Angelic Patches for Improving Third-Party Object Detector Performance

Wenwen Si[1], Shuo Li[1], Sangdon Park[2], Insup Lee[1], Osbert Bastani[1]
[1]Dept. of Computer & Info. Science, University of Pennsylvania
[2]School of Cybersecurity & Privacy, Georgia Institute of Technology
{wenwens, lishuo1, lee, obastani}@seas.upenn.edu, sangdon@gatech.edu

Abstract

Deep learning models have shown extreme vulnerability to distribution shifts such as synthetic perturbations and spatial transformations. In this work, we explore whether we can adopt the characteristics of adversarial attack methods to help improve robustness of object detection to distribution shifts such as synthetic perturbations and spatial transformations. We study a class of realistic object detection settings wherein the target objects have control over their appearance. To this end, we propose a reversed Fast Gradient Sign Method (FGSM) to obtain these angelic patches *that significantly increase the detection probability, even without pre-knowledge of the perturbations. In detail, we apply the patch to each object instance simultaneously, strengthening not only classification, but also bounding box accuracy. Experiments demonstrate the efficacy of the partial-covering patch in solving the complex bounding box problem. More importantly, the performance is also transferable to different detection models even under severe affine transformations and deformable shapes. To the best of our knowledge, we are the first object detection patch that achieves both cross-model efficacy and multiple patches. We observed average accuracy improvements of* 30% *in the real-world experiments. Our code is available at:* `https://github.com/averysi224/angelic_patches`.

1. Introduction

Deep learning models have been heavily deployed in many safety-critical settings such as autonomous vehicles. However, these models have been notoriously fragile to mild perturbations. For example, natural corruptions like weather conditions and simple lightning effects can significantly degrade the performance of state-of-the-art models [11, 14]. Similarly, the performance under small spatial transformations exhibits a large gap compared to clean benchmarks [2, 7, 15]. On the other hand, a set of carefully designed adversarial examples [10] are able to manipulate the prediction behavior arbitrarily without notice of human eyes. The untrustworthiness of deep learning systems leads to high stake failures and devastating consequences.

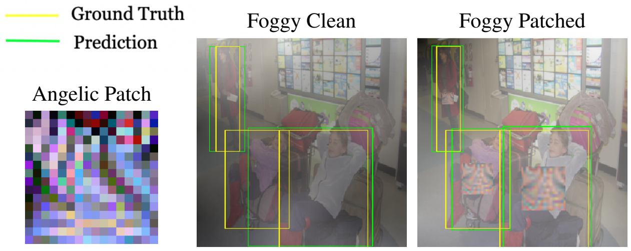
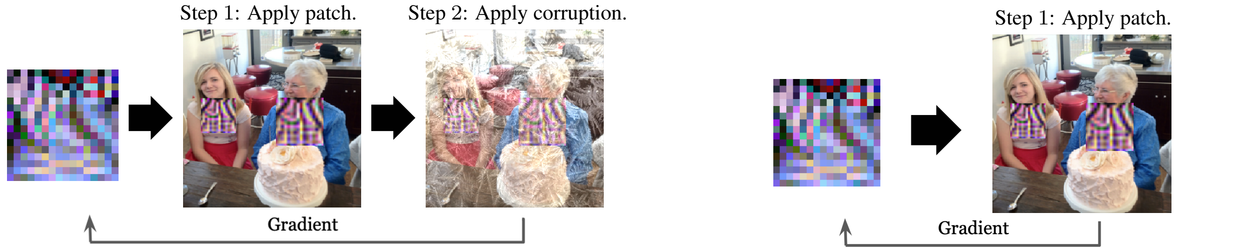


Figure 1. In this paper, we demonstrate that optimizing a partial-cover patch for pre-trained object detectors can improve robustness and significantly boost performance for both the classification and bounding box regression accuracy. In the left picture, one of the three people is mis-detected after applying the fog corruption. However, in the right image with our *angelic patch*, the detector was able to detect all three people with even more accurate bounding boxes. Consider all three people wearing an angelic raincoat, we could save lives from the foggy-blinded autonomous car!

Two main streams of algorithms investigate solving these problems. One is to improve out-of-distribution behavior by adding more robustness interventions and diverse data during training. However, they do not fully close the gap between standard model performance and perturbed results [8]. Others applied domain adaptation over covariate shift achieving reasonable performance [23], yet this method does not generalize on unseen domains. Whereas the misspecified test time distribution occurs dominantly in practice.

Motivated and inspired by the efficacy of these perturbation/adversarial methods above, we ask the question: *Can we adopt the characteristics of adversarial attack methods to help improve perturbation-robustness?* We propose to reconsider the problem setup itself and study in a scenario where the target objects are in control of their appearance. To simplify, we build the objects instead of the models to improve detection reliability. As a concrete example, consider a pedestrian interacting with autonomous cars that use deep learning models for detection. Our approach is to provide a wearable patch designed to improve the visibility of these people to these models (Figure 1). Such practices



(a) An example of corruption-aware angelic patch training procedure.

(b) An example of corruption-agnostic angelic patch training.

Figure 2. Examples of the two considered methods for constructing angelic patches. In the corruption-aware setting, we compute loss by the predictions of the corrupted patched images. At test time, we test on the same type of corruption; In the corruption-agnostic setting, we compute loss by the predictions of the corrupted clean images. At test time, we test on arbitrary corruption.

are already common for dealing with human drivers—e.g., wearing bright or reflective clothing at night.

