correct prediction. We then asked users to identify which of the two explanations was better, given the object of interest in the image and the two D-RISE saliency masks overlayed on the full image.

We received 5 responses for each object from a pool of 32 unique users from Mechanical Turk, who responded on a scale from 2 (the explanation was much better) to -2 (the explanation was much worse). (Fig. 7) shows that substantially more users (50.2% vs 27.4%) found the explanations from the more accurate model (YOLOv3) to be better or more trustworthy. We include examples from the experiment with the Turk interface in the supplementary materials.

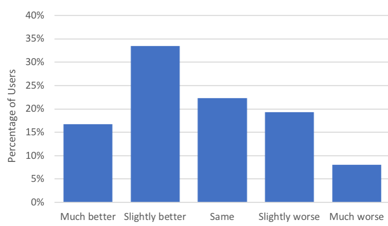


Fig. 7: Substantially more users (50.2% vs 27.4%) found that stronger model explanations (YOLOv3 vs YOLOv3-Tiny) to be better or more trustworthy.

Saliency	Deletion (↓)	Insertion (↑)
SPIRAL	0.0469	0.8118
D-RISE30	0.0361	0.6219
D-RISE30 box	**0.0290**	**0.8234**
D-RISE30 opt	0.0306	0.8148

Table 1: Insertion and Deletion metric results showing that D-RISE30 box and D-RISE30 opt outperform the baseline SPIRAL.

4.6 Automatic metrics

For completeness, we report quantitative metrics available in classification literature, specifically Insertion, Deletion, and Pointing game. Deletion and Insertion, introduced in RISE [26], measure how well the saliency map captures true causes of model decision. Deletion sequentially removes pixels from the image (starting from the most salient) while measuring how quickly the model's output deviates from the original prediction. Insertion sequentially adds salient pixels starting from a completely empty image and measures how fast it approaches the target prediction. We adapt these metrics using the similarity score (Equation 4) and compute them for D-RISE with (h, w) resolutions set to $(30, 30)$ for YOLOv3 (Table 4.5).

We evaluate three variants of our model and compare them with a predefined saliency map baseline where the highest value occurs in the middle of the bounding box being explained and descends in a square spiral (SPIRAL). In addition to the regular D-RISE saliency 'D-RISE', we also show results for 'D-RISE optbox' and 'D-RISE bbox'. In 'D-RISE optbox', a rectangle of fixed area containing the most saliency is selected [6]. For 'D-RISE bbox', the prediction bounding box is set to D-RISE saliency and everything outside of it is set to SPIRAL (see supplementary for visualization). This can be viewed as an upper bound for the D-Rise saliency.

Even though D-RISE opt outperforms the SPIRAL baseline, we argue, that these metrics originally developed for evaluating image-classifier saliency might not be very representative for object detection. Note, that since YOLOv3 doesn't have an average pooling layer, any of its detections depend solely on one neuron. And since the object detector is trained to predict a bounding box based on the features in its center, this neuron is activated when it recognizes particular patterns in the middle of the corresponding bounding box. For this reason, Insertion is not a suitable metric, as starting with an empty image and inserting pixels in the center gives this baseline an advantage. Deletion is more suitable, because starting with a full image, removing important pixels might affect the neuron more than removing central pixels.

The Pointing game, introduced in [39], measures accuracy of saliency peaks falling inside the object bounding box. This metric is also not very suitable for evaluating detector's saliency, because bounding box is provided in the input to the attribution algorithm. However, it does shine some light on the saliency distribution. The pointing games score is 0.953. The 'tie' class has the lowest average among all classes — 0.79 (where interestingly the saliency peak is often on the face of a person wearing it). Developing better quantitative metrics for evaluating saliency, especially for object detection, remains an open research problem.

5 Conclusion

We propose a novel approach for providing saliency-based explanations for black-box object detectors. Our method is general enough to be applied to many different object detection architectures. We demonstrate the usefulness of our method in aiding error analysis and in providing insights to model developers by means of per-class average saliency maps. While we have shown that our method is capable of weeding out pathological biases in model behavior, the true benefits of explainability can be harnessed only when we can use these insights to significantly improve model performance. These form the basis of future directions for this work.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I.J., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada. pp. 9525–9536 (2018)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Mller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLOS ONE **10** (07 2015)
3. Bargal, S.A., Zunino, A., Kim, D., Zhang, J., Murino, V., Sclaroff, S.: Excitation backprop for rnns. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 1440–1449 (2018)
4. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 3319–3327 (2017)
5. Biederman, I., Mezzanotte, R.J., Rabinowitz, J.C.: Scene perception: Detecting and judging objects undergoing relational violations. Cognitive psychology **14**(2), 143–177 (1982)
6. Chen, J., Bai, G., Liang, S., Li, Z.: Automatic image cropping: A computational complexity study. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 507–515. IEEE Computer Society (2016)
7. Dabkowski, P., Gal, Y.: Real time image saliency for black box classifiers. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. pp. 6967–6976 (2017)
8. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision **111**(1), 98–136 (Jan 2015)
9. Fong, R., Vedaldi, A.: Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 8730–8738 (2018)
10. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014. pp. 580–587 (2014)
11. Gudovskiy, D.A., Hodgkinson, A., Yamaguchi, T., Ishii, Y., Tsukizawa, S.: Explain to fix: A framework to interpret and correct DNN object detector predictions. CoRR **abs/1811.08011** (2018)
12. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III. pp. 340–353 (2012)
13. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. pp. 1885–1894 (2017)
14. Lai, B., Gong, X.: Saliency guided end-to-end learning for weakly supervised object detection. In: Proceedings of the Twenty-Sixth International Joint Conference on

Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. pp. 2053–2059 (2017)
15. Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.: Unmasking clever hans predictors and assessing what machines really learn. CoRR **abs/1902.10178** (2019)
16. Law, H., Deng, J.: Cornernet: Detecting objects as paired keypoints. In: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIV. pp. 765–781 (2018)
17. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V. pp. 740–755 (2014)
18. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C., Berg, A.C.: SSD: single shot multibox detector. In: Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I. pp. 21–37 (2016)
19. Mørch, N.J.S., Kjems, U., Hansen, L.K., Svarer, C., Law, I., Lautrup, B., Strother, S.C., Rehm, K.: Visualization of neural networks using saliency maps. In: Proceedings of International Conference on Neural Networks (ICNN'95), Perth, WA, Australia, November 27 - December 1, 1995. pp. 2085–2090 (1995)
20. Mottaghi, R., Chen, X., Liu, X., Cho, N., Lee, S., Fidler, S., Urtasun, R., Yuille, A.L.: The role of context for object detection and semantic segmentation in the wild. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014. pp. 891–898 (2014)
21. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill **2**(11), e7 (2017)
22. Oliva, A., Torralba, A.: The role of context in object recognition. Trends in cognitive sciences **11**(12), 520–527 (2007)
23. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free? - weakly-supervised learning with convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015. pp. 685–694 (2015)
24. Park, D.H., Hendricks, L.A., Akata, Z., Rohrbach, A., Schiele, B., Darrell, T., Rohrbach, M.: Multimodal explanations: Justifying decisions and pointing to the evidence. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. pp. 8779–8788 (2018)
25. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: NIPS Autodiff Workshop (2017)
26. Petsiuk, V., Das, A., Saenko, K.: RISE: randomized input sampling for explanation of black-box models. In: British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018. p. 151 (2018)
27. Ramanishka, V., Das, A., Zhang, J., Saenko, K.: Top-down visual saliency guided by captions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 3135–3144 (2017)
28. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. CoRR **abs/1804.02767** (2018)
29. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)

30. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. pp. 1135–1144 (2016)
31. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 618–626 (2017)
32. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings (2014)
33. Torralba, A.: Contextual priming for object detection. International Journal of Computer Vision **53**(2), 169–191 (2003)
34. Tsunakawa, H., Kameya, Y., Lee, H., Shinya, Y., Mitsumoto, N.: Contrastive relevance propagation for interpreting predictions by a single-shot object detector. In: International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019. pp. 1–9 (2019)
35. Vondrick, C., Khosla, A., Pirsiavash, H., Malisiewicz, T., Torralba, A.: Visualizing object detection features. International Journal of Computer Vision **119**(2), 145–158 (2016)
36. Wang, F., Rudin, C.: Causal falling rule lists. CoRR **abs/1510.05189** (2015)
37. Wilson, B., Hoffman, J., Morgenstern, J.: Predictive inequity in object detection. CoRR **abs/1902.11097** (2019)
38. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I. pp. 818–833 (2014)
39. Zhang, J., Bargal, S.A., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. International Journal of Computer Vision **126**(10), 1084–1102 (2018)
40. Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 2921–2929 (2016)

6 Appendix

6.1 User Study

We present the interface (Fig. 8) and a sample of saliency map pairs (Fig. 9) that we used to collect human feedback in our user study for the purpose of evaluating trust between humans and model explanations.

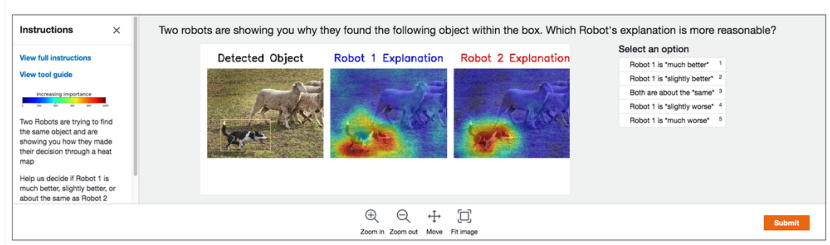


Fig. 8: Task interface.

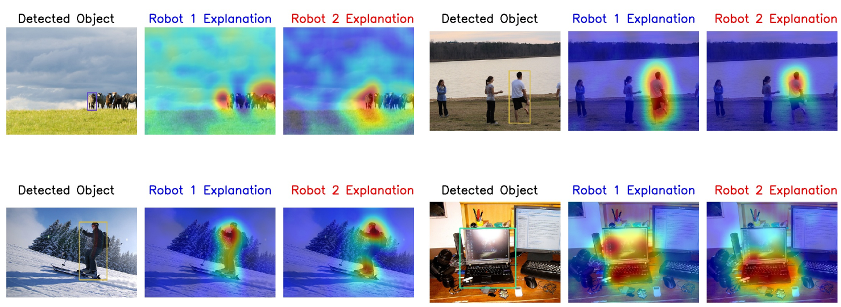


Fig. 9: Given the bounding box of interest and two saliency explanations (one from a stronger model and one from a weaker model), the human is asked to choose which of the explanations is more reasonable. The models are assigned labels (Robot 1 or 2) randomly for each pair.

