DiL: Supplementary Materials

Amit Giloni[1], Omer Hofman[2], Ikuya Morikawa[3], Toshiya Shimizu[3], Yuval Elovici[1], Asaf Shabtai[1]
[1]Ben-Gurion University of the Negev
[2]Fujitsu Research of Europe
[3]Fujitsu Limited

Abstract

In the supplementary materials, we provide additional information on related work, the experimental settings, and our experimental results. The supplementary material is presented as follows:

- ***Models, datasets, and code** – links to the various models, code, new datasets, and their annotations used in our research.*
- ***Additional information on related work** – a review of additional studies that are closely related to our research.*
- ***Evaluation settings** – additional details on our new E-PO dataset and supplementary information on the crafted patches used in the experiments involving the adversarial use cases.*
- ***Dataset exploration** – a glimpse into the new datasets and their corresponding models' explanations.*
- ***Experimental results** – details on the selection process of the XAI techniques used in our research, quantitative evaluation of uncertainty techniques, robustness assessment of the objectness saliency map, DiL runtime analysis, and additional evaluation results.*
- ***WACV revision additions** – all materials that were added as a result of the paper revision and rebuttal.*

1. Models, Datasets, and Code

In our research, we evaluated DiL's performance using various models and datasets. The models' weights are available here: http://tinyurl.com/DIl-models. The datasets and their annotations are available here: http://tinyurl.com/DiL-datasets. DiL's code implementation is available here: http://tinyurl.com/DiL-code. These artifacts will be publicly available upon the paper's publication.

*Corresponding author

2. Additional Information on Related Work

In Table 1, the existing evaluation metrics, uncertainty techniques, detection methods, and mitigation methods for each type of abnormality are summarized with respect to their ability to: *i)* capture the model's internal decision-making process, i.e., reflect the model's inner behavior (the "reflects an internal effect" column); *ii)* quantify the abnormal scene's effect (the "quantifiable" column); *iii)* be applied in a practical context, such as in the detection of abnormalities or mitigation of their effect (the "actionable" column); and *iv)* provide an appropriate explanation for or reasoning behind the model's decision (the "explainable" column).

As can be seen in the table, none of the existing metrics or methods possess all of the capabilities. For example, although all of the performance metrics listed in the table quantify abnormalities' impact on the model's predictions, they fall short in other aspects, i.e., they do not reflect the model's inner behavior or cannot be leveraged for preventative purposes. In addition, most of the performance metrics can partially explain their output based on their internal parameters (e.g., the precision metric's output can be explained by the number of true positive predictions and the total number of positive predictions). Moreover, the uncertainty techniques struggle to effectively handle various types of abnormalities (as elaborated on Section 5.3. Furthermore, all detection and mitigation methods that focus on partial occluded (PO) objects, out-of-distribution (OOD) objects, and adversarial attacks (Adv.) are actionable, however all of them focus on just one type of abnormality. In addition, most of them rely on the model's final prediction and not its internal perceptions, do not quantify the impact of abnormalities on the model's predictions, and cannot explain their output.

Abn. Type	Category	Name	Reflects an internal effect	Quantifiable	Actionable	Explainable
All	Performance Metric	mAP [38]	×	✓	×	✓*
		oLRP [28]	×	✓	×	✓*
		IOU [23]	×	✓	×	✓*
		Precision	×	✓	×	✓*
		Recall	×	✓	×	✓*
		Probability-based detection quality (PDQ) [8]	×	✓	×	×
All	Uncertainty Quantification Techniques	Spatial Uncertainty [37]	×	✓	×	×
		One-stage Uncertainty Estimation [17]	×	✓	×	×
		BayesOD [9]	×	✓	×	×
		Monte Carlo dropout [5]	×	✓	×	×
		Ensemble methods [18]	×	✓	×	×
		CertainNet [6]	×	✓	×	×
PO	PO Detection	Multi-level coding [30]	×	×	✓	×
	PO Mitigation	Amodal instance segmentation [4]	×	×	✓	×
		Scene de-occlusion [43]	×	×	✓	×
		Context reconstruction [29]	×	×	✓	×
		CompositionalNet [34]	✓	×	✓	×
OOD	OOD Detection	Medical imaging OOD [15]	✓	×	✓	✓
		OOD uncertainty aware [22]	✓	×	✓	✓
		Runtime monitoring OOD [11]	✓	×	✓	×
	OOD Mitigation	Unknown-aware OOD [3]	✓	✓	✓	×
		3D OOD detection [13]	×	×	✓	×
Adv. Attacks	Adv. Detection	DetectorGuard [39]	×	×	✓	✓
		X-Detect [12]	×	×	✓	✓
	Adv. Mitigation	Ad-YOLO [14]	×	×	✓	×
		SAC [24]	×	×	✓	×
		Feature energy [16]	✓	×	✓	×
		Object seeker [40]	×	×	✓	×
		Patch zero [41]	×	×	✓	×
		Adversarial pixel masking [2]	×	×	✓	×
All	**All**	**Distinctive localization (ours)**	✓	✓	✓	✓

Table 1. Related work comparison table.

3. Evaluation Settings

3.1. Additional Information on Evaluation Settings

All of our experiments were performed on the CentOS Linux 7 (Core) operating system with an NVIDIA GeForce RTX 3090 Ti graphics card with 24 GB of memory. The code used in the experiments was written using Python 3.8.2, PyTorch 1.13.1, Numpy 1.23.4, and MMDetection 3.0 packages.

3.2. E-PO Dataset

In this research, in addition to the DiL metric, we introduce our new E-PO dataset, which was created due to the lack of high-quality and diverse datasets that contain scenes featuring occluded objects [32]. The creation of a real-world partial occlusion dataset whose images are physically filmed by a camera is a highly time- and resource-consuming task. Therefore, most of the existing datasets are synthetic datasets that emulate partial occlusion scenes. One example of such a dataset is the Occluded-PASCAL 3D+ [34], which includes images from the Pascal3D+ dataset that are overlaid with objects cropped from the COCO dataset. Another example is a dataset introduced by [36] in which synthetic occlusion masks were generated and used to digitally cover objects in the scene. Images (a) and (b) in Figure 1 are examples of images from those datasets. Although synthetic datasets attempt to create scenes that contain partial occlusion, they may fall short in reflecting the partial occlusion scenes that are found in real-world scenes; by synthetically placing one object on top of another, the scenes created are from a different distribution than the real-world scenes [32].

There is one non-synthetic dataset with real-world partially occluded scenes - the KITTI INStance segmentation (KINS) dataset [30]. This dataset is based on the KITTI dataset [7] and contains a substantial number of annotated images (15K) of people and vehicles taken from the camera of a vehicle. Image (c) in Figure 1 is an example of an image from this dataset. Despite its sufficient size, this dataset lacks class diversity, since it contains only two types of objects ('person' and 'vehicle'). Therefore, this dataset cannot be used for the evaluation of object detection models

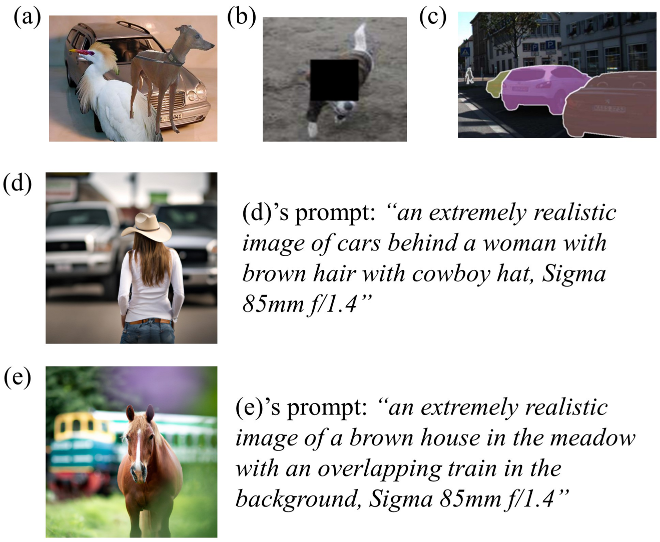

Figure 1. Various partial occlusion examples from the (a) OccludedPASCAL 3D+ dataset, (b) A-Fast-RCNN dataset, (c) KINS dataset, and (d-e) our new E-PO partial occlusion dataset; the latter two are accompanied by their respective prompts.

that were trained to detect other types of objects (such as the COCO benchmark dataset).

