
Natural Adversarial Objects

Felix Lau Scale AI felix.lau@scale.com	Sasha Harrison Scale AI sasha.harrison@scale.com	Nishant Subramani Intel Intelligent Systems Lab nishant.subramani23@gmail.com
Aerin Kim Scale AI aerin.kim@scale.com	Elliot Branson Scale AI elliott.branson@scale.com	Rosanne Liu ML Collective rosanne@mlcollective.org

Abstract

1 Although state-of-the-art object detection methods have shown compelling perfor-
2 mance, models often are not robust to adversarial attacks and out-of-distribution
3 data. We introduce a new dataset, Natural Adversarial Objects (NAO), to evaluate
4 the robustness of object detection models. NAO contains 7,936 images and 13,604
5 objects that are unmodified, but cause state-of-the-art detection models to misclas-
6 sify with high confidence. The mean average precision (mAP) of EfficientDet-D7
7 drops 68.3% when evaluated on NAO compared to the standard MSCOCO valida-
8 tion set. We investigate why examples in NAO are difficult to detect and classify.
9 Experiments of shuffling image patches reveal that models are overly sensitive to
10 local texture. Additionally, using integrated gradients and background replacement,
11 we find that the detection model is reliant on pixel information within the bounding
12 box, and insensitive to the background context when predicting class labels.

13 1 Introduction

14 It is no longer surprising to have machine learning vision models perform well on large scale
15 training sets and also generalize on canonical test sets coming from the same distribution. However,
16 generalization towards difficult, out-of-distribution samples still poses difficulty. Recht et al. [16]
17 showed that model performance on canonical test sets is an overestimate of how they will perform
18 on new data. Moreover, recent research on adversarial attacks has shown that deep neural networks
19 are surprisingly vulnerable to artificially manipulated images, casting new doubt on the efficacy and
20 security of such models.

21 The vulnerability of neural networks to adversarial attacks that are deliberately generated to fool the
22 system is unsurprising, and well studied. However, this type of attack represents a narrow threat
23 model because it necessitates that the adversary has control over the raw input, or has access to
24 the model weights. It is often overlooked that real-world, unmodified images can also be used
25 adversarially to cause models to fail. These “natural” adversarial attacks represent a less restricted
26 threat model, where an attacker can easily create black-box attacks without carefully constructing
27 input perturbations [5], but only by using naturally occurring images that are easily obtainable. Such
28 images are called natural adversarial examples [7]: unmodified, real-world images that cause modern
29 image classification models to make egregious, high-confidence errors.

30 In [7] natural adversarial examples are only constructed for image classification models. In this work,
31 we seek to create an evaluation set analogous to [7], but instead targeted at object detection tasks. We
32 name such a dataset Natural Adversarial Objects (NAO). The goal of NAO is to benchmark the worst
33 case performance of state-of-the-art object detection models, while requiring that examples included
34 in the benchmark are unmodified and naturally occurring in the real world.

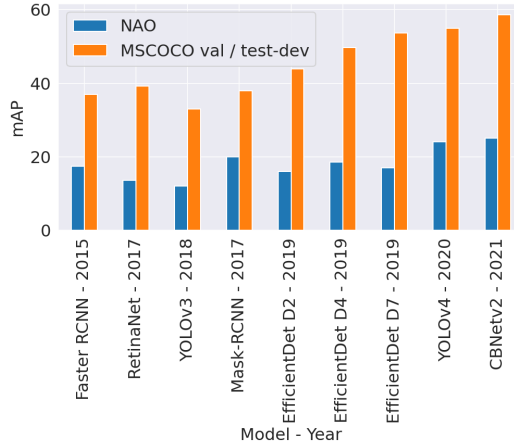


Figure 1: Mean average precision (mAP) of various detection models evaluated on NAO and MSCOCO *val* or *test-dev* set. All models show significant reduction in performance on NAO despite their accuracy improvement in MSCOCO in recent years. NAO is a challenging test set for detection models trained on MSCOCO and future work is required to close the performance gap.