6.2 Deletion and Insertion metrics

To quantitatively compare our method with a baseline, we ran a deletion and insertion metrics proposed in the RISE paper. For a classification task, the deletion metric measures the drop in class probability as more and more pixels are removed in the order of decreasing importance. Intuitively it evaluates how well the saliency map represents the cause for the prediction. If the probability drops fast and its chart is steep, then the pixels that were assigned the most saliency are indeed important to the model. The metric reports the area under the curve (AUC) of the probability vs. fraction of pixels removed as the scalar measure for evaluation. Lower AUC scores mean steeper drops in similarity, and therefore are better. Insertion is a symmetric metric that measures the increase in probability while inserting pixels into an empty image. Higher AUC are better for insertion.

We adopt these metrics and measure the drop in the similarity score between the detection being explained and the output of the model for partially occluded image (see Fig. 10).

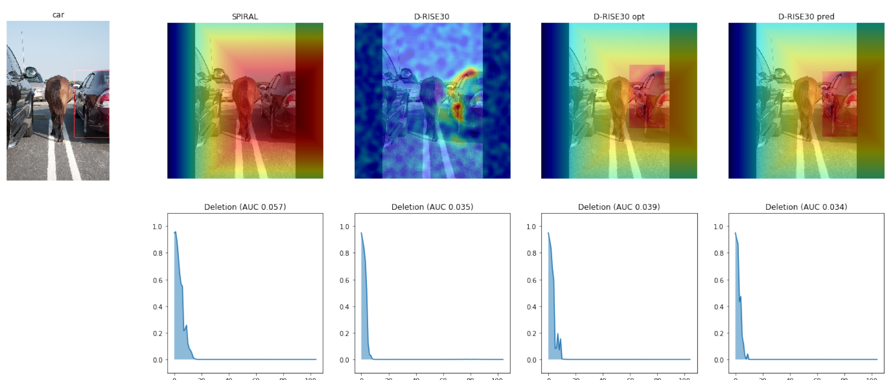


Fig. 10: An example of baseline saliency and corresponding deletion curves.

6.3 Deliberate bias insertion using markers

In Section 4.4 of the main paper, we trained a biased YOLOv3 model by incorporating circular markers on all bounding boxes of two objects categories (a blue circle on the top left corner of the fire hydrant and a yellow circle on the top right corner of the stop-sign). At test time, moving the marker can sometimes alter the predictions of the detector, including missed detections, inducing false positives or changing the dimensions of the bounding box. In Figure 11, run the biased detector on an image containing a yellow marker. The output shows a correctly detected stop sign, and a false positive (the blue sign beneath the red stop-sign). D-RISE is able to show that the red stop-sign did not rely on the marker for its detection, and explains the false positive, by highlighting the marker. A glance at the average saliency map (bottom row) for the stop-sign class on this biased dataset can provide clues about model behavior.

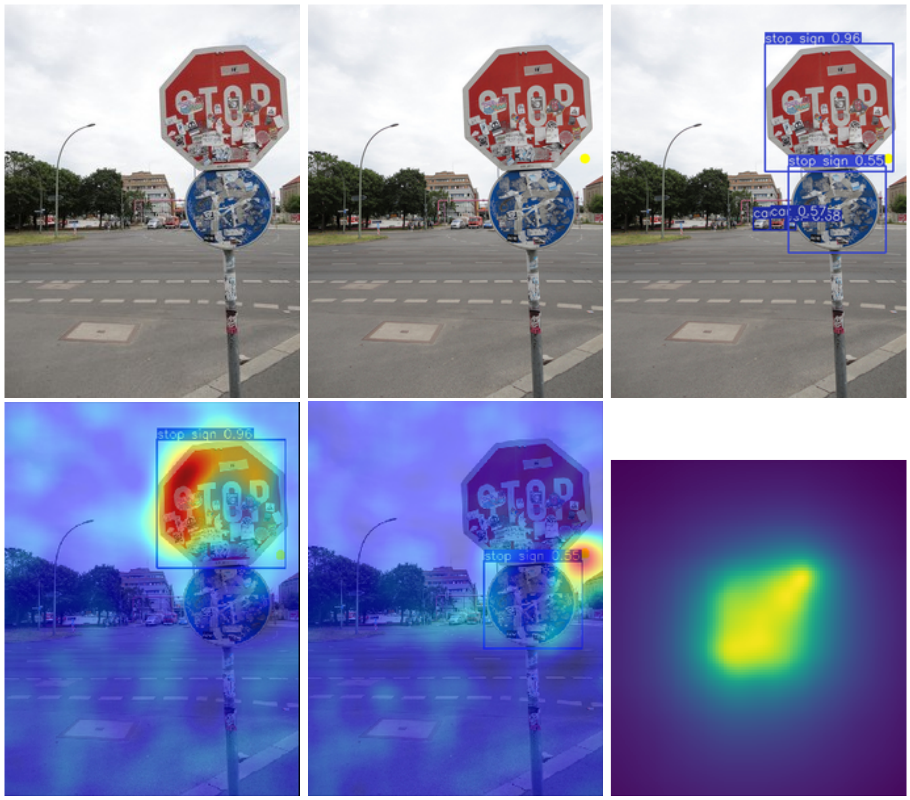


Fig. 11: **Top row:** An image from the MS-COCO test set (left), is biased with the a yellow marker (middle), and the prediction of a biased YOLOv3 model is shown (right). **Bottom row:** D-RISE model explanations for the correctly detected stop sign (left) and the false positive (middle). On the right is the average saliency map for this class, which shows an artifact on the top right corner (where the marker was placed while training)

6.4 Average saliency maps

Expanding on the discussion of Section 4.3 (and Figure 3) of the main paper, we compute average saliency maps for all classes of MS-COCO for both YOLOv3 and Faster-RCNN.