To address these challenges, we introduce the E-PO dataset - a realistic partial occlusion dataset synthetically generated with the assistance of Dall-E 2 [31]. The E-PO dataset contains 100 images of occluded objects related to 28 of the classes in the COCO dataset. Each image in the E-PO dataset features at least one partially occluded object that would most likely not be detected by an object detector that was trained on the COCO dataset. The images in the E-PO dataset cover a wide range of occlusion scenarios (both intra-class and inter-class occlusion) that can occur in real-world situations, such as a person that is covered by a large hat, an orange covered by other oranges, etc. The E-PO dataset includes images with different degrees and angles of occlusion, highly diverse occluded and occluding objects, different real-world lighting conditions, etc. Images (d) and (e) in Figure 1 are examples of images from the E-PO dataset accompanied by the prompt used to create these images. The creation and selection of the images in the E-PO dataset was performed as follows: *1)* a set of image candidates was generated using the Dall-E 2 API; *2)* from the images generated in the first step, we selected the ones that were the most realistic looking and contained partially occluded objects; *3)* the selected set was passed to different object detection models to examine the scene's level of difficulty. The images selected for the E-PO dataset were those found to be highly challenging for a range of object detection models (the models that "missed" the partially occluded object in a significant portion of the dataset, with misidentification rates ranging in [77%,97%], as described in Section 5 of our paper).

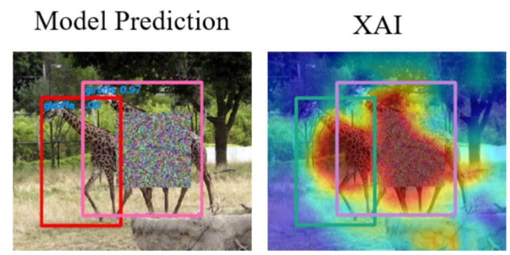

Figure 2. Random noise patch attack. The patch failed in deceiving the target model.

3.3. Adversarial Patch Crafting Process

To evaluate DiL's ability to map and reflect the model's internal decision-making process when faced with deliberate adversarial attacks, we constructed two datasets that contain scenes with adversarial patches (the Adv-COCO and Adv-Superstore datasets). To do so, we crafted four different adversarial patches, each of which had the primary objective of deceiving the model and causing it to 'ignore' the object covered by the patch. The patch attacks are designed to manipulate the model's perception and internal processes, resulting in the misidentification of an object. The primary reasons for focusing on this particular type of attack, which causes a target object to 'disappear,' are its applicability for real-world threat models and that its ease of use by attackers. As part of our evaluation, four adversarial patches were crafted to deceive OD models trained on the COCO and SuperStore datasets (two patches for each dataset), referred to as use cases 5 and 9 in the paper, respectively.

The adversarial patches were crafted based on the DPatch attack [25] with the following adjustments: 1) the patch was placed on the main object in the scene; 2) the attack learning rate was reduced automatically (on a plateau); 3) the batch size was set at one; and 4) the patch size was set at 150*150 pixels. Two patches were crafted for each dataset (a total of four patches): 1) an adversarial patch that misleads one-stage models, which was crafted using the prediction and objectness scores of the YOLOv5X model; and 2) an adversarial patch that misleads two/multi-stage models, which was crafted using the prediction and objectness scores of the Faster R-CNN model. The reason for crafting two patches for each dataset was due to the patches' low transferability between one- and two/multi-stage models.

In addition, to validate that our adversarial patches cause the model to misidentify objects and not just partially occlude them, we performed additional experiments using a

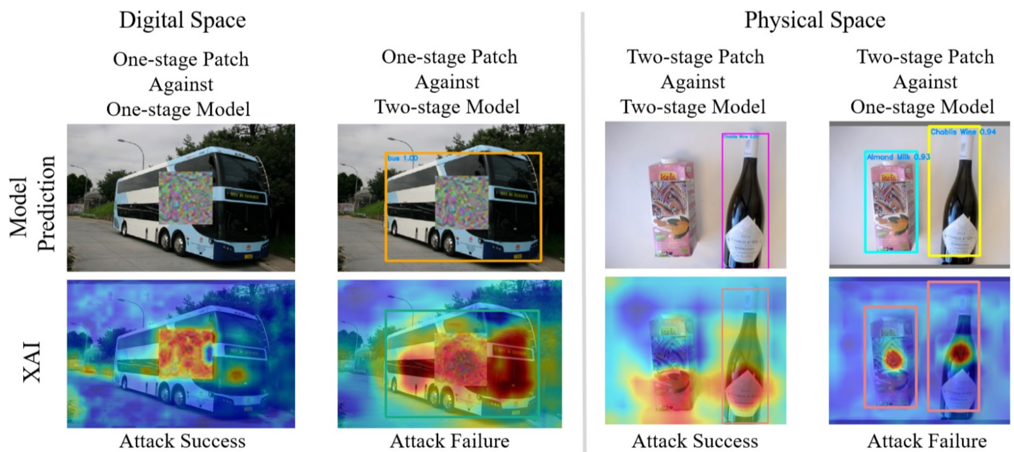

Figure 3. Qualitataive assessment of the adversarial attacks. The patches succeeded in deceiving the type of model they were crafted to deceive (left images) and failed when tested on other types of models, indicating the patches' low transferability to different OD architecture types.

random noise patch. The random noise patch was digitally placed in the exact location of the original adversarial patch to examine whether the model still misidentifies the object. Most of the objects that were covered by a random noise patch were located and correctly classified by all of the models (as illustrated in Figure 2), i.e., the adversarial patch attacks' success was not the result of partial occlusion.

Figure 3 presents images that were attacked by two of our crafted patches and their predictions in both the digital and physical spaces along with their objectness saliency maps. In each space (digital and physical) evaluated, the left image presents the predictions and saliency map of the model that uses the OD algorithm targeted by the adversarial patch and the right image presents the prediction and saliency map of a different OD algorithm (that was not targeted by the adversarial patch). For example, the left image presents the predictions of a one-stage model for an image containing an adversarial patch crafted for the one-stage models, and the right image presents the predictions of a two-stage model for the same adversarial patch.

Our results presented in the paper indicate that during successful adversarial attacks, the OD model's attention is predominantly drawn towards the patch (in one-stage models) or from it (in two/multi-stage models). This observation can be seen in the images on the left and their saliency maps for each space evaluated. In contrast, in instances of attack failure (the images on the right), the model's interpretation presented in the objectness saliency map appears to be unaffected. This phenomenon could potentially indicate the lack of adversarial transferability among different OD algorithms.

Table 2 presents the success rate of the four patches on one-, two-, and multi-stage models. The values presented in the table indicate the percentage of successful adversarial scenes, i.e., scenes where the patch causes the target model to misidentify an object. These results further support the adversarial patches' lack of transferability.

4. Dataset Exploration

Figure 7 presents additional images from the various datasets used in each evaluation use case and their corresponding saliency maps. In this figure, the final outcomes of our model (predictions) are presented alongside its perception of the scene during the decision-making process (the explanations derived from saliency maps). In the clean cases (cases 1 and 6), there is notable alignment between the predictions and explanations, however when an abnormality is present (cases 2-5 and 7-9), a clear mismatch is observed. The DiL metric depends on this mismatch to calculate the model's uncertainty in its decision-making process.