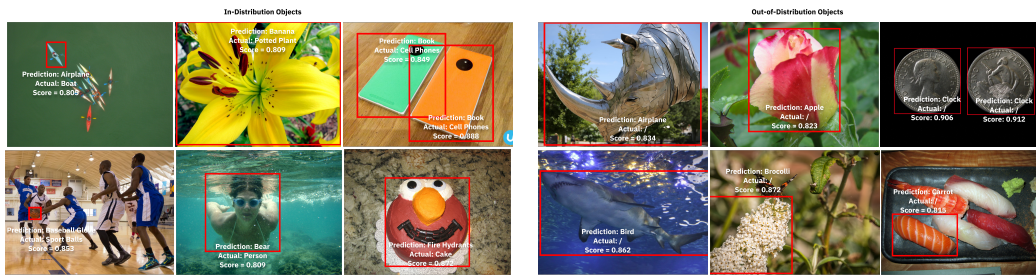


Figure 2: Sample images from NAO where EfficientDet-D7 produces high confidence false positives and egregious classification. **Left:** High confidence misclassified objects where the ground truth label is in-distribution and among the MSCOCO object categories. **Right:** High confidence false positives where the ground truth object is out-of-distribution (i.e. not part of MSCOCO object categories). The misclassified objects and false positives are superficially similar to the predicted classes – for example, the fin of the shark is visually similar to the airplane tail and the yellow petals of the flower is similar to a bunch of bananas.

35 We present a method to identify natural adversarial objects using a combination of existing object
 36 detection models and human annotators. First, we compare the predictions from various off-the-shelf
 37 detection models against a dataset already annotated with ground truth bounding boxes. We consider
 38 images containing high confidence false positives and misclassified objects as candidates for NAO.
 39 Then, we use a human annotation pipeline to filter out mislabeled images and non-obvious objects
 40 (e.g. occluded or blurry objects). Finally, we re-annotate the images using the object categories of the
 41 Microsoft Common Objects in Context (MSCOCO) dataset [13].

42 We perform extensive analyses to understand why objects in NAO are naturally adversarial. We
 43 visualize the embedding space common to MSCOCO, OpenImages, and NAO, and show that
 44 NAO images exist in the "blind spots" of the MSCOCO dataset. Next, by comparing integrated
 45 gradients [23] with predicted bounding boxes and replacing object backgrounds, we show that the
 46 detection model seldom makes use of object contexts. Lastly, by shuffling patches within the bounding
 47 box, we show that models relies on object subparts and texture to detect and classify the objects.

48 NAO ground truth bounding box annotations are made available under CC-BY 4.0.

49 2 Related Work

50 **Natural Adversarial Examples** Hendrycks et al. [7] construct two datasets, namely *ImageNet-A*
51 and *ImageNet-O*, to measure the robustness of image classifiers against out of distribution examples.
52 To construct these two datasets, they choose images on which a pretrained ResNet model failed to
53 make a correct prediction. We adopt a similar approach for selecting adversarial examples but use
54 an object detection model and take extra steps to ensure high quality annotations by using human
55 annotators. Zhao et al. [27] develops a method to generate adversarial perturbation that lies on the
56 data manifold where the perturbation is meaningful to the semantic of the images.

57 **Adversarial Examples** Adversarial examples are inputs that are specifically designed to cause the
58 target model to produce erroneous outputs. Arpit et al. [1] analyzed the capacity of neural networks
59 to memorize training data, and found that models with a high degree of memorization are more
60 vulnerable to adversarial examples. Jo and Bengio [9] have shown that convolutional neural networks
61 tend to learn the statistical regularities in the training dataset, rather than the high level abstract
62 concepts. Since adversarial examples are transferable between models that are trained on the same
63 dataset, these different models may have learned the same statistics and therefore are vulnerable to
64 similar adversarial attacks. Brendel and Bethge [3] show that small local image features are sufficient
65 for deep learning model to achieve high accuracy. Geirhos et al. [4] show that ImageNet-trained
66 CNNs are biased toward texture and created *Stylized-ImageNet* to reveal the severity of such bias.
67 Similarly, Ilyas et al. [8] showed that adversarial examples are a byproduct of exploiting non-robust
68 features that exist in a dataset. Non-robust features are derived from patterns in the data distribution
69 that are highly predictive, yet brittle and incomprehensible to humans. Undoubtedly, the reasons
70 behind the existence and pervasiveness of adversarial examples still remains an open research problem.
71 Zhang and Wang [26] developed a method to improve the robustness of object detection model by
72 identifying an asymmetric role of task losses.

73 **Model Interpretability** While the interpretability of deep neural networks remains an open re-
74 search question, there exist attribution methods that help explain the relationships between the input
75 and output of such models. Sundararajan et al. [23] suggests that attribution methods should satisfy
76 two axioms: sensitivity and implementation invariance, and proposes a new method, *Integrated*
77 *Gradient*, to understand which parts of an image influence the prediction the most.