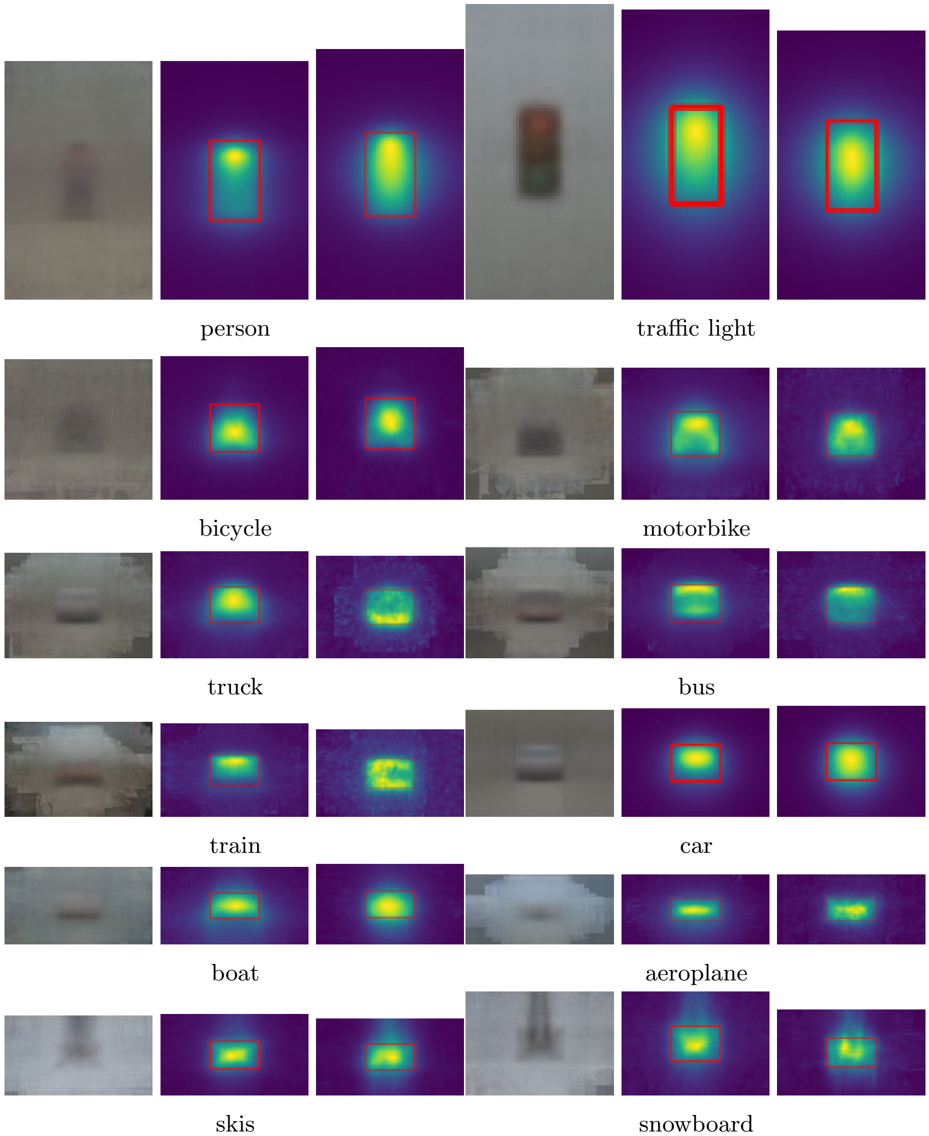
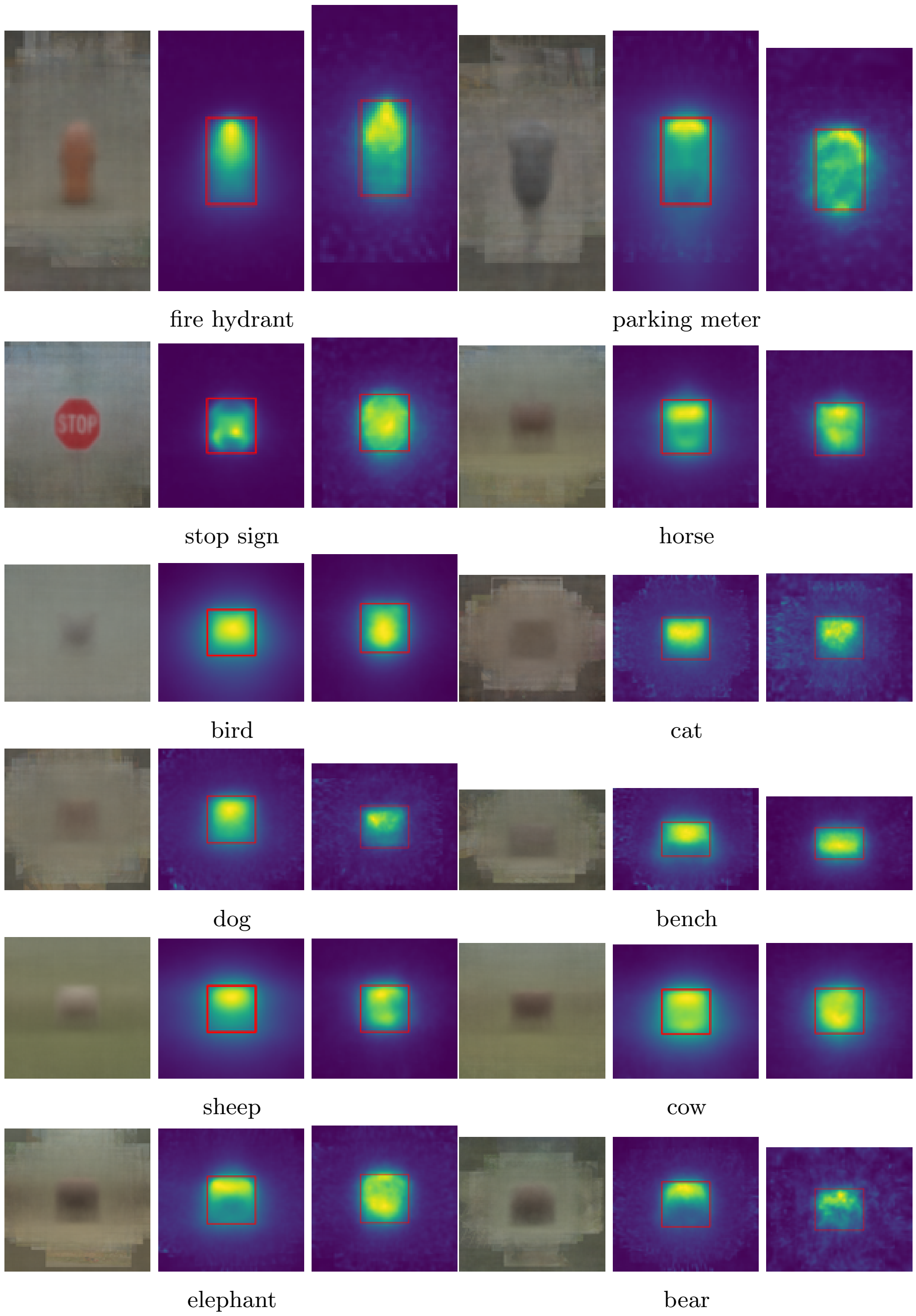