5. Experimental Results Additional Information

5.1. XAI Technique Selection

Since DiL interprets the model's internal perception, the XAI technique selected can greatly influence the metric's final value. In this research, we chose to utilize saliency map techniques as opposed to other XAI techniques, since their characteristics are the most suitable for interpreting OD models. Saliency maps are derived directly from the activations or gradient of a chosen layer with respect to the input image. This approach offers two primary advantages for our research: 1) computational efficiency – saliency maps produce their output faster than other XAI

Target model	COCO Random Noise Patch	COCO One-Stage Patch	COCO Two/Multi-Stage Patch	SuperStore One-Stage Patch	SuperStore Two/Multi-Stage Patch
One-Stage	28%	85%	66%	100%	16%
Two-Stage	20%	55%	85%	61%	72%
Multi-Stage	28%	57%	76%	72%	75%

Table 2. Adversarial attacks' success rate against one-, two-, and multi-stage OD models; the gray cells indicate the datasets chosen for evaluation.

methods (such as LIME and SHAP). Their reliance on activations or gradients, which are computed through a single forward and backward pass, ensures rapid calculations. The saliency maps' efficiency is especially crucial in our research, where we evaluate OD models in real time in the inference phase. 2) simplicity – saliency maps are considered relatively straightforward and easy to understand. Unlike other XAI techniques, saliency maps disregard feature interactions which can lead to visually complex explanations. Since one of our research goals is to visually represent a model's internal perception, the explanations' clarity is essential.

In our research, we evaluated four saliency map techniques: GradCAM [33], GradCAM++ [1], EigenCAM [27], and enhanced EigenGradCAM. GradCAM [33] and GradCAM++ [1] rely on the model's gradients, whereas EigenCAM [27] relies on the model's activations. The enhanced EigenGradCAM technique relies both on the model's gradients and activations. Figure 4 presents the output of each saliency map technique for images from the digital COCO clean and physical SuperStore clean use cases. Since DiL was inspired by the localization objective [20, 21], it relies on a saliency map's ability to discriminate between the object and its background, i.e., the saliency map will have higher values in pixels related to any object. Consequently, saliency maps that are well-localized contribute to more consistent DiL scores. On the other hand, saliency maps that are either too dense (the focus is concentrated in the center of the object) or too noisy (the focus extends beyond the object's boundaries) lead to inferior DiL scores. In Figure 4 it can be seen that the saliency maps derived from the GradCAM technique appear to be the most localized; the saliency maps derived from the GradCAM++ technique can be perceived as noisy; and the saliency maps derived from the EigenCAM and EigenGradCAM techniques can be perceived as dense. Table 6 presents a comparison of the mean DiL scores obtained using those four techniques. Each cell in the table presents the mean DiL score for one-, two-, and multi-stage models corresponding to a specific saliency map technique. The results in the table show that the DiL scores obtained with the GradCAM technique are the most effective in distinguishing between clean and abnormal scenes. While all of the examined techniques yield high DiL scores for abnormal scenes, GradCAM consistently produces the lowest scores for clean scenes. More detailed Dil results for each of the examined saliency map techniques are presented in Tables 7-10. Those results presented in the tables are aligned with the findings presented in Figure 4. Since the explanations obtained from the GradCAM technique are notably localized, they effectively capture the model's perception in both clean and abnormal scenes.

5.2. Robustness Assessment of the Objectness Saliency Map

Furthermore, we evaluated the robustness of the objectness saliency map used by DiL. When mapping the model's final outputs, saliency maps can be susceptible to minor scene variations. However, our approach diverges by mapping the model's objectness and not the final prediction. This difference enhances the saliency map's robustness to changes in the input scene; only a drastic change will cause the model to "ignore" the indications for objects in the scene, thus only substantial alterations in the objectness scores will impact the outputted saliency map. To further support this claim, we performed a sensitivity analysis of the objectness saliency map when encountering noisy samples. We added uniform random noise on scales of 0.1 and 0.2 to 100 clean images and evaluated the changes in their saliency maps. The changes were quantified by measuring the MSE distance between the saliency maps produced for each noisy and clean pair of samples. The results indicate that the average MSE values were 0.008 and 0.017 with a standard deviation of 0.083 and 0.11 for random noise of 0.1 and 0.2 respectively, indicating the objectness saliency map robustness. Examples of the clean and noisy samples with various noise levels used in this analysis can be seen in Figure 5.

5.3. Label-Uncertainty Techniques Implementations and Quantitative Analysis

In the main manuscript, we argue that existing label-uncertainty techniques are less effective when applied in abnormal scenarios, as demonstrate in various output examples in Figure 3b. To perform this demonstration, we implemented and evaluated three established label-uncertainty techniques: Bayesian estimation [9], Monte Carlo dropout [5], and Ensemble methods [18]. These implementations were based on publicly available code and

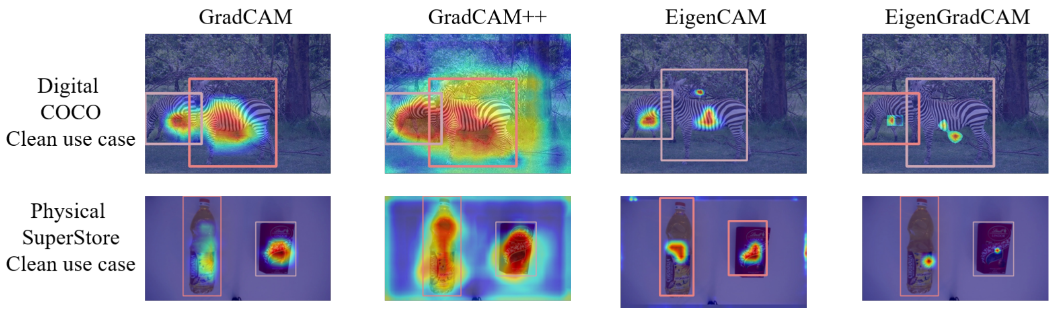

Figure 4. Examples for saliency map techniques outputs on the clean COCO and clean Superstore datasets.

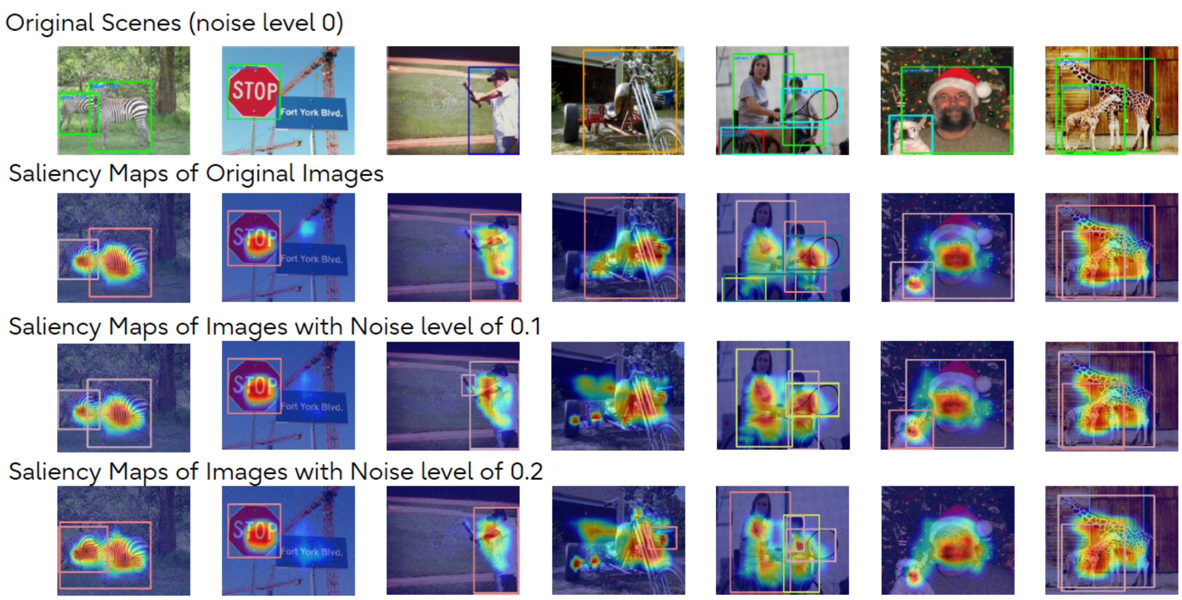

Figure 5. Examples of the clean and noisy samples with various noise levels used in the objectness saliency map sensitivity analysis.