Furthermore, though the adversarial attacks and robustness of classifiers are well-studied, we focus on the less studied yet practically broadly used object detection setting. In detail, the detectors learn not only the object class information but also learn about the localization and the size of an object, e.g. bounding boxes. Besides, the object detection problem is essentially multiple instances, causing implicit interaction between objects with even occlusions. To our knowledge, we are the first to systematically investigate this setting. We validated our method on both the single-stage detector and the two-stage detector that are with the proposal network as well as *cross model* experiments.

Our contribution is three-fold: First, we propose the novel angelic patches with a *Reversed Fast Signed Gradient Method* to improve the performance of both single-stage and two-stage third-party object detectors. Second, we demonstrate the efficacy of our framework on a wide variety of detection settings including dozens of synthetic corruptions and affine transformations without additional augmentation during training. Third, we are the first defense physical patch that achieves cross-model validation on several state-of-the-art unseen models. We extensively evaluated our approach with both programmatic patches as well as real-world experiments. We believe our approach identifies a highly practical valuable strategy that can be used in a broad range of applications.

In the following sections, we start with a review of related work, then give the proposed *angelic patch* framework and experiment results. After that, we conclude the paper.

2. Related Work

Adversarial Attacks and Patch Attacks. Our approach is based closely on techniques from the literature on *adversarial attacks*, where the goal is to design techniques to design inputs that confuse the model. While this literature initially considered attacks in the form of L_∞-norm bounded perturbations [10, 27], they have since considered more realistic perturbations such as adding noise or corruptions such as rain and snow [12]. Specifically, we build on a class of light-weighted effective attacks called *adversarial patches*. Adversarial patches confuse a classification or detection model if they are present in an image. For instance, [4, 5] propose universal physical applicable adversarial patches, misleading a classifier to output any targeted class. [25] designs glasses frames that cause facial recognition models to misclassify faces. Alternatively, [9] proposes a black-and-white mask-guided sticker to generate stickers of certain shapes for traffic sign classifiers.

Patch Methods on Object Detection. In the detection setting, most patch methods focused on adversarial attacks. [16, 20, 26] generate patches that aim to make objects invisible to detectors. Of all the works, [16, 20] float the patch arbitrarily on the images without interaction with the objects. In an application that is closely related to ours, [28] proposes to use these techniques to design wearable patches that prevent detection, which enables users to ensure their privacy. However, the achieved patch is not transferable across models. Besides, notice the misclassification of an object could lead to a misdetection, previous works gave no systematic analysis of the effect of regression loss and classification loss. Our framework, on the contrary, illustrates that the patch impacts on both ends.

Adversarial Defences. One line of promising defense strategy use data augmentation to expand the training set and cover the perturbations. The augmentations are applied either on norm-bounded perturbations [10] spatial transformations [8] or more general perturbations. In the latter direction, CutMix [32], Mixup [33], RandAugment [6], AugMix [13], and Augmax [31] strategically aggregate several general transformations to augment the training data. There are also dynamic defence [30] and structural defence [3]. Most relatedly, [24] proposed *unadversarial examples* for classification and regression settings. In contrast, our work systematically investigates the more challenging object detection setting; furthermore, we demonstrate critical features of our approach such as robustness to unseen perturbations and transferability to new object detectors, which have not been previously studied.

3. Background

3.1. Fast Gradient Sign Method (FGSM)

The fast gradient sign method [10] is an effective method for generating adversarial images, originally focusing on image classification. At a high level, this approach aims to *maximize* the loss $J(x;\theta)$ as a function of the input image x (instead of the parameters θ). A naïve strategy is to perform gradient ascent—i.e., take gradient steps $\nabla_x J(x;\theta)$. However, this approach can be inefficient.

Instead, FGSM is a *projected* update designed to maximize the step in the direction of the gradient under the L_∞ norm. This step is maximized by projecting the gradient onto the L_∞ ball $B_\infty(0,\epsilon)$, which can be efficiently computed by taking the sign of the gradient—i.e.,

$$x_{adv} = x + \epsilon \cdot \text{sign}(\nabla_x J(x,y,\theta)), \quad (1)$$

where the sign function is applied element-wise, and where the loss J is either the classification or detection loss. While the original strategy was to take a single signed gradient step, subsequent approaches also considered taking multiple steps to further improve the performance of the attack.

3.2. Patch Attack

A *patch attack* is an algorithm that constructs an *adversarial patch* p designed to mislead a specific pretrained model to incorrect predictions at higher rates. The Expectation over Transformation framework of [4] is a widely adopted framework for patch attacks. In this approach, the patch is constructed by maximizing the expected loss:

$$\hat{p} = \arg\max_{p} \mathbb{E}_{x,t,\ell} \log \Pr(\hat{y} \mid A(p,x,\ell,t)),$$

where x is an image, t is a randomly chosen transformation on the patch, ℓ is a randomly chosen location in the image, and A applies the patch p on the image x at the location ℓ by transforming the patch using t. Note that the expectation is taken over images, which encourages the identified patch to work regardless of the background. One strategy to optimize p is to use iterative FGSM updates restricted to p.