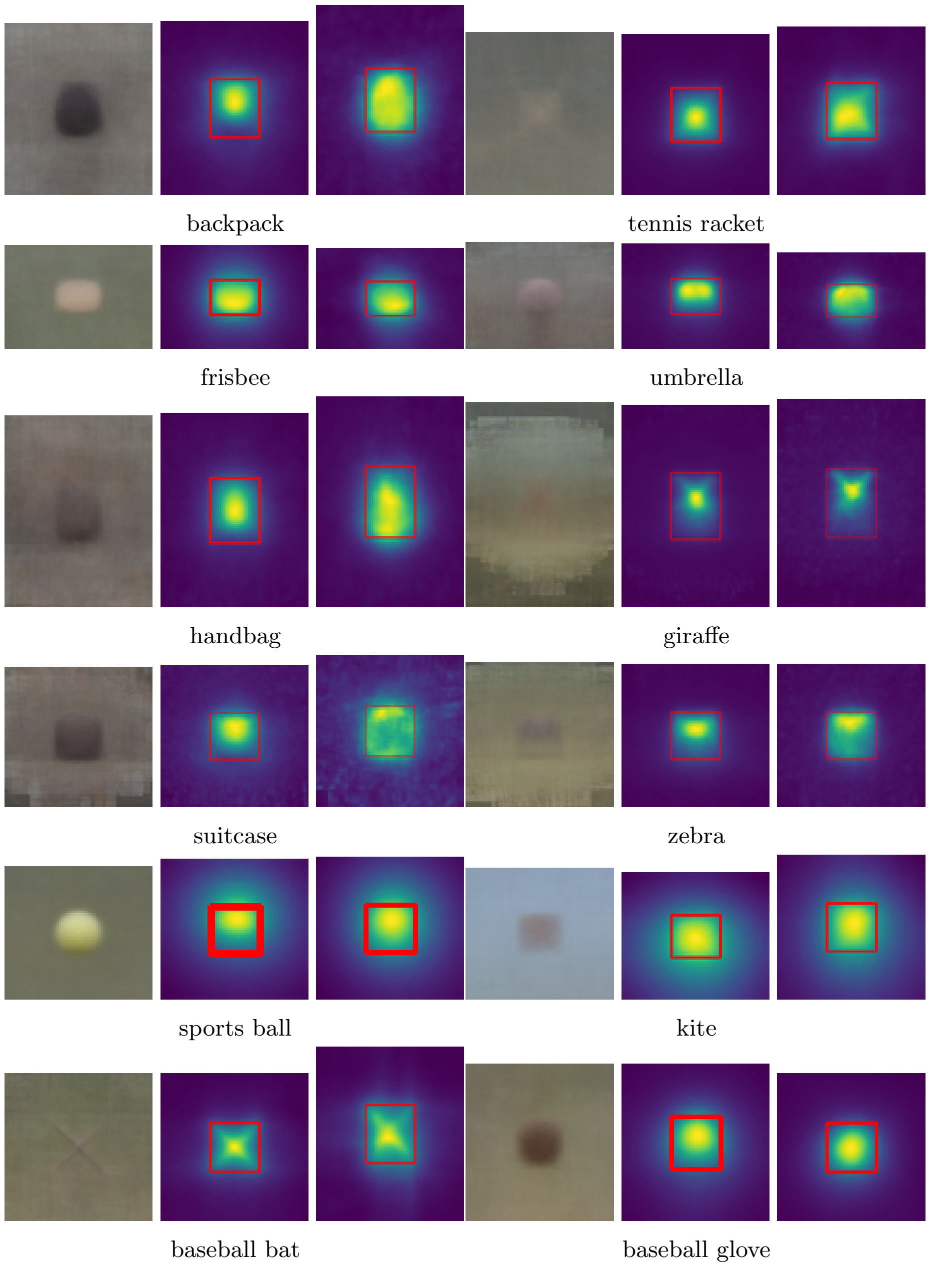