models detailed in [10][1], enabling us to compare various uncertainty techniques applied to the Faster R-CNN model. We tested these techniques using our dataset of abnormal use cases and assessed their effectiveness. Additionally, we explored several newer techniques cited in [6,19,26,35,42]. However, the lack of available code implementations for these methods hindered our ability to reproduce them reliably.

In this supplementary section, we extend the qualitative experiment from the paper with a quantitative analysis to further demonstrate the breadth of our findings. We applied the three label-uncertainty techniques to a Faster R-CNN model across various abnormal use cases in the digital domain, including unrealistic partial occlusion, realistic partial occlusion, out-of-distribution objects, and adversarial attack scenarios. Our objective is to demonstrate that abnormalities can effectively deceive the target object detection model into missing objects, which occurs on an earlier stage of the predicion process, before the uncertainty techniques are employed. To quantify this, we use the misclassification metric, which tracks the portion of scenes where the object detection model failed to detect the targeted object due to abnormalities. Table 3 compares the misclassification rates obtained from the base model and the three uncertainty techniques across the abnormal use cases.

Although the usage of uncertainty techniques yields a modest improvement in misclassification rates—indicative of a reduction in errors—more than 50% of cases still result in misclassification, underscoring the persistent challenges these techniques face in effectively addressing abnormalities. This finding supports our assumption that these techniques, designed to assess label uncertainty and thus applied at the final stages of the prediction process, are less effective in abnormal scenarios. The impact of abnormalities occurs at earlier stages of the model's prediction process, preventing it from 'proposing' objects to be processed by the un-

[1] https://github.com/asharakeh/probdet

certainty technique

Model	Unrealistic PO	Realistic PO	OOD	Adversarial
Base model	0.89	0.93	0.83	0.85
Bayes-OD	0.93	0.93	0.77	0.61
Dropout	0.59	0.68	0.55	0.56
Ensemble	0.92	0.95	0.78	0.60

Table 3. Miscalssificaiotn rate of the Faster RCNN model in abnormal scenarios using various label-uncertainty technique.

5.4. Runtime Analysis

Since the DiL metric is used during inference, it should be as efficient as possible. To calculate the DiL score for a given input scene, one should obtain the input scene's predictions and the saliency map of the model's objectness. Since the predictions for a scene are computed during inference, DiL's additional runtime overhead stems primarily from generating the saliency map and the final DiL score's computation time. When employing a saliency map technique that only uses the model's activations, the saliency map is produced in parallel with the model's predictions. When employing a saliency map technique that uses the gradients, the saliency map generation requires a single backpropagation. In addition to the saliency map generation overhead (if any), there is a subsequent final DiL score calculation consisting of basic mathematical operations on the produced saliency map, which is computationally trivial. Thus, the additional overhead for DiL score computation for a single scene is essentially one backpropagation (if any), which is a relatively small addition to the total inference time.

5.5. Comparative Analysis of DiL in Unrealistic vs. Realistic Partially Occluded Scenarios

Throughout the experiments establishing Table 2 in the main manuscript, we observed consistent trends in the DiL scores for the COCO PO use cases (2-3), with slight differences - the mean DiL score of the unrealistic PO use case was slightly higher than the realistic PO use case. This may occur due to the varying levels of alienation from the distribution of normal scenes. The distribution of scenes used in the unrealistic PO use case is more alienated from that of normal scenes due to their creation process - objects are cropped and pasted onto different backgrounds, leading to unnatural combinations like a 'pizza' with a sky background. Conversely, the PO scenes in the realistic PO use case were designed to simulate real-world scenes, resulting in a closer resemblance to natural scenes.

5.6. Challenges in DiL's Detection Capabilities

In our experiments, DiL effectively reflected abnormalities in most cases but had limited success in a small fraction of scenes. Those scenes reflected DiL's limitation and were characterized by a specific layout of objects in which the object related to the abnormality was surrounded by other objects, causing it to fall into other objects' bounding boxes. This occurs when an object is shaped in such a way that it cannot fit within a bounding box without including a large portion of the background. In those cases, the *BL* does not consider the object related to the abnormality, since it is covered by other bounding boxes, resulting in a lower DiL score than expected. An additional potential limitation could be DiL's effectiveness when concerning extremely small objects.

5.7. DiL Robustness Additional Results

The results presented in our paper show that the DiL metric can be utilized to enhance the model's performance when faced with abnormal scenes, as described in Section 3.2. This enhancement is achieved by using a a dynamic decision threshold (DDT) that changes based on the DiL score, rather than using a fixed decision threshold. A higher DiL score indicates that an abnormal scene has been presented and prompts a reduction in the detection threshold. In our experiments on the DDT, we observed that lowering the decision threshold improved the recall value for the abnormal scenes at a minor cost in the precision of the clean scenes. Consequently, we selected the GradCAM++ technique, which consistently resulted in the highest DiL values. Table 11 provides an extended analysis of DiL's robustness when using the DDT, as described in Section 5 of our paper. The table presents the performance metrics for one-, two-, and multi-stage models across all nine use cases. The results presented in the table support our claim that the use of DDT mitigates the abnormalities' effect without harming the model's performance (FPR).

6. WACV revision additions

6.1. Applying DiL on Additional Object detection Models

DiL calculation heavily relies on the object detection objectness score However, an alternative approach is necessary for models like SSD, RetinaNet, or non-CNN architectures such as DETR or ViT, which do not inherently produce objectness values as part of the prediction process. In these instances, classification logits can effectively be used to create saliency maps.

To explore the efficacy of this method, we conducted an experiment using the classification logits from an SSD model across the various use cases within the COCO dataset (clean and abnormal). The findings, detailed in Table 4, illustrate that while the DiL values tend to be higher in clean scenarios, there is a noticeable distinction between the DiL values obtained in the clean scenario compared to those from abnormal scenarios. This variation highlights the utility of classification logits in enhancing model interpretability, particularly when objectness values are unavailable.

	Clean	Unrealistic PO	Realistic PO	OOD	Adversarial
CL	0.26	0.34	0.30	0.41	0.33
BL	0.12	0.29	0.23	0.40	0.28
DiL	0.42	0.79	0.67	0.95	0.75

Table 4. Applying DiL on SSD model relying on class logistic rather then objectness score.

6.2. Experimenting with Minimal Negative Sample Filtering

DiL calculation heavily relies on the object detection objectness score. While YOLO, a one-stage model series, directly produces objectness scores during the prediction process, two- and multi-stage models generate *proposal candidates* through the RPN. Typically, these models produce more ”background” candidates than ”object” candidates, which can distort the DiL scores. To address this imbalance, we employed minimal negative sample filtering to better balance these two groups. In our evaluation, we experimented with generating DiL scores from both the base and the filtered proposals. The results presented in the paper indicate that DiL scores based on the base proposals are superior.