3.3. Object Detection

We consider the object detection problem, where the goal is to learn a model f that predicts, for each category $i \in \mathcal{I}$, a list of bounding boxes along with confidence associated with each bounding box that its label is i. Let x be an image, $\mathbf{b}$ be a list of ground-truth bounding boxes in x, $\hat{\mathbf{b}}$ be a list of bounding boxes predicted by f for x, where the kth bounding box of $\hat{\mathbf{b}}$ matches the k-th bounding box of $\mathbf{b}$, and $\hat{\mathbf{c}}$ be a list of confidence scores for being the category i, where the k-th confidence score of $\hat{\mathbf{c}}$ corresponds to the k-th bounding box of $\hat{\mathbf{b}}$. Then, the simplified single-shot detection (SSD) detection loss [19] is

$$J_{\text{ssd}}(x,\mathbf{b},i,f) = J_{\text{loc}}(\hat{\mathbf{b}},\mathbf{b}) + J_{\text{conf}}(\hat{\mathbf{c}},i).$$

Here, J_{loc} is the localization loss that computes the dissimilarity between the ground-truth bounding boxes $\mathbf{b}$ and the matched predicted bounding boxes $\hat{\mathbf{b}}$, and J_{conf} is confidence loss that penalizes when the confidence score for category i is small. Other detection loss functions (e.g., Faster R-CNN [22]) are defined similarly.

4. Angelic Patches

Leverage the more controlled setting Realizing the difficulty of general distribution shift robustness discussed earlier in previous sections, we leverage a novel setting where the target objects have control over their appearance. For example, pedestrians tend to wear raincoats with flamboyant colors to be noticed by the human driver. In the case of machine, this inspires us to search for lightweight patches/textures that rely on model priors instead.

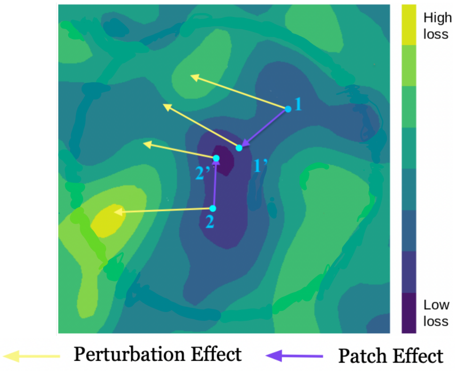


Figure 3. Perturbations on images often result in higher loss. Our angelic patches use reversed FGSM to compute a patch that, when applied to an input image, moves it to a lower point in the loss landscape.

4.1. From Patch Attack to Patch Defense

In contrast to adversarial attacks, our goal is to construct a patch p_i for each object category $i \in \mathcal{I}$ such that when p_i are present in the input image x, then the probability that instances of category i in x are correctly detected by a given model f is maximized; we refer to p_i as an *angelic patch*. Intuitively, we can attach the constructed patch p_i to an object of category i to improve the probability that it is correctly detected by f.

Our strategy for constructing angelic patches is to construct a patch that minimizes the classification and the bounding box regression loss on the training set as a function of the patch p, which is programmatically applied to images of category i. Importantly, our patch does not require modification of the given model f, instead relying solely on modifying the input image to improve performance. We focus on the whitebox setting where we have access to gradients of f—for instance, we may have reverse-engineered the model of a self-driving car, but we cannot modify the models running on other cars. Through achieving cross-model transferability, we address the problem of the practical use of possible blackbox models.

Reversed FGSM. We build on the approach of constructing adversarial patches using FGSM updates but reverse the procedure. In other words, we apply signed gradient updates to the patch to minimize the loss (rather than maximize it). This strategy generates a patch that increases the performance of the given model (see Figure 3 for an illustration). Compared to naive gradient descent methods, FGSM is effective under the scaling setup, thus tailored to this problem.

4.2. Angelic Patches for Detection

Given a detector f, our goal is to find a patch p_i for a given category i such that objects of category i in an input image x are correctly detected with a higher probability if they include p_i. Let x be an image that contains an instance for the object category i, $\mathbf{b}$ be the corresponding ground-truth bounding boxes for the category i, $\delta \in \Delta$ be perturbations, and $\mathbf{t}$ be the list of transformations for patches for ground-truth bounding boxes. We consider a patch application operator $A_{\text{det}}(p, x, \mathbf{b}, \mathbf{t}, \delta)$. Our goal is to compute the patch that minimizes a detection loss:

$$p_i = \arg\min_{p} \mathop{\mathbb{E}}_{x,\mathbf{b},\mathbf{t},\delta} J_{\text{det}}(A_{\text{det}}(p, x, \mathbf{b}, \mathbf{t}, \delta), \mathbf{b}, i, f) \qquad (2)$$

where J_{det} is the detection loss. Then, our algorithm iterates over training images x; for each one, it also iterates over objects of category i in x, samples a random transformation t and perturbation δ, and then takes a gradient step

$$p \leftarrow p - \epsilon \cdot \text{sign}\left(\nabla_p J_{\text{det}}(A_{\text{det}}(p, x, \mathbf{b}, \mathbf{t}, \delta), \mathbf{b}, i, f)\right).$$

Without loss of practical value, we simply apply the patch to the center of each object (bounding box). The pseudocode is provided in Algorithm 1.