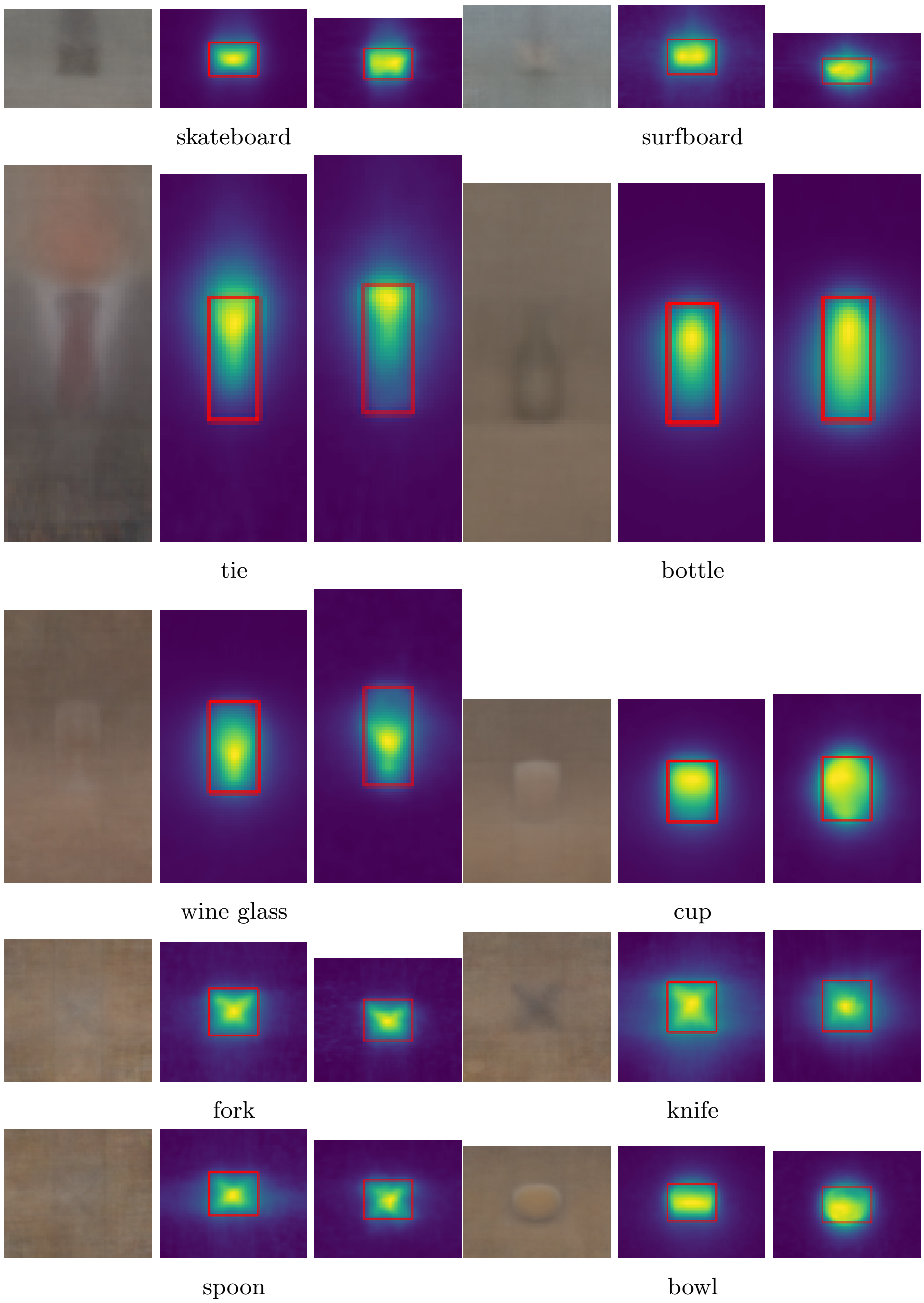
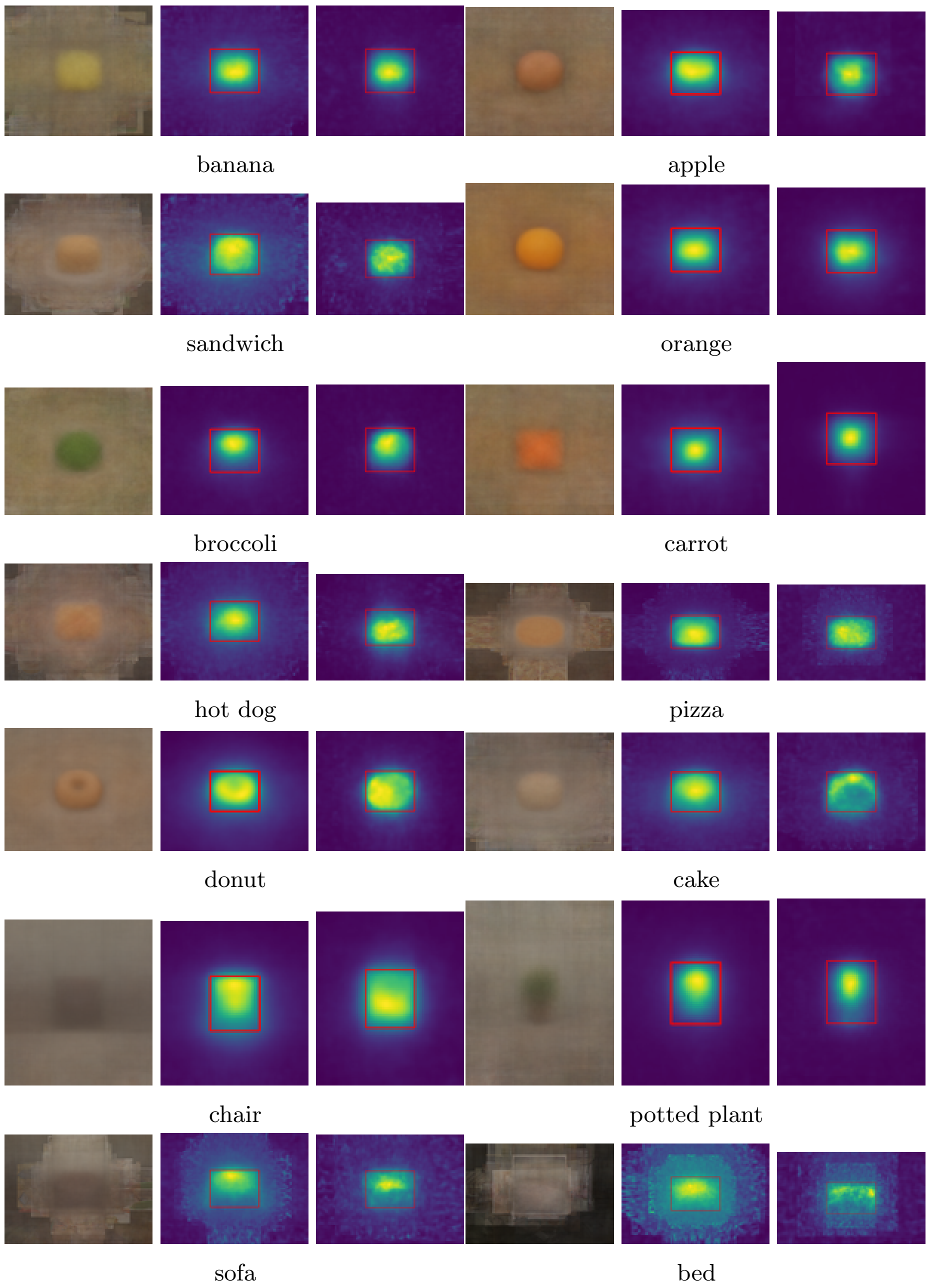