78 **Object Detection Architectures** Detection models fall into two categories: one-stage
79 ([25], [18], [17]) and two-stage models ([20], [22]), differentiated by whether the model has a
80 region pooling stage. Single-stage model are more computationally efficient, but usually less accurate
81 than the 2-stage models. In this paper, we evaluate both single and two-stage models using the NAO
82 dataset. Tan et al. [25] introduced EfficientDet, which uses EfficientNet [24] as backbone and uses
83 BiFPN such that the model is more efficient while more accurate, achieving state-of-the-art results in
84 MSCOCO at 54.4 on the *val* set.

85 3 Creating Natural Adversarial Objects (NAO) Dataset

86 3.1 Limitations of MSCOCO

87 MSCOCO [13] is a common benchmark dataset for object detection models. It contains 118,287
88 images in the training set, 5,000 images in the *val* set and 20,288 images in the *test-dev* set. MSCOCO
89 contains 80 object categories consisting of common objects such as *horse*, *clock*, and *car*. The goal
90 of MSCOCO is to introduce a large-scale dataset that contains objects in non-iconic or non-canonical
91 views. The images in MSCOCO were originally sourced from Flickr, then filtered down in order to
92 limit the scope of the benchmark to a set of 80 categories. These 80 categories were chosen from a
93 list of the most commonplace visually identifiable objects. Still, this category list represents only
94 a small subset of object categories in real life. For example, 'fish' is not among the 80 categories,
95 and as a result there are only a few photos taken underwater. This leads to a biased benchmark with
96 limitations for generalizability and robustness. As a result, in this work, we ensure more diverse
97 sourcing — choosing images from OpenImages v6 [11], a dataset with 600 object categories, in order
98 to create a more representative dataset.

	Statistics		Top 3 Objects (Count)		
	# of Images	# of Objects	1st	2nd	3rd
MSCOCO <i>val</i>	5,000	36,781	Person (11,004)	Car (1,932)	Chair (1,791)
MSCOCO <i>test-dev</i>	20,288	-	-	-	-
NAO	7,936	13,604	Person (4,693)	Cup (2,257)	Car (752)

Table 1: Dataset statistics of MSCOCO *val*, *test-dev* and NAO.

99 **3.2 Sourcing Images for NAO from OpenImages**

100 To create NAO, we first sourced images from the training set of OpenImages [11], a large, annotated
 101 image dataset containing approximately 1.9 million images and 15.8 million bounding boxes across
 102 600 object classes.

103 One challenge of using OpenImages is that the bounding boxes are not exhaustively annotated. Each
 104 image is first annotated with positive and negative labels which indicate the presence or absence of
 105 an object in the image. Only objects belonging to the positive label categories are annotated with
 106 bounding boxes. As a result, some objects that belong to the OpenImages object categories are not
 107 labeled with a bounding box. For example, imagine both horse and pig are represented in the 600
 108 object classes. If an image contains a horse and a pig, and only the category of horse is included
 109 in preliminary round of positive labels, then the image would be labeled with a bounding box for
 110 the horse but not the pig. This non-exhaustive annotation approach makes it difficult to produce
 111 and compare precision and recall to other exhaustively annotated dataset such as MSCOCO. This
 112 is because false positives and false negatives can only be evaluated accurately if the ground truth
 113 bounding boxes are exhaustive.

114 One other challenge that arises when sourcing images from OpenImages is that the object categories
 115 of OpenImages and MSCOCO are not the same. Therefore, after obtaining a set of natural adversarial
 116 images, we exhaustively annotate the images with all 80 MSCOCO object classes to facilitate
 117 straightforward comparison between NAO and the MSCOCO *val* and *test* sets.

118 **3.3 Candidate Generation**

119 To generate object candidates, we perform inference on OpenImages using an EfficientDet-D7
 120 model [25] pretrained on MSCOCO, which yields predicted object bounding boxes for each candidate
 121 image. Our goal is to find two types of errors: (i) **hard false positives** (i.e. false positives with high
 122 confidence) and (ii) **egregiously misclassified** objects. For a detection to be a hard false positive, we
 123 require the prediction to have no matching ground truth box with intersection over union (IoU) greater
 124 than 0.5, and class confidence score greater than 0.8. We define egregiously misclassified objects as
 125 predictions that have a matching ground truth bounding box with an IoU greater than 0.5, but have an
 126 incorrect classification with a confidence greater than 0.8. We do not consider false negatives with
 127 high confidence because we observe that these are commonplace especially in crowded scenes. There
 128 are 43,860 images containing at least one hard false positive or egregiously misclassified object.