Patch application operators. So far, we have left the patch application operators unspecified; we provide details here. Unlike previous object detector patches that arbitrarily float the patch [16, 20], we resize the patch to the object size to implicitly gain spatial information. Notice we constraint the patch to be at most one-fourth of the bounding box area to make it more practically realizable. Thus, $A_{\text{det}}(p, x, \mathbf{b}, \mathbf{t}, \delta)$ performs the following steps; (i) apply a transformation $t \in \mathbf{t}$, which is a differentiable scaling of p to final patch length

$$l_p \propto \min(\text{width}(b), \text{height}(b))$$

for each bounding box $b \in \mathbf{b}$, (ii) include p in x at each bounding box b, and (iii) apply perturbation δ to x. Note that in this case, a separate patch is added to x for each object in x with ground truth category i.

Algorithm 1 Angelic patch algorithm for object detection

Input: Object detector f, Set of images Z with bounding box annotations for category i, Learning rate ϵ, Set of patch transformations $\mathcal{T}$, and Set of perturbations Δ.
Output: Patch p_i

```
Initialize the patch $p_i$ to be zeros.
for $j \in \{1, 2, ...\}$ do
    for $(x, \mathbf{b}) \in Z$ do
        Choose $\mathbf{t}$ from $\mathcal{T}$.
        Randomly choose $\delta$ from $\Delta$
        $x' \leftarrow A_{\text{det}}(p, x, \mathbf{b}, \mathbf{t}, \delta)$        ▷ apply patches
        $g_p \leftarrow \nabla_p J_{\text{det}}(x', \mathbf{b}, i, f)$
        $p_i \leftarrow p_i - \epsilon \cdot \text{sign}(g_p)$        ▷ reversed FGSM
    end for
end for
return $p_i$
```

Corruption-aware vs Corruption-agnostic. We first verify the efficacy of our method through a corruption-aware setting, where we train and test with the same type of corruption applied (say frost). We then leverage a complex setting, in which we train with no corruption applied. In this way, neither model nor the patch is aware of the test corruption during training. We demonstrate the two training procedures in Figure 2.

Multiple objects. Previous adversarial patch algorithms [5, 20] focus on optimizing the patch for a single object in each image at each gradient step. However, in detection, there may be multiple patches; as a consequence, patches may be occluded by other patches.

We handle overlapping patches by backpropagating through the patch application function across all patches, ensuring that only the visible portion of each patch is updated by each gradient step.

4.3. Cross Model Patch Training

A batch of detector patch attack works investigate the transferability across different models [16, 20, 26] in a naive floating patch setup. In a similar and more realistic setting, [28] designed wearable adversarial patches and stated that it was impossible to achieve cross-model transferability. Intuitively, we know the transferability of attack methods should be easier to achieve as the break of either the classification head or the regression head could cause a misdetection. Yet for the patch defense setting, we must improve both heads to improve the overall performance.

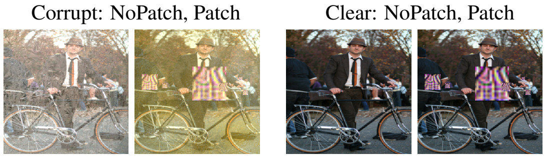


Figure 4. Example images of two comparison groups to evaluate our efficacy of the angelic patch. The left two images are for corrupted testing scenarios; the right two images denote the no corruption (e.g. clear weather) testing scenarios, both for whether the objects wear a patch. Notice that most of the corruptions in ImageNet-C are random corruptions with different appearances even for a single corruption category. However, we validate our patch from the average performance of the large testset.

Avg. Precision	IoU			Area		
	0.5:0.95	0.5	0.75	S	M	L
Corrupt, NoPatch	14.0	24.3	22.1	0.6	12.3	26.8
Corrupt, Patch	**22.0**	**39.9**	**34.1**	**2.24**	**15.6**	**35.9**
Avg. Recall	#Dets			Area		
	1	10	100	S	M	L
Corrupt, NoPatch	8.9	23.6	27.0	2.1	19.3	37.3
Corrupt, Patch	**17.6**	**37.3**	**41.1**	**9.9**	**32.0**	**54.3**

Table 1. mAP and mAR comparison results for corruption-aware Faster R-CNN patches over seven categories. For the category-wise table, please refer to the supplementary. Results show our patch improves both mAP and mAR in the corrupted setting.

To this end, we propose to evaluate our method in both the common single model transferability and a novel efficient double model transferability. Specifically, when we train a patch on one pretrained detector f_1, then apply the patch to another detector f_2, the performance improvement is inconsistent on different object categories. In this way, we constructed the double model transferability. That is, we alternatingly apply signed gradients of two different detectors to update a single patch. Thus, we obtain a single patch that is robust to both detectors, encouraging our patch to generalize to new detectors.

5. Experiments

We empirically justify the efficacy of the proposed angelic patches in both the corruption-aware and corruption-agnostic settings on two detectors, Faster R-CNN [22] and SSD [19], over the MS-COCO dataset [18] (Section 5.1 and 5.2). Then, we evaluate our patch for spatial robustness and cross-model transferability (Section 5.3 and 5.4). Finally, we *printed out* the angelic patches to a set of various real-world experiments to demonstrate the practical value of angelic patches (Section 5.6). For each experiment, we provide statistics and sample visualization images. For full results, please refer to the supplementary.