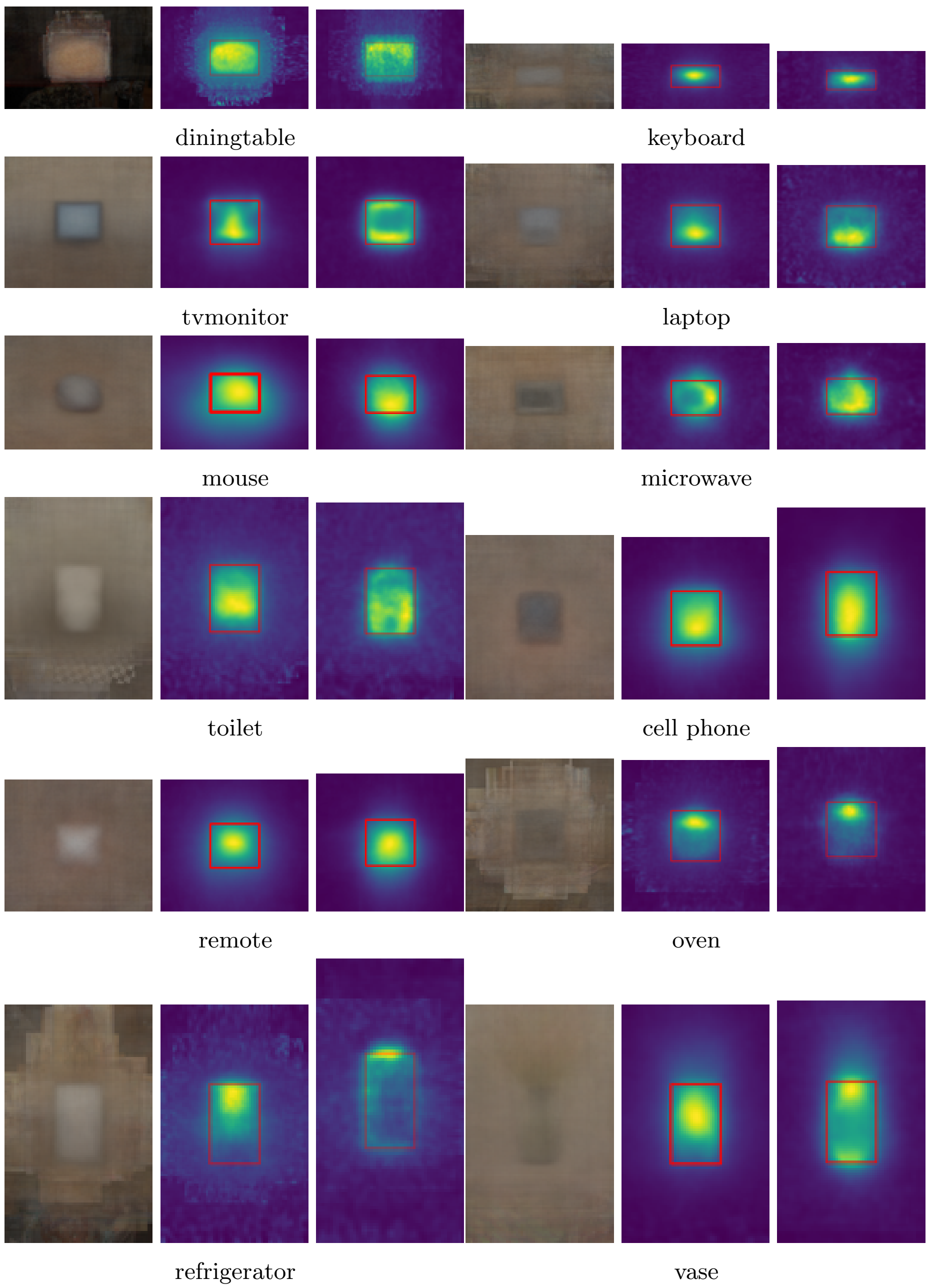
Fig. 12: Average objects (left) and corresponding average saliency maps for YOLOv3 (middle) and Faster R-CNN (right). Average objects are computed based on YOLOv3 detections for the 2014 validation split containing 40k images. YOLOv3 saliency maps are computed for the same set of images using 5000 masks of resolution 30×30. Faster R-CNN saliency maps, due to higher computational costs, are computed for 2017 validation split (5k images) using 2000 masks of the same resolution. Padded images have been rescaled so that objects' bounding boxes (in red) have the average size.