129 **3.4 Annotation Process**

130 Our annotation process has two annotation stages: classification and bounding box annotation.

131 **Classification Stage** In the classification stage, annotators identify whether the object described
 132 by the bounding box shown indeed belongs to the ground truth class as defined by the annotation
 133 in OpenImages or as predicted by the EfficientDet-D7. The purpose of this stage is to remove the
 134 possibility that the model prediction is "incorrect" due to the ground truth label being incorrect.

135 In addition, we ask the annotators to confirm whether the object can be "obviously classified"
 136 according to the following criteria:

- 137 1. Is the bounding box around the object correctly sized and positioned such that it is not too
 138 big or too small?
- 139 2. Does the object appear blurry?

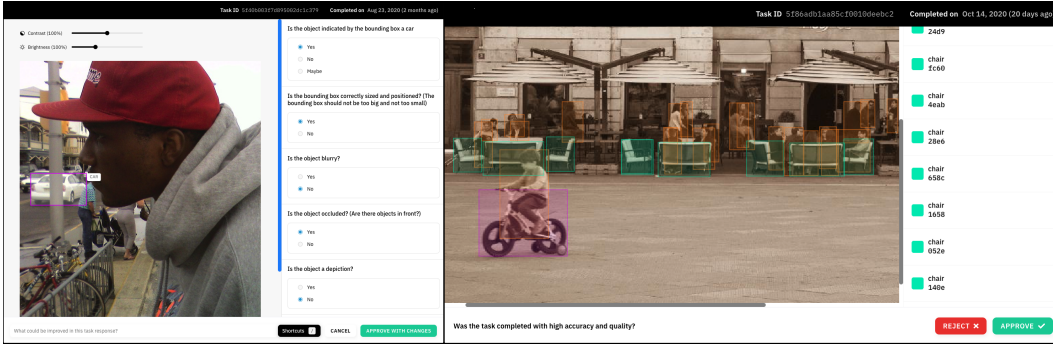


Figure 3: **Left:** Annotation interface for the first annotation stage (classification) where the annotator confirms that the object belongs to the correct category, not occluded, not blurry and not a depiction. **Right:** Annotation interface for second annotation stage (bounding box) where the annotators locate and classify all objects in the images using the MSCOCO object categories.

- 140 3. Is the object occluded (i.e. are there other objects in front of this one)?
 141 4. Is the object a depiction of the correct class (such as a drawing or an image on a billboard)?

142 We ask these additional questions to filter out ambiguous objects, such that a human can easily
 143 identify what class an object belongs to. After this filtering, 18.1% of the images (7,936) remain; each
 144 of the remaining images are confirmed to fulfill the 4 criteria, and represent true misclassifications by
 145 the model. In this first annotation stage (classification), 5 different annotators are asked to annotate
 146 the same image and we use their consensus to produce an aggregated response by majority vote.

147 **Bounding Box Stage** In the second annotation stage (bounding box), annotators exhaustively
 148 identify and put boxes around all objects that belong to the MSCOCO object categories. We are
 149 unable to directly use the annotations from OpenImages because there is not a one-to-one mapping
 150 between the OpenImages and MSCOCO object categories, and because the bounding box annotations
 151 from OpenImages are not exhaustively annotated. However, the bounding box annotations from
 152 OpenImages are provided to the annotators as a starting point.

153 These bounding box annotation tasks are completed by 2 sets of annotators. The first set of annotators
 154 complete the bulk of the task by placing bounding boxes around objects that belong to the MSCOCO
 155 object categories. The second set of annotators review the work of the first set of annotators,
 156 sometimes adding missing bounding boxes or editing the existing ones.

157 To ensure the quality of the annotation is high, in both of these stages, the annotators have to pass
 158 multiple quizzes before they can start working tasks to ensure they understand the instructions well.
 159 If the annotator fails to maintain a good score, they are no longer eligible to continue to annotate
 160 the images. This process of vetting annotators is consistent with the methodology used to construct
 161 MSCOCO [13].

162 When the annotators from the 2 different stages disagree, we tie break by choosing second annotator
 163 who is positioned as the reviewer.

164 3.5 Evaluation Protocol

165 The goal of NAO is to test the robustness of object detection models against edge cases and out-
 166 of-distribution images. We propose two main evaluation metrics: **overall mAP** and **mAP without**
 167 **out-of-distribution objects**. mAP without out-of-distribution objects evaluates against edge cases of
 168 object categories that the detection models are trained on, while the overall mAP evaluates robustness
 169 against out-of-distribution objects. For calculating mAP without out-of-distribution objects, any
 170 detection matched to an object not belong to the 80 MSCOCO object categories is not considered a
 171 false positive.