Experimental Setup. We consider both the one-stage and two-stage detectors: Faster R-CNN [22] and SSD [19], respectively. We apply patches to objects in the MS-COCO dataset [18]. As we will consider physical patches applied to real-world objects, we focus on seven categories that are easy to get in the real world: "bus", "cup", "person", "bottle", "bowl", "laptop", and "chair".

Baselines. We consider two groups of comparison setups: ("Corrupt, NoPatch"), ("Corrupt, Patch"); and ("Clear, NoPatch"), ("Clear, Patch"), see Figure 4. Here "Patch" setups are our results. For more details about the experiment setups, please refer to Appendix A.

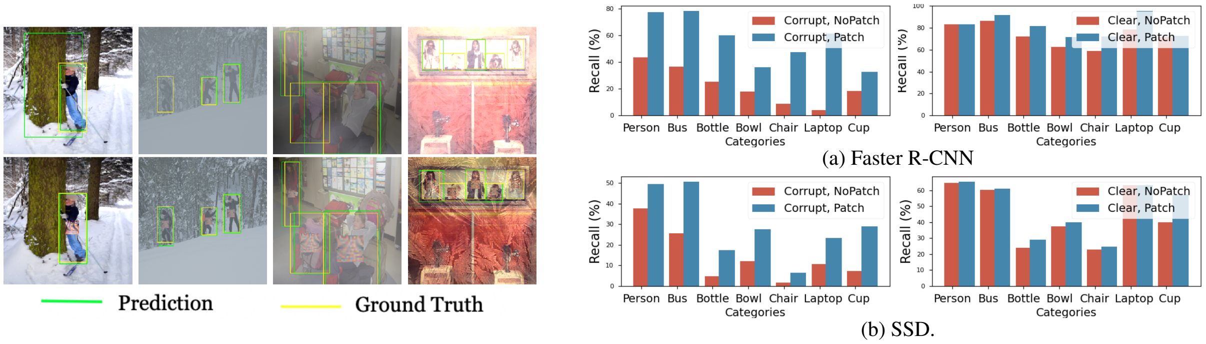
Metrics. For evaluation, we consider two sets of metrics. First, we use the standard set of COCO object detection evaluation metrics [18], consisting of the average precision (AP) at the different intersection of union (IoU) thresholds (i.e., $0.5 : 0.05 : 0.95$), and the average recall (AR) metrics for each category. Second, we use the AR at IoU 0.5 for high confidence (i.e., confidence > 0.5) predictions.

5.1. Corruption-aware Angelic Patches

For the perturbation set Δ, here we consider frost of severity level 3 from ImageNet-C [11] for our corruption-aware training and testing. For a set of patch transformations $\mathcal{T}$, we use zooming in/out to mimic the varied viewpoints we expect for real-world patches. Intuitively, corruption-aware patches achieve good performance more easily since they are aware of the corruption distribution in identifying patches. The mAP and mAR results for Faster R-CNN are shown in Table 1.

For each object category, the primary average precision metrics ($0.5 : 0.05 : 0.95$) of patched images improve by a large margin. In particular, this improvement corresponds to fewer false positive detections, indicating more precise bounding box locations on patched images. On the other hand, the impressive improvements in average recall demonstrate that our patch helps the detector miss fewer ground truth detections. Detection samples in Figure 5 demonstrate how our corruption-aware angelic patches help improve detection performance under corrupted images.

In addition, we show the easier-to-understand valid detection accuracy in Figure 5a and Figure 5b using our second metric. Our patch increases performance by large margins in each category, both with and without corruption. We see as much as a two to three times improvement in accuracy in some categories. In particular, we find that for corruptions that cause greater degradation in performance, our patch provides the largest gain in performance. Thus, our patches work well at recovering performance in settings where shifts and corruptions reduce performance. In the next section, we consider a more challenging setting where corruptions are unseen in patch construction.



(a) Faster R-CNN

(b) SSD.

Figure 5. We show corruption-aware AR (high IoU high confidence) of the two detectors in both the corrupted and the clear setting (no patch results, patch results). Results show drastic improvements on patched images in both detectors.

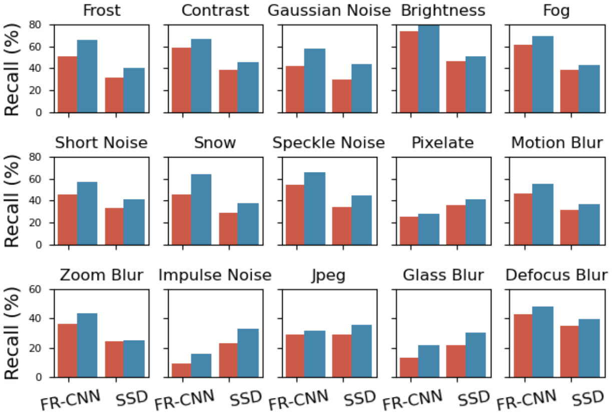


Figure 6. High IoU high confidence recall for corruption-agnostic Faster R-CNN and SSD patch under corruption on COCO. For each color, the first bar in red is the performance on original images, the second bar in blue is the performance on patched images.