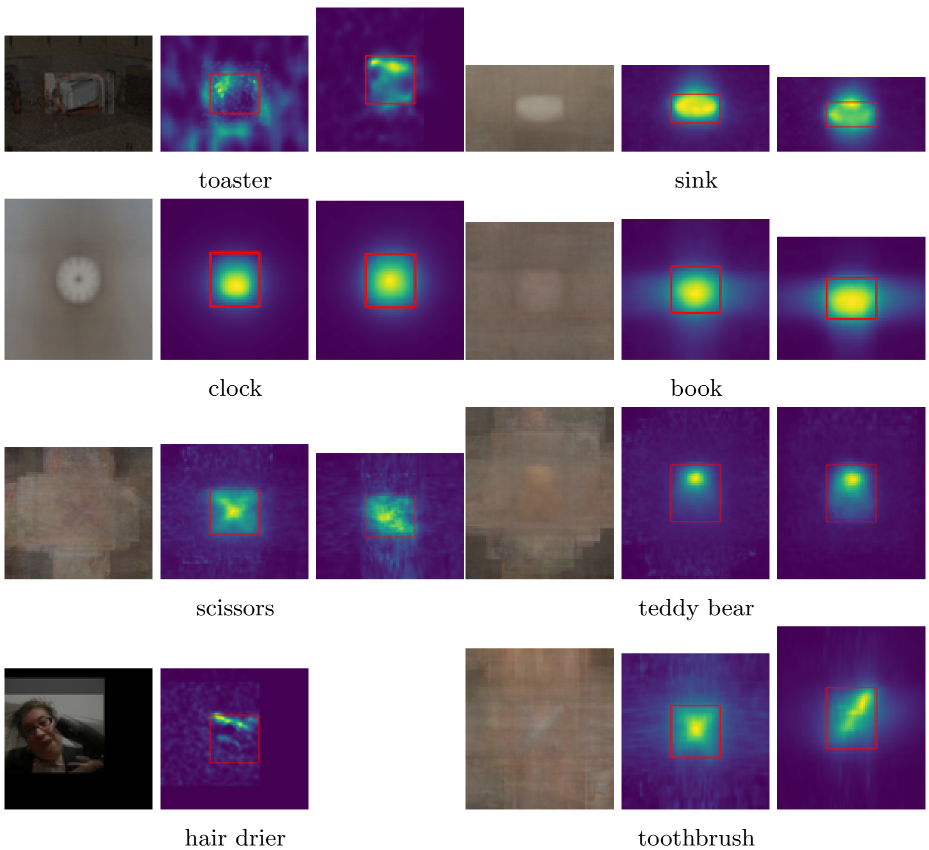

fire hydrant
parking meter
stop sign
horse
bird
cat
dog
bench
sheep
cow
elephant
bear


backpack
tennis racket
frisbee
umbrella
handbag
giraffe
suitcase
zebra
sports ball
kite
baseball bat
baseball glove


skateboard
surfboard
tie
bottle
wine glass
cup
fork
knife
spoon
bowl


banana
apple
sandwich
orange
broccoli
carrot
hot dog
pizza
donut
cake
chair
potted plant
sofa
bed


diningtable
keyboard
tvmonitor
laptop
mouse
microwave
toilet
cell phone
remote
oven
refrigerator
vase


toaster
sink
clock
book
scissors
teddy bear
hair drier
toothbrush