172 NAO should be mainly used as a test set to evaluate detection models trained on MSCOCO. However,
 173 a split of train, validation and test set is also provided for robustness approaches that require training.

	Params	MSCOCO <i>val</i>	MSCOCO <i>test-dev</i>	NAO		
		mAP	mAP	mAP	mAR	mAP w/o OOD
Faster RCNN [19]	42M	21.2	21.5	17.4	48.9	28.2
RetinaNet-R50 [14]	34M	39.2	39.2	13.7	41.1	22.8
YOLOv3 [18]	62M	-	33.0	12.1	30.4	19.6
Mask RCNN R50 [6]	44M	37.9	-	20.0	51.4	30.6
EfficientDet-D2 [25]	8.1M	43.5	43.9	16.1	46.1	28.6
EfficientDet-D4 [25]	21M	49.3	49.7	18.6	50.0	34.3
EfficientDet-D7 [25]	52M	53.4	53.7	17.0	46.3	30.6
YOLOv4-P7 [2]	28.7M	55.3	55.5	24.1	62.0	41.6
CBNetv2-HTC [12]	231M	58.2	58.6	25.1	61.5	43.2

Table 2: mAP of various detection models evaluated on MSCOCO *val* and *test-dev* set and NAO. Accuracy of all models were significantly lower on NAO than on MSCOCO. There is a slight increase in mAP when out-of-distribution (OOD) objects are excluded.

174 4 Results

175 4.1 Evaluation of Detection Models

176 Figure 1 and Table 2 show the mean average precision (mAP) of several state-of-the-art detection
 177 models evaluated on MSCOCO and NAO. Despite the fact that the images in NAO were chosen
 178 using an EfficientDet-D7 model, we observe that other object-detection architectures show a similar
 179 reduction in mAP when evaluated on NAO. Concretely, when using NAO the mAP of EfficientDet-D7
 180 is reduced by 68.3%, while Faster RCNN is reduced by 19.1% when compared to MSCOCO. Even
 181 though EfficientDet-D7 was developed more recently than Faster RCNN, the mAP on NAO is similar.
 182 This indicates that latest models are not more robust on NAO, despite their superior performance on
 183 MSCOCO evaluation sets. This in turn suggests that modeling improvements from recent years do
 184 not address the issue of high confidence misclassification in out-of-distribution samples.

185 We also calculate the mAP without out-of-distribution objects. That is, if a detection matches a
 186 bounding box that does not belong the MSCOCO object categories, the detection is not counted as a
 187 false positive. We can see that, this exclusion improves mAPs on NAO, but overall, the results are
 188 still considerably worse than those from the MSCOCO *val* and *test-dev* set.

189 4.2 Common Failure Modes

190 In Figure 4, we visualize some failure modes of the detection models on NAO. In most of the
 191 misclassified objects, the predicted class is superficially similar to the ground truth class, but obviously
 192 different in terms of function. For example, clocks and coins are similar in shape (circular), texture
 193 (metallic in some cases) and both have characters near the perimeters. However, they are very
 194 different in function and in scale, such that any human can easily tell the difference between the
 195 two. Similarly, airplanes and sharks are similar in overall shape, color, and texture, but exist in rather
 196 different scenes.

197 Another common failure mode is differentiating different animal species. For example, elephant and
 198 rhinoceros both have somewhat similar skin color and texture but they are very different in size and
 199 rhinoceros do not have the distinctive elephant trunk.

200 4.3 Dataset Blind Spot

201 As mentioned in Section 3.1, MSCOCO sourced images from Flickr search queries related to the
 202 80 object categories. This process can be seen as a biased sampling process of all captured photos,
 203 resulting in "blind spots" in MSCOCO. For example, because there is not any "fish" category, the
 204 frequency of photos taken underwater in MSCOCO is much lower than all captured photos. In this
 205 section, we investigate this sampling bias by comparing the image embeddings of BiT ResNet-50 [10]
 206 pretrained on ImageNet-21k [21] across the 3 datasets – OpenImages *train*, MSCOCO *train* and
 207 NAO. We consider OpenImages as a universal set for available images and with MSCOCO and NAO
 208 being a subset of the captured images. The image embedding is the output of the global average




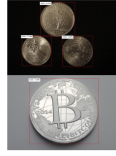
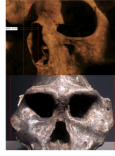



Prediction	Umbrella	Bird	Motorcycle	Clock	Donut	Zebra	Elephants	Airplane
Ground Truth	Moth (out of distribution)	Fish (out of distribution)	Car	Coin (out-of-distribution)	Skull (out-of-distribution)	Tiger (out-of-distribution)	Rhinoceros (out-of-distribution)	Shark (out-of-distribution)
Samples								

Figure 4: Selected samples to showcase common failure modes.