In this section, we elaborate on our filtering approach and the qualitative results obtained. Our experiments included two methods of filtering: hard filtering and weighted filtering. Hard filtering employs a non-differentiable operation using a threshold that blocks gradient flow. This method resulted in blank saliency maps, as it does not allow for gradient-based data propagation.

On the other hand, weighted filtering adjusts the impact of each proposal based on its objectness score. While theoretically promising, weighted filtering presented challenges in our tests. It tended to disproportionately emphasize central regions of the image where objectness scores are usually higher. This characteristic of the weighted approach proved problematic in scenarios involving adversarial attacks. Such attacks typically involve strategically placed patches at the center of objects, which artificially lower the objectness scores in these central areas. Consequently, this manipulation led to distorted DiL scores, suggesting that the system might overestimate the certainty (and consequently robustness) of the model in the face of adversarial inputs. Figure 6 shows an example of saliency map outputs with and without filtering.

6.3. DDT with smart degradation factor - Future work

We believe that the degradation factor can be set in a smart manner based on the evidence for an object present

Figure 6. Saliency map outputs with various filtering techniques.

in the background (a.k.a, ”undetected hot areas”). The further away an ”undetected hot area” is from any recognized bounding box, the less likely it is to be associated with that detected object. Hence, this is evidence of a different object from the detected one. Therefore, in scenarios where the “undetected hot area” is near a detected object, a smaller decrease in the threshold would be sufficient. (Since there is less evidence of an undetected object) and vice versa. The degradation factor can be computed by the distance from “undetected hot areas” and existing bounding boxes. By that, the degradation factor further enhances the effectiveness of DDT.

6.4. Additional comparison of mAP and DiL

Table 5 presents the mean DiL score, mAP, normalized mAP, and the decrease in mAP. The normalized mAP and the decrease in mAP are calculated with respect to the mAP in the clean use case. The first row (DiL mean) and the last row (mAP norm. decrease) can be compared as they both range between 0-1 and have the same expected behavior (indicated by the arrows).

When comparing the two rows we can see that both DiL and mAP are aligned featuring low scores in the clean use cases and higher scores in abnormal use cases. However, when examining the relations between the scores for different abnormalities, DiL is more stable. The DiL scores of OOD use cases are the highest, PO use cases are the lowest, and adversarial use cases are in the middle. In contrast, the mAP scores show different relations between abnormalities between the COCO use cases and the Superstore use cases. In the COCO use cases, the behavior of the mAP scores is aligned with the DiL scores. However, in the Superstore use cases, the adversarial use case has a higher score than the OOD use case.

These phenomena show that the DiL scores are more stable as an uncertainty metric than mAP when the model is faced with an abnormal scene.

Model Type	Metric	Use case								
		[1] Clean	[2] Unrealistic PO	[3] Realistic PO	[4] OOD	[5] Adv.	[6] Clean	[7] PO	[8] OOD	[9] Adv.
All types	DiL mean	**0.154 ↓**	**0.54 ↑**	**0.497 ↑**	**0.911 ↑**	**0.557 ↑**	**0.09 ↓**	**0.39 ↑**	**0.73 ↑**	**0.63 ↑**
All types	mAP	0.359 ↑	0.23* ↓	0.253 ↓	0.0 ↓	0.212 ↓	0.9 ↑	0.5 ↓	0.5 ↓	0.01 ↓
All types	Normilized mAP	1.0 ↑	0.64 ↓	0.704 ↓	0.0 ↓	0.59 ↓	1.0 ↑	0.555 ↓	0.555 ↓	0.111 ↓
All types	mAP norm. decrease	0.0 ↓	0.35 ↑	0.29 ↑	1.0 ↑	0.409 ↑	0.0 ↓	0.444 ↑	0.444 ↑	0.988 ↑

Table 5. Mean DiL scores and mAP for all types of OD models in the digital COCO (1-5) and physical Superstore use cases (6-9).

	Digital COCO use cases					Physical SuperStore use cases			
Saliency map technique	Clean	Unrealstic PO	Realistic PO	OOD	Adv.	Clean	PO	OOD	Adv.
GradCAM	0.158	0.535	0.504	0.914	0.563	0.084	0.406	0.71	0.625
GradCAM++	0.487	0.667	0.629	0.898	0.686	0.65	0.787	0.851	0.823
EigenCAM	0.312	0.584	0.536	0.89	0.765	0.657	0.742	0.83	0.757
EigenGradCAM	0.215	0.548	0.482	0.914	0.626	0.199	0.499	0.894	0.755

Table 6. Mean DiL scores for each saliency map technique for every use case. DiL scores obtained using the GradCAM technique are the most productive at differentiating between clean and abnormal scenes.

			Digital COCO use case					Physical SuperStore use case			
Target model type	Target model	Metric	Clean	Unrealistic PO	Realistic PO	OOD	Adv.	Clean	PO	OOD	Adv.
One-stage	YOLOv5	Complete localization	0.011	0.01	0.0047	0.002	0.012	0.017	0.004	0.018	0.02
		Background localization	0.003	0.0043	0.0028	0.002	0.008	3E-04	0.002	0.01	0.008
		DiL	0.228	0.43	0.5957	0.917	0.672	0.018	0.622	0.556	0.396
	YOLOF	Complete localization	0.013	0.01	0.0049	0.004	0.018	0.017	0.014	0.021	0.021
		Background localization	0.003	0.0056	0.003	0.004	0.009	3E-05	0.005	0.009	0.008
		DiL	0.234	0.56	0.6122	0.951	0.497	0.002	0.379	0.414	0.367
	YOLOv3	Complete localization	0.011	0.01	0.0046	0.004	0.018	0.017	0.014	0.02	0.021
		Background localization	0.002	0.0042	0.0027	0.004	0.007	0.001	0.004	0.011	0.006
		DiL	0.184	0.42	0.587	0.947	0.389	0.076	0.25	0.55	0.267
Two-stage	Faster R-CNN	Complete localization	0.116	0.155	0.15	0.089	0.069	0.039	0.025	0.061	0.014
		Background localization	0.01	0.099	0.063	0.077	0.047	0.005	0.01	0.057	0.011
		DiL	0.086	0.63871	0.42	0.87	0.681	0.128	0.4	0.934	0.786
	Grid R-CNN	Complete localization	0.114	0.155	0.15	0.091	0.067	0.039	0.022	0.061	0.014
		Background localization	0.017	0.091	0.075	0.083	0.042	0.004	0.01	0.044	0.011
		DiL	0.149	0.5871	0.5	0.914	0.621	0.103	0.455	0.721	0.793
	Double Heads R-CNN	Complete localization	0.114	0.155	0.149	0.088	0.068	0.037	0.02	0.061	0.014
		Background localization	0.015	0.084	0.062	0.081	0.039	0.004	0.006	0.038	0.012
		DiL	0.132	0.54194	0.4161	0.92	0.574	0.108	0.3	0.623	0.821
Multi-stage	Cascade R-CNN	Complete localization	0.114	0.1559	0.149	0.088	0.066	0.04	0.025	0.06	0.011
		Background localization	0.016	0.095	0.072	0.081	0.035	0.005	0.01	0.058	0.008
		DiL	0.14	0.60936	0.4832	0.918	0.527	0.127	0.4	0.967	0.752
	Cascade RPN	Complete localization	0.114	0.1539	0.1496	0.086	0.066	0.04	0.023	0.06	0.008
		Background localization	0.013	0.0762	0.062	0.075	0.036	0.006	0.009	0.055	0.007
		DiL	0.11	0.49513	0.4144	0.872	0.545	0.139	0.391	0.917	0.821
All types	**All types**	**Mean DiL**	**0.154**	**0.54**	**0.497**	**0.911**	**0.557**	**0.089**	**0.4**	**0.73**	**0.63**

Table 7. DiL scores using GradCAM saliency map technique.