5.2. Corruption-agnostic Angelic Patches

We now leverage our approach in the more challenging corruption-agnostic setting, where the goal is to improve performance without prior knowledge of corruption. For corruption-agnostic experiments, we broadly consider 15 different corruptions including weather corruptions like frost and fog and lighting corruptions like brightness and contrast. We applied the same set of patch transformations $\mathcal{T}$ as in the corruption-aware setting. In this case, we trained patches without any corruption. Then, we applied corruptions to the patched and clean image during testing.

As before, we show the high IoU high confidence recall for both detectors under 15 corruption types in Figure 6. We observe reasonable increases for both detectors in the majority of the corruption categories. In Table 2, we show key statistics of mean average precision (AP) and mean average recall (AR) across all seven categories of frost corruption in comparison with the corruption-aware results in Table 1. Our results demonstrate that despite no prior knowledge of the corruption during training, the corruption-agnostic patch is still effective at improving performance in the presence of corruption applied at test time. The agnostic patch achieved fewer improvements on single corruption (frost) when compared to the aware setting. The results agree with our intuition that exposing the corruption distribution during training helps with a performant patch.

Avg. Precision	IoU			Area		
	0.5:0.95	0.5	0.75	S	M	L
Corrupt, NoPatch	14.3	24.9	20.9	1.2	10.4	26.5
Corrupt, Patch	**17.8**	**33.1**	**27.2**	**3.3**	**12.3**	**31.8**
Avg. Recall	#Dets			Area		
	1	10	100	S	M	L
Corrupt, NoPatch	8.7	23.4	26.6	6.2	17.2	37.0
Corrupt, Patch	**14.4**	**32.8**	**37.9**	**9.6**	**25.3**	**49.6**

Table 2. mAP and mAR results for corruption-agnostic Faster R-CNN patch on the seven selected categories in the corrupted COCO dataset. For each corruption, the first row is the performance on the original corrupted images, the second row is the performance on patched images.

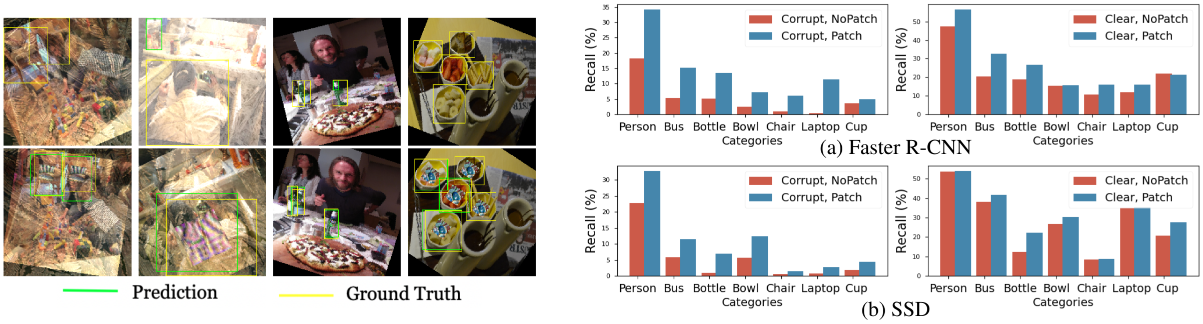
We also provide the training curves of both settings in Figure 8 where both losses improved to lower than on the corresponding no-patch images.

5.3. Spatial Transformation

We then move on to a more realistic setting, in which we apply random affine transformations on the whole patched image. We show detection accuracies in clear and corrupted settings for both detectors in Figure 7. Example images are also provided on the left. We observe that despite no explicit spatial transformation augmentation applied during training, our patch still improves the spatial robustness of both object detectors even more significantly than no transformation testing. This enhances our confidence in applying the patch to the real world.

5.4. Transferability

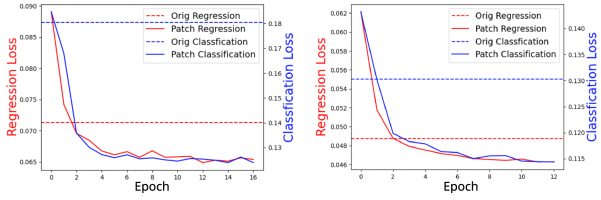
Imagine the real-world scenario that our pedestrian wears an angelic raincoat, there could be autonomous cars from different companies driving on the road. The companies may deploy different perception models for system



(a) Faster R-CNN

(b) SSD

Figure 7. Spatial transformation results for both detectors. We provide corrupted (left) and clear (right) results.



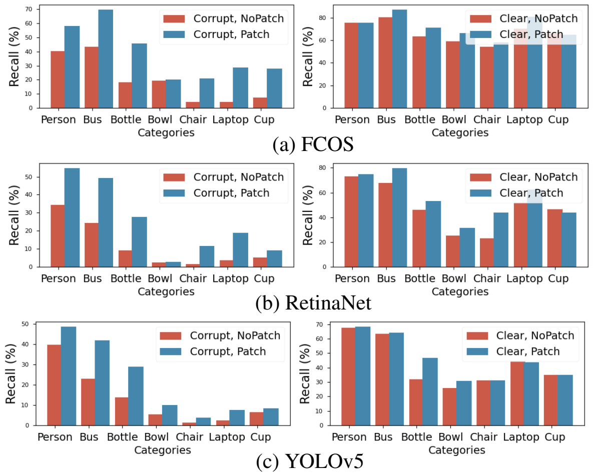
(a) Faster R-CNN aware.

(b) Faster R-CNN agnostic.