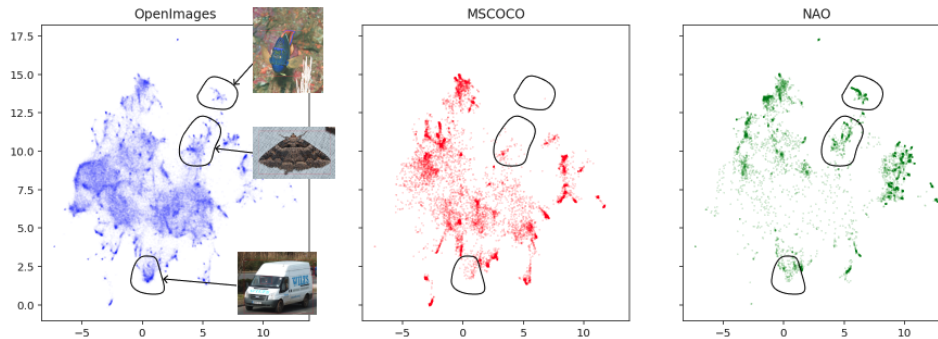


Figure 5: BiT ResNet-50 embeddings projected by UMAP on OpenImages *train*, MSCOCO *train* and NAO. NAO images are under-represented in MSCOCO.

209 pooling layer, resulting in a vector of size 2,048. We then use UMAP [15] to reduce the dimension to
 210 2 for visualization as shown in Figure 5.

211 When comparing the embedding space of MSCOCO with OpenImages, we found that there are
 212 regions where the density is significantly lower in MSCOCO than in OpenImages. Some of these
 213 low-density regions are indicated by the black circles in Figure 5. When cross-referencing with
 214 the embedding space of NAO, we can see that these low-density regions of MSCOCO are in fact high-
 215 density in NAO, indicating that the examples in NAO are exploiting the under-represented regions
 216 that arise from MSCOCO’s biased sampling process. We visualize 3 of such low-density clusters
 217 and they each reveal a common failure mode (i.e. fish misclassified as bird, insects misclassified as
 218 umbrella and van misclassified as truck.)

219 4.4 Background Cues

220 Hendrycks et al. [7] suggest that classification models are vulnerable to natural adversarial examples
 221 because classifiers are trained to associate the entire image with an object class, resulting in frequently
 222 appearing background elements being associated with a class. Object detection models are different
 223 from image classifiers in that they receive additional supervision about the object position and size.
 224 We instead argue that the primary cause of detection models being vulnerable to NAO is their tendency
 225 to focus too much on the information within the predicted bounding boxes.

226 In this section, we study the effect of object background on classification probabilities. Specifically,
 227 we quantify the change in probability of the detected object when its background is replaced. We use
 228 a MSCOCO-pretrained Mask-RCNN [6] with a ResNet 50 backbone to obtain instance segmentation
 229 masks on MSCOCO *val* and NAO. Then, we use the instance segmentation masks to retain only the
 230 most confident object and replace the rest of the image with a new background. There are 6 new
 231 backgrounds – underwater, beach, forest, road, mountain and sky – where Mask-RCNN detects no
 232 objects of probability higher than 0.1 from the backgrounds themselves. We measure the change
 233 of probability by matching the bounding box detected on the original image and the bounding box

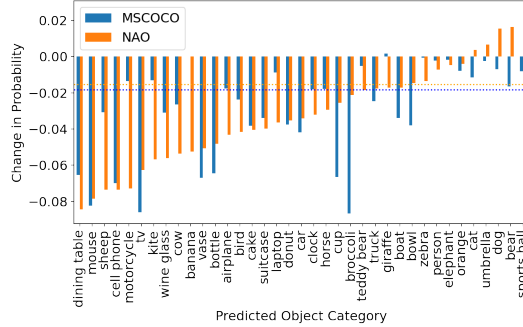


Figure 6: Average change in probability of objects when the backgrounds are replaced. The orange and blue dotted lines indicate average change in probabilities across all classes in MSCOCO and NAO. The small change in probability indicates that the detection model did not make use of background to classify the objects.