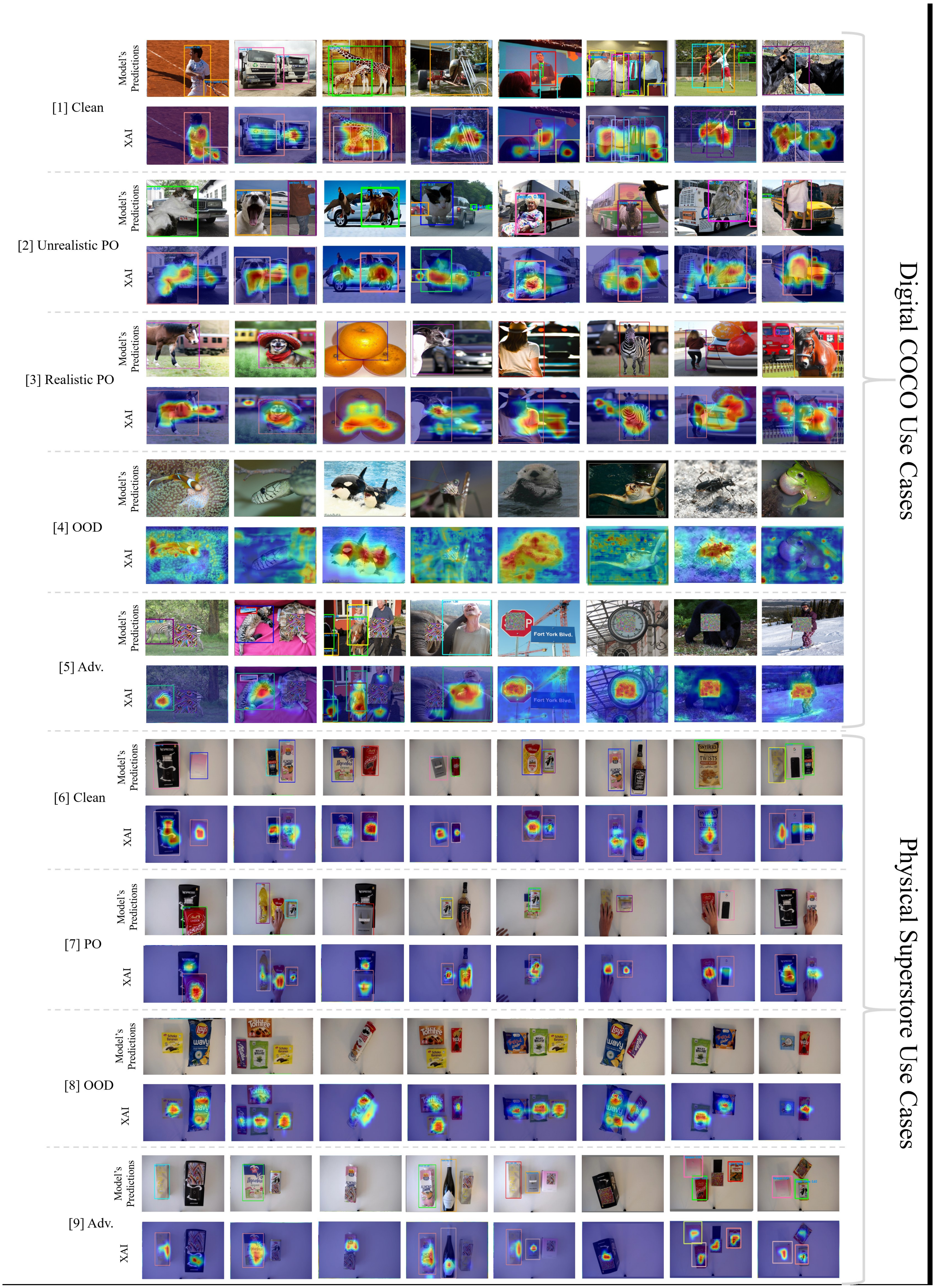

Figure 7. Various datasets used in each evaluation use case and their corresponding saliency maps.

			Digital COCO use case					Physical SuperStore use case			
Target model type	Target model	Metric	Clean	Unrealistic PO	Realistic PO	OOD	Adv.	Clean	PO	OOD	Adv.
One-stage	YOLOv5	Complete localization	0.3	0.33	0.304	0.322	0.269	0.35	0.47	0.345	0.345
		Background localization	0.194	0.23	0.235	0.275	0.227	0.27	0.44	0.289	0.302
		DiL	0.634	0.699	0.79	0.85	0.855	0.77	0.93	0.826	0.858
	YOLOF	Complete localization	0.3	0.333	0.3063	0.325	0.266	0.35	0.305	0.345	0.346
		Background localization	0.18	0.242	0.2058	0.308	0.17	0.27	0.268	0.264	0.273
		DiL	0.6	0.716	0.679	0.944	0.65	0.75	0.874	0.753	0.775
	YOLOv3	Complete localization	0.299	0.334	0.3075	0.325	0.267	0.35	0.305	0.345	0.34
		Background localization	0.16	0.198	0.185	0.288	0.156	0.28	0.259	0.282	0.29
		DiL	0.56	0.587	0.619	0.879	0.577	0.77	0.851	0.806	0.83
Two-stage	Faster R-CNN	Complete localization	0.533	0.572	0.576	0.52	0.539	0.339	0.32	0.4	0.441
		Background localization	0.201	0.415	0.336	0.446	0.42	0.2	0.245	0.38	0.39
		DiL	0.389	0.726	0.584	0.858	0.792	0.607	0.77	0.95	0.9
	Grid R-CNN	Complete localization	0.53	0.57	0.578	0.519	0.54	0.341	0.32	0.404	0.44
		Background localization	0.23	0.38	0.366	0.48	0.37	0.198	0.226	0.317	0.34
		DiL	0.45	0.67	0.637	0.919	0.71	0.59	0.71	0.79	0.79
	Double Heads R-CNN	Complete localization	0.533	0.57	0.577	0.519	0.53	0.341	0.322	0.4	0.44
		Background localization	0.216	0.37	0.322	0.476	0.34	0.176	0.218	0.31	0.35
		DiL	0.41	0.64	0.562	0.915	0.65	0.56	0.68	0.77	0.8
Multi-stage	Cascade R-CNN	Complete localization	0.53	0.57	0.576	0.52	0.53	0.341	0.32	0.4	0.44
		Background localization	0.238	0.4	0.349	0.484	0.34	0.19	0.241	0.397	0.36
		DiL	0.454	0.7	0.607	0.92	0.646	0.55	0.75	0.98	0.82
	Cascade RPN	Complete localization	0.535	0.57	0.578	0.519	0.57	0.341	0.322	0.4	0.442
		Background localization	0.2	0.346	0.317	0.466	0.35	0.195	0.232	0.374	0.361
		DiL	0.4	0.6	0.553	0.896	0.61	0.6	0.733	0.93	0.81
All types	**All types**	**Mean DiL**	**0.487**	**0.66725**	**0.6289**	**0.898**	**0.686**	**0.65**	**0.787**	**0.851**	**0.823**

Table 8. DiL scores using GradCAM++ saliency map technique.