Figure 8. Sampled training curves for classification loss and regression loss on the person category.

consideration. In that case, the feature of cross-model transferability on both the known and unknown detector models would boost our protection for pedestrians.

As stated in a prior work [28], we observed cross-model improvements on only some of the categories when trained on a single detector and tested on the other. However, we pushed the experiment to a double-model setting, where we achieve transferability on all target object categories. In this case, we update the patch with two pretrained detectors during training by a simple iterative reversed FGSM procedure. Here we train our patch with SSD and Faster-RCNN, and evaluate its transferability with two object detectors from torchvision [21] including FCOS [29], RetinaNet [17] and the well-known YOLOv5 [1]. We show high confidence (> 0.5) recall and precision results in Figure 9. We see drastic performance improvements for all three unseen models. We also provide the performance of single detector transferability results in supplementary.



(a) FCOS

(b) RetinaNet

(c) YOLOv5

Figure 9. Cross model results.

5.5. Extra Results

Considering that it could be hard to control all objects in real deployments, we evaluate under two less satisfying scenarios. The first scenario is when not all objects wear the patch, and the second scenario is when the patch is not placed at the center of the objects. High IoU and high confidence recall results are shown in Figure 11 and Figure 12, respectively. Again, our patch gains an advantage under both cases.

5.6. Physical Patch Experiments

Finally, we demonstrate the critical property that our patches continue to work well when applied physically in new real-world scenes. To this end, we printed out our patches and attached them to real-world objects to verify that our patch continues to work well. We collected videos in four different environments from different angles for each category and converted each video to approximately 100 image frames. We controlled the target object moving trajectory to ensure the same object videos behave as similarly as possible before and after attaching the patches.

We show results in Table 3; we observed average accuracy improvements of 30% with the patch applied. To visualize the patch performance, we show example images in Figure 10. Following our perspective experiments in the previous section, we tested not only front-facing patches but also unseen viewpoints such as extremely skewed patches, e.g., on cups and bottles. We also tested our patches on deformable objects such as laptops in extreme poses. We tested under different poses for the "person" category, including jumping jacks, partial occlusion, sitting, etc. Our patches performed well across all viewpoints and poses.

6. Conclusion

We have proposed angelic patches, a promising strategy enabling users to improve their detection probability on third-party detectors. Our extensive experimental re-

Category	Env	Faster R-CNN				SSD			
		Corrupt, NoPatch	Corrupt, Patch	Clear, NoPatch	Clear, Patch	Corrupt, NoPatch	Corrupt, Patch	Clear, NoPatch	Clear, Patch
Cup	Env01	0.0%	**41.2%**	100.0%	100.0%	0.0%	**12.0%**	100.0%	100.0%
	Env02	5.3%	**42.1%**	100.0%	89.5%	4.0%	**12.0%**	68.4%	**73.7%**
	Env03	14.3%	**35.7%**	96.4%	**100.0%**	0.0%	**28.0%**	100.0%	100.0%
	Env04	0.0%	**23.8%**	90.5%	**100.0%**	13.4%	**30.0%**	100.0%	100.0%
Person	Env01	100.0%	100.0%	100%	**100%**	77.8%	**100.0%**	100.0%	**100.0%**
	Env02	25.0%	**35.4%**	81.3%	**85.4%**	0.0%	**8.5%**	31.2%	**46.8%**
	Env03	29.4%	**62.7%**	85.7%	85.7%	27.9%	**60.5%**	76.7%	**88.4%**
	Env04	74.0%	**100.0%**	100.0%	100.0%	36.0%	**56.0%**	**68.0%**	64.0%
Bottle	Env01	15.7%	**73.7%**	100.0%	100.0%	0.0%	**56.3%**	50.0%	**88.9%**
	Env02	31.6%	**84.2%**	100.0%	100.0 %	0.0%	**51.1%**	86.4%	**95.5%**
	Env03	17.6%	**47.1%**	100.0%	100.0%	26.3%	**78.9%**	57.9%	**78.9%**
	Env04	75.0%	**100.0%**	100.0 %	**100.0%**	4.3%	**17.4%**	66.2%	**69.6%**
Laptop	Env01	38.1%	**66.7%**	100.0%	100.0%	12.7%	**45.0%**	100%	100%
	Env02	0.0%	**19.2%**	96.2%	96.2%	23.3%	**72.3%**	93.8%	**100%**
	Env03	0.0%	**41.2%**	100.0%	100.0%	0.0%	**51.3%**	100%	100%
	Env04	0.0%	**19.1%**	100.0%	100.0 %	6.7%	**27.8%**	100%	100%

Table 3. Real-world high IoU high confidence accuracy w/w.o corruption-aware patch in varied scenes under frost corruptions.