234 detected on the new image with the background replaced. We repeat this process for all images in
 235 NAO and MSCOCO *val* set and all 6 backgrounds.

236 As show in Figure 6, in both NAO and MSCOCO, the change in probabilities is low, indicating that
 237 the model does not make use of the background when detecting the object. While this robustness
 238 against background change is favorable in most cases, this also shows that the model also does not
 239 account for unlikely combinations of background and foreground objects. For example, when the
 240 model misclassifies a shark as an airplane, the network could have noticed that the detected "airplane"
 241 is underwater and assigned a lower probability to the class airplane.

242 4.5 Integrated Gradients Analysis

243 We further try to understand the source of the egregious misclassifications by computing the integrated
 244 gradients [23] of the network classification head output with respect to the input image. We aim to
 245 find the proportion of integrated output within the bounding box to understand if the network makes
 246 use the context of the object for detection and classification.

247 Specifically, we computed the gradients of the classification output of highest-scored bounding box
 248 with respect to the input image and measure the proportion of the sum of attribution inside the
 249 bounding box with respect to the total attribution. When there are multiple same-class objects to
 250 detect, we make sure to attribute each object separately. For example, when there are 2 people, we
 251 calculate the attribution of one person, ensuring the attribution of the other person is not counted
 252 towards the background. We used EfficientDet-D4 and randomly sampled 1000 images for this
 253 experiment. We found that for most classes, the majority of the attributions come from inside the
 254 bounding box.

255 Both Figure 6 and Figure 7 suggest that the detection model do not make use of background enough
 256 and instead mainly focus on the information within the predicted bounding box.

257 4.6 Patch Shuffling and Local Texture Bias

258 Geirhos et al. [4] demonstrated that classification networks are biased towards recognizing texture
 259 instead of shape. Brendel and Bethge [3] showed that a classification network can still reach a high
 260 level of accuracy using just small patches extracted from images. In this work, we show that detection
 261 models also show strong local texture bias, making them susceptible to adversarial objects with
 262 similar object subparts but are of another object category. For each prediction from EfficientDet-D7
 263 on MSCOCO and NAO, we randomly sample a patch from within the bounding box and swap it
 264 with another patch from inside the bounding box. We swap these random patches 3 times such that
 265 object is barely recognizable by just the shape. We then match the detected bounding box from
 266 the shuffled images to the original bounding box with the highest overlap. We repeat this shuffling
 267 process independently 10 times and record the absolute change in probability of the detected object.
 268 Figure 8 shows that there is only a modest reduction in probability after the shuffles.

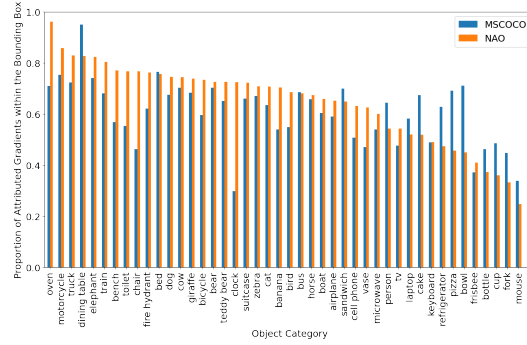


Figure 7: Proportion of attributed gradients within the bounding box by object category. In many classes, the detection model seldom make use of the object surroundings for classification and detection.

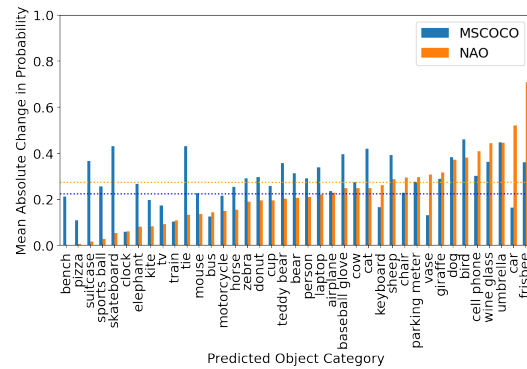


Figure 8: Mean absolute change in probability when patches inside the bounding box are swapped randomly. The blue and orange dotted line represent the mean average change in probability across all object categories for MSCOCO and NAO respectively. This confirms the texture bias hypothesis because even if the shape of the objects are heavily distorted while the local texture is intact, the network is still able to make the same prediction with similar confidence in most object categories.