			Digital COCO use case					Physical SuperStore use case			
Target model type	Target model	Metric	Clean	Unrealistic PO	Realistic PO	OOD	Adv.	Clean	PO	OOD	Adv.
One-stage	YOLOv5	Complete localization	0.035	0.041	0.0291	0.027	0.058	0.033	0.027	0.036	0.036
		Background localization	0.012	0.023	0.0195	0.022	0.048	0.009	0.011	0.022	0.019
		DiL	0.342	0.56098	0.6701	0.806	0.825	0.273	0.401	0.6	0.534
	YOLOF	Complete localization	0.035	0.04	0.0291	0.027	0.058	0.033	0.029	0.037	0.035
		Background localization	0.011	0.025	0.017	0.025	0.055	0.008	0.01	0.022	0.017
		DiL	0.322	0.625	0.58419	0.925	0.953	0.248	0.353	0.601	0.484
	YOLOv3	Complete localization	0.035	0.0417	0.0291	0.027	0.059	0.033	0.029	0.036	0.037
		Background localization	0.01	0.019	0.0147	0.023	0.038	0.011	0.013	0.026	0.017
		DiL	0.286	0.45564	0.50515	0.852	0.643	0.333	0.443	0.714	0.457
Two-stage	Faster R-CNN	Complete localization	0.251	0.246	0.255	0.265	0.237	0.322	0.318	0.322	0.209
		Background localization	0.067	0.163	0.1248	0.234	0.202	0.286	0.306	0.319	0.204
		DiL	0.267	0.6626	0.48941	0.883	0.852	0.888	0.962	0.991	0.975
	Grid R-CNN	Complete localization	0.251	0.246	0.256	0.265	0.236	0.322	0.318	0.322	0.209
		Background localization	0.085	0.153	0.144	0.242	0.178	0.285	0.298	0.287	0.184
		DiL	0.339	0.62195	0.5625	0.913	0.754	0.885	0.937	0.891	0.879
	Double Heads R-CNN	Complete localization	0.251	0.246	0.252	0.264	0.238	0.322	0.318	0.322	0.209
		Background localization	0.077	0.141	0.119	0.241	0.171	0.279	0.298	0.282	0.187
		DiL	0.307	0.57317	0.47222	0.913	0.718	0.866	0.937	0.876	0.895
Multi-stage	Cascade R-CNN	Complete localization	0.251	0.246	0.253	0.266	0.238	0.322	0.318	0.322	0.209
		Background localization	0.09	0.157	0.134	0.25	0.178	0.282	0.303	0.32	0.195
		DiL	0.359	0.63821	0.52964	0.94	0.748	0.876	0.953	0.994	0.933
	Cascade RPN	Complete localization	0.251	0.246	0.2557	0.265	0.237	0.322	0.319	0.322	0.209
		Background localization	0.07	0.131	0.121	0.236	0.149	0.286	0.302	0.313	0.189
		DiL	0.279	0.53252	0.47321	0.891	0.629	0.889	0.948	0.972	0.903
All types	**All types**	**Mean DiL**	**0.312**	**0.58376**	**0.5358**	**0.89**	**0.765**	**0.657**	**0.742**	**0.83**	**0.757**

Table 9. DiL scores using EigenCAM saliency map technique.

			Digital COCO use case					Physical SuperStore use case			
Target model type	Target model	Metric	Clean	Unrealistic PO	Realistic PO	OOD	Adv.	Clean	PO	OOD	Adv.
One-stage	YOLOv5	Complete localization	0.016	0.005	0.004	0.017	0.011	0.002	0.002	0.001	0.003
		Background localization	0.006	0.003	0.002	0.015	0.009	9E-04	0.001	0.001	0.002
		DiL	0.35	0.547	0.653	0.875	0.772	0.45	0.722	0.979	0.84
	YOLOF	Complete localization	0.016	0.005	0.004	0.014	0.011	0.002	0.001	0.001	0.003
		Background localization	0.006	0.003	0.002	0.013	0.011	8E-04	0.001	0.001	0.002
		DiL	0.374	0.631	0.525	0.963	0.947	0.4	0.929	0.966	0.84
	YOLOv3	Complete localization	0.016	0.006	0.004	0.015	0.012	0.002	0.002	0.001	0.003
		Background localization	0.005	0.003	0.002	0.014	0.006	9E-04	0.001	0.001	0.002
		DiL	0.307	0.455	0.493	0.945	0.543	0.45	0.813	0.986	0.84
Two-stage	Faster R-CNN	Complete localization	0.011	0.011	0.014	0.014	0.012	0.01	0.01	0.012	0.032
		Background localization	0.001	0.007	0.006	0.012	0.008	8E-04	0.003	0.011	0.024
		DiL	0.091	0.627	0.396	0.876	0.672	0.082	0.309	0.948	0.75
	Grid R-CNN	Complete localization	0.011	0.012	0.015	0.014	0.012	0.01	0.01	0.012	0.049
		Background localization	0.002	0.007	0.007	0.013	0.007	7E-04	0.004	0.009	0.034
		DiL	0.177	0.556	0.486	0.906	0.565	0.067	0.354	0.737	0.688
	Double Heads R-CNN	Complete localization	0.011	0.012	0.014	0.013	0.011	0.01	0.01	0.012	0.048
		Background localization	0.002	0.006	0.006	0.012	0.006	1E-04	0.002	0.008	0.034
		DiL	0.15	0.513	0.419	0.925	0.504	0.01	0.227	0.661	0.701
Multi-stage	Cascade R-CNN	Complete localization	0.011	0.011	0.015	0.014	0.011	0.01	0.01	0.012	0.039
		Background localization	0.002	0.007	0.007	0.013	0.006	6E-04	0.003	0.011	0.026
		DiL	0.171	0.575	0.472	0.942	0.524	0.061	0.333	0.974	0.664
	Cascade RPN	Complete localization	0.011	0.012	0.015	0.014	0.012	0.01	0.01	0.012	0.044
		Background localization	0.001	0.006	0.006	0.012	0.006	7E-04	0.003	0.011	0.031
		DiL	0.099	0.478	0.413	0.884	0.478	0.076	0.303	0.897	0.715
All types	**All types**	**Mean DiL**	**0.215**	**0.548**	**0.482**	**0.914**	**0.626**	**0.199**	**0.499**	**0.894**	**0.755**

Table 10. DiL scores using EigenGradCAM saliency map technique.

Target model	Metric	Use case and abnormality								
		[1] Clean	[2] Unrealistic PO	[3] Realistic PO	[4] OOD	[5] Adv.	[6] Clean	[7] PO	[8] OOD	[9] Adv.
One-stage	Base recall	0.594	0.51	0.463	0.275	0.439	0.94	0.563	0.25	0.676
	With DDT	0.659 (**+10%**)	0.698 (**+36%**)	0.629 (**+35%**)	0.4 (**+45%**)	0.51 (**+16%**)	0.94 (**0%**)	0.575 (**+2%**)	0.68 (**+270%**)	0.7 (**+3%**)
	TP improvement	12%	28%	25%	16%	13%	0%	12%	7%	9%
	FPR	4%	6%	2%	15%	4%	1%	0.3%	8%	3%
Two-stage	Base recall	0.66	0.517	0.469	0.26	0.49	0.95	0.646	0.23	0.6
	With DDT	0.69 (**+4%**)	0.76 (**+47%**)	0.625 (**+33%**)	0.36 (**+38%**)	0.6 (**+22%**)	0.95 (**0%**)	0.7 (**+8%**)	0.44 (**+91%**)	0.715 (**+19%**)
	TP improvement	8%	52%	29%	13%	21%	0%	31%	26%	24%
	FPR	4.2%	8%	4%	30%	5%	1%	0.7%	2%	5%
Multi-stage	Base recall	0.625	0.505	0.484	0.19	0.525	0.835	0.537	0	0.56
	With DDT	0.665 (**+6%**)	0.68 (**+36%**)	0.593 (**+22%**)	0.32 (**+68%**)	0.605 (**+15%**)	0.845 (**1%**)	0.591 (**+10%**)	0.17 (**+%**)	0.6 (**+7%**)
	TP improvement	3%	35%	21%	15%	3%	12%	17%	7%	9%
	FPR	3%	4%	1%	25%	4%	0.5%	0.3%	0%	0.4%

Table 11. Original and DDT performance for all OD models' types and all use cases.