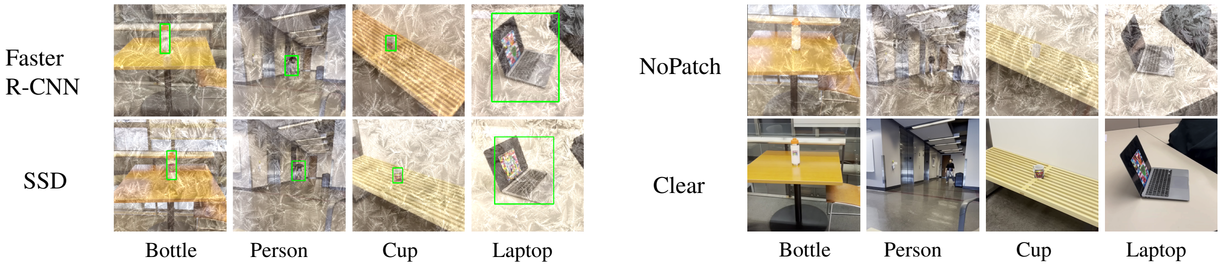
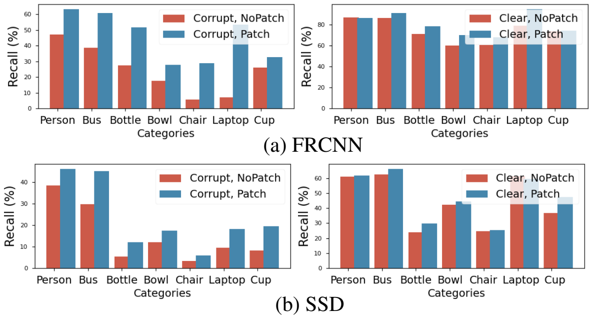


Figure 10. Example Real-world predictions under frost corruptions and clear/no patch images for comparison. The objects with patches (on the left) had a much higher chance of being detected under the frost corruption.

sults demonstrate that our approach can significantly improve detection probability in both the corruption-aware and corruption-agnostic settings, including in the physical patch setting representative of many use cases in safety-critical settings. We believe our patches can be used to make many important objects more visible to third-party detectors—e.g., in the autonomous car setting: pedestrians, other vehicles, signs, traffic cones, and barriers, among others. Most importantly, we designed patches that transfer well across multiple detectors.



(a) FRCNN

(b) SSD

Figure 11. High IoU high confidence results when patches are not applied on all the object instances (partial).

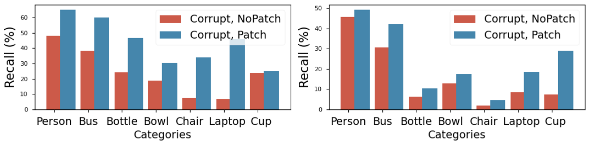


Figure 12. High IoU high confidence results when patches are not applied in the center of object instances (random placed in bbox).

Future work is needed to understand whether our patches work well for natural covariant shifts. In addition, while we have demonstrated that our approach does not reduce performance on average (e.g., of other objects in the scene), it is important to validate that our patches do not interfere with detecting other objects (e.g., unpatched pedestrians) in more realistic scenarios.

Acknowledgements This work was supported in part by ARO W911NF-20-1-0080. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Army Research Office (ARO), or the Department of Defense, or the United States Government.

References

[1] GitHub - ultralytics/yolov5: YOLOv5 in PyTorch ONNX CoreML TFLite — github.com. https://github.com/ultralytics/yolov5. [Accessed 03-Nov-2022]. 7

[2] Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4845–4854, 2019. 1

[3] Motasem Alfarra, Juan C Pérez, Ali Thabet, Adel Bibi, Philip HS Torr, and Bernard Ghanem. Combating adversaries with anti-adversaries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5992–6000, 2022. 2

[4] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018. 2, 3

[5] Tom Brown, Dandelion Mane, Aurko Roy, Martin Abadi, and Justin Gilmer. Adversarial patch. 2017. 2, 4

[6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 2

[7] Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)*, pages 1–7. IEEE, 2017. 1

[8] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International conference on machine learning*, pages 1802–1811. PMLR, 2019. 1, 2

[9] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018. 2

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2, 3

[11] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018. 1, 5

[12] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, 32, 2019. 2

[13] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019. 2

[14] Hossein Hosseini, Baicen Xiao, and Radha Poovendran. Google's cloud vision api is not robust to noise. pages 101–105, 12 2017. 1

[15] Can Kanbak, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Geometric robustness of deep networks: analysis and improvement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4441–4449, 2018. 1

[16] Mark Lee and Zico Kolter. On physical adversarial patches for object detection. *arXiv preprint arXiv:1906.11897*, 2019. 2, 4

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 7

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*. Springer International Publishing, 2014. 5

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3, 5

[20] 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. 2, 4

[21] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 7

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 3, 5

[23] Evgenia Rusak, Steffen Schneider, Peter Gehler, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Adapting imagenet-scale models to complex distribution shifts with self-learning. *arXiv preprint arXiv:2104.12928*, 2021. 1

[24] Hadi Salman, Andrew Ilyas, Logan Engstrom, Sai Vemprala, Aleksander Madry, and Ashish Kapoor. Unadversarial examples: Designing objects for robust vision. *Advances in Neural Information Processing Systems*, 34:15270–15284, 2021. 2

[25] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016. 2

[26] Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramer, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX workshop on offensive technologies (WOOT 18)*, 2018. 2, 4

[27] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 2

[28] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 2, 4, 7

[29] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 7

[30] Dequan Wang, An Ju, Evan Shelhamer, David Wagner, and Trevor Darrell. Fighting gradients with gradients: Dynamic defenses against adversarial attacks. *arXiv preprint arXiv:2105.08714*, 2021. 2

[31] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. *Advances in neural information processing systems*, 34:237–250, 2021. 2

[32] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 2

[33] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2