269 5 Conclusion

270 We introduce "Natural Adversarial Objects" (NAO), a challenging robustness evaluation dataset for
 271 detection models trained on MSCOCO. We evaluated seven state-of-the-art detection models from
 272 various families, and show that they consistently fail to perform accurately on NAO, comparing to
 273 MSCOCO *val* and *test-dev* set, including on both in-distribution and out-of-distribution objects. We
 274 explained the procedure of creating such a dataset which can be useful for creating similar datasets in
 275 the future.

276 We expose that these naturally adversarial objects are difficult to classify correctly due to the "blind-
 277 spots" in the MSCOCO dataset. We also utilize integrated gradients, background replacement, and
 278 patch shuffling to demonstrate that detection models are overly sensitive to local texture but insensitive
 279 to background change, leading such models to be susceptible to natural adversarial objects. We hope
 280 NAO can facilitate further research about model robustness and handling out-of-distribution inputs.

281 References

- 282 [1] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S.
283 Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A
284 closer look at memorization in deep networks. volume 70 of *Proceedings of Machine Learning Research*,
285 pages 233–242, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL
286 <http://proceedings.mlr.press/v70/arpit17a.html>
- 287 [2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy
288 of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- 289 [3] Wieland Brendel and Matthias Bethge. Approximating CNNs with bag-of-local-features models works
290 surprisingly well on imagenet. In *International Conference on Learning Representations*, 2019. URL
291 <https://openreview.net/forum?id=SkfMWhAqYQ>.
- 292 [4] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland
293 Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and
294 robustness. In *International Conference on Learning Representations*, 2019. URL [https://openreview](https://openreview.net/forum?id=Bygh9j09KX)
295 [.net/forum?id=Bygh9j09KX](https://openreview.net/forum?id=Bygh9j09KX).
- 296 [5] Justin Gilmer, Ryan P. Adams, Ian Goodfellow, David Andersen, and George E. Dahl. Motivating the rules
297 of the game for adversarial example research, 2018.
- 298 [6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE*
299 *international conference on computer vision*, pages 2961–2969, 2017.
- 300 [7] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial
301 examples. July 2019.
- 302 [8] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry.
303 Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing*
304 *Systems*, pages 125–136, 2019.
- 305 [9] Jason Jo and Yoshua Bengio. Measuring the tendency of cnns to learn surface statistical regularities. *CoRR*,
306 abs/1711.11561, 2017. URL <http://arxiv.org/abs/1711.11561>.
- 307 [10] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil
308 Houlsby. Big transfer (BiT): General visual representation learning. December 2019.
- 309 [11] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab
310 Kamali, Stefan Popov, Matteo Mallocci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open
311 images dataset v4: Unified image classification, object detection, and visual relationship detection at scale.
312 November 2018.
- 313 [12] Tingting Liang, Xiaojie Chu, Yudong Liu, Yongtao Wang, Zhi Tang, Wei Chu, Jingdong Chen, and
314 Haibing Ling. Cbnetv2: A composite backbone network architecture for object detection. *arXiv preprint*
315 *arXiv:2107.00420*, 2021.
- 316 [13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár,
317 and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on*
318 *computer vision*, pages 740–755. Springer, 2014.
- 319 [14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object
320 detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988,
321 2017.
- 322 [15] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection
323 for dimension reduction. February 2018.
- 324 [16] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers
325 generalize to ImageNet? February 2019.
- 326 [17] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference*
327 *on computer vision and pattern recognition*, pages 7263–7271, 2017.
- 328 [18] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. April 2018.
- 329 [19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time object
330 detection with region proposal networks. June 2015.

- 331 [20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection
332 with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- 333 [21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang,
334 Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge.
335 *International journal of computer vision*, 115(3):211–252, 2015.
- 336 [22] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. OverFeat:
337 Integrated recognition, localization and detection using convolutional networks. pages 1–15, December
338 2013.
- 339 [23] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. March 2017.
- 340 [24] Mingxing Tan and Quoc V Le. EfficientNet: Rethinking model scaling for convolutional neural networks.
341 May 2019.
- 342 [25] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In
343 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–
344 10790, 2020.
- 345 [26] Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In *Proceedings of the*
346 *IEEE/CVF International Conference on Computer Vision*, pages 421–430, 2019.
- 347 [27] Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *International*
348 *Conference on Learning Representations*, 2018. URL [https://openreview.net/forum?id=](https://openreview.net/forum?id=H1BLjgZCb)
349 [H1BLjgZCb](https://openreview.net/forum?id=H1BLjgZCb).