References

[1] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. 5

[2] Ping-Han Chiang, Chi-Shen Chan, and Shan-Hung Wu. Adversarial pixel masking: A defense against physical attacks for pre-trained object detectors. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1856–1865, 2021. 2

[3] Xuefeng Du, Xin Wang, Gabriel Gozum, and Yixuan Li. Unknown-aware object detection: Learning what you don't know from videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13678–13688, 2022. 2

[4] Patrick Follmann, Rebecca König, Philipp Härtinger, Michael Klostermann, and Tobias Böttger. Learning to see the invisible: End-to-end trainable amodal instance segmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1328–1336. IEEE, 2019. 2

[5] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 2, 5

[6] Stefano Gasperini, Jan Haug, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, Benjamin Busam, and Federico Tombari. Certainnet: Sampling-free uncertainty estimation for object detection. *IEEE Robotics and Automation Letters*, 7(2):698–705, 2021. 2, 6

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 2

[8] David Hall, Feras Dayoub, John Skinner, Haoyang Zhang, Dimity Miller, Peter Corke, Gustavo Carneiro, Anelia Angelova, and Niko Sunderhauf. Probabilistic object detection: Definition and evaluation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1031–1040, 2020. 2

[9] Ali Harakeh, Michael Smart, and Steven L Waslander. Bayesod: A bayesian approach for uncertainty estimation in deep object detectors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 87–93. IEEE, 2020. 2, 5

[10] Ali Harakeh and Steven L Waslander. Estimating and evaluating regression predictive uncertainty in deep object detectors. *arXiv preprint arXiv:2101.05036*, 2021. 6

[11] Vahid Hashemi, Jan Křetínskỳ, Sabine Rieder, and Jessica Schmidt. Runtime monitoring for out-of-distribution detection in object detection neural networks. In *International Symposium on Formal Methods*, pages 622–634. Springer, 2023. 2

[12] Omer Hofman, Amit Giloni, Yarin Hayun, Ikuya Morikawa, Toshiya Shimizu, Yuval Elovici, and Asaf Shabtai. X-detect: Explainable adversarial patch detection for object detectors in retail. *arXiv preprint arXiv:2306.08422*, 2023. 2

[13] Chengjie Huang, Vahdat Abdelzad, Christopher Gus Mannes, Luke Rowe, Benjamin Therien, Rick Salay, Krzysztof Czarnecki, et al. Out-of-distribution detection for lidar-based 3d object detection. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 4265–4271. IEEE, 2022. 2

[14] Nan Ji, YanFei Feng, Haidong Xie, Xueshuang Xiang, and Naijin Liu. Adversarial yolo: Defense human detection patch attacks via detecting adversarial patches. *arXiv preprint arXiv:2103.08860*, 2021. 2

[15] Davood Karimi and Ali Gholipour. Improving calibration and out-of-distribution detection in deep models for medical image segmentation. *IEEE Transactions on Artificial Intelligence*, 4(2):383–397, 2022. 2

[16] Taeheon Kim, Youngjoon Yu, and Yong Man Ro. Defending physical adversarial attack on object detection via adversarial patch-feature energy. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1905–1913, 2022. 2

[17] Florian Kraus and Klaus Dietmayer. Uncertainty estimation in one-stage object detection. In *2019 ieee intelligent transportation systems conference (itsc)*, pages 53–60. IEEE, 2019. 2

[18] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 2, 5

[19] Youngwan Lee, Joong-won Hwang, Hyung-Il Kim, Kimin Yun, Yongjin Kwon, Yuseok Bae, and Sung Ju Hwang. Localization uncertainty estimation for anchor-free object detection. In *European Conference on Computer Vision*, pages 27–42. Springer, 2022. 6

[20] Xuhong Li, Haoyi Xiong, Xingjian Li, Xuanyu Wu, Xiao Zhang, Ji Liu, Jiang Bian, and Dejing Dou. Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. *Knowledge and Information Systems*, 64(12):3197–3234, 2022. 5

[21] Xiao-Hui Li, Yuhan Shi, Haoyang Li, Wei Bai, Yuanwei Song, Caleb Chen Cao, and Lei Chen. Quantitative evaluations on saliency methods: An experimental study. *arXiv preprint arXiv:2012.15616*, 2020. 5

[22] Yimeng Li and Jana Košecká. Uncertainty aware proposal segmentation for unknown object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 241–250, 2022. 2

[23] Tsung-Yi Lin. Y, dollár p, girshick r, he k, hariharan b, belongie s. feature pyramid networks for object detection. In *Proceedings-30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, volume 2017, pages 936–944, 2017. 2

[24] Jiang Liu, Alexander Levine, Chun Pong Lau, Rama Chellappa, and Soheil Feizi. Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14973–14982, 2022. 2

[25] Xin Liu, Huanrui Yang, Ziwei Liu, Linghao Song, Hai Li, and Yiran Chen. Dpatch: An adversarial patch attack on object detectors. *arXiv preprint arXiv:1806.02299*, 2018. 3

[26] Dimity Miller, Niko Sünderhauf, Michael Milford, and Feras Dayoub. Uncertainty for identifying open-set errors in visual object detection. *IEEE Robotics and Automation Letters*, 7(1):215–222, 2021. 6

[27] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2020. 5

[28] 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)*, pages 504–519, 2018. 2

[29] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[30] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2019. 2

[31] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. *URL https://arxiv.org/abs/2204.06125*, 7, 2022. 3

[32] Kaziwa Saleh, Sándor Szénási, and Zoltán Vámossy. Occlusion handling in generic object detection: A review. In *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 000477–000484. IEEE, 2021. 2

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 5

[34] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan L Yuille. Robust object detection under occlusion with context-aware compositionalnets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12645–12654, 2020. 2

[35] Ke Wang, Yong Wang, Bingjun Liu, and Junlan Chen. Quantification of uncertainty and its applications to complex domain for autonomous vehicles perception system. *IEEE Transactions on Instrumentation and Measurement*, 72:1–17, 2023. 6

[36] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2606–2615, 2017. 2

[37] Zining Wang, Di Feng, Yiyang Zhou, Lars Rosenbaum, Fabian Timm, Klaus Dietmayer, Masayoshi Tomizuka, and Wei Zhan. Inferring spatial uncertainty in object detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5792–5799. IEEE, 2020. 2

[38] Everingham M Van Gool L Williams. Ck winn j zisserman a the pascal visual object classes (voc) challenge. *Int. J. Comput. Vis*, 88(2):303, 2010. 2

[39] Chong Xiang and Prateek Mittal. Detectorguard: Provably securing object detectors against localized patch hiding attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3177–3196, 2021. 2

[40] Chong Xiang, Alexander Valtchanov, Saeed Mahloujifar, and Prateek Mittal. Objectseeker: Certifiably robust object detection against patch hiding attacks via patch-agnostic masking. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1329–1347. IEEE, 2023. 2

[41] Ke Xu, Yao Xiao, Zhaoheng Zheng, Kaijie Cai, and Ram Nevatia. Patchzero: Defending against adversarial patch attacks by detecting and zeroing the patch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4632–4641, 2023. 2

[42] Sai Harsha Yelleni, Deepshikha Kumari, PK Srijith, et al. Monte carlo dropblock for modeling uncertainty in object detection. *Pattern Recognition*, 146:110003, 2024. 6

[43] Xiaohang Zhan, Xingang Pan, Bo Dai, Ziwei Liu, Dahua Lin, and Chen Change Loy. Self-supervised scene de-occlusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3784–3792, 2020. 2