Abstract

We introduce a set of image transformations that can be used as corruptions to evaluate the robustness of models. The primary distinction of the proposed transformations is that, unlike existing approaches such as Common Corruptions (Hendrycks & Dietterich, 2019), the geometry of the scene is incorporated in the transformations – thus leading to corruptions that are more likely to occur in the real world. We apply these corruptions to the ImageNet validation set to create 3D Common Corruptions (ImageNet-3DCC) benchmark. The evaluations on recent ImageNet models with robustness mechanisms show that ImageNet-3DCC is a challenging benchmark for object recognition task. Furthermore, it exposes vulnerabilities that are not captured by Common Corruptions, which can be informative during model development.

1. Introduction

Computer vision models deployed in the real world will encounter naturally occurring distribution shifts from their training data. These shifts range from lower-level distortions, such as motion blur and illumination changes, to semantic ones, like object occlusion. Each of them represents a possible failure mode of a model and has been frequently shown to result in profoundly unreliable predictions (Dodge & Karam, 2017; Hendrycks & Dietterich, 2019; Szegedy et al., 2013; Jo & Bengio, 2017; Geirhos et al., 2020). Thus, a systematic testing of vulnerabilities to these shifts is critical before deploying these models in the real world.

This work presents a set of distribution shifts in order to test models’ robustness for object recognition task. To achieve this, we leverage our recently proposed framework in (Kar et al., 2022), denoted as 3D Common Corruptions (3DCC). In contrast to previously proposed shifts which perform uniform 2D modifications over the image, such as Common Corruptions (2DCC, or equivalently, ImageNet-2DCC) (Hendrycks & Dietterich, 2019), 3DCC incorporates 3D information to generate corruptions that are consistent with the scene geometry. This leads to shifts that are more likely to occur in the real world.

Using the methods provided in (Kar et al., 2022), we apply 12 corruptions from 3DCC on ImageNet (Deng et al., 2009)

See the project page for code, data, models, and more results: http://3dcommoncorruptions.epfl.ch/
We propose a diverse set of new corruption operations ranging from defocusing (near/far focus) to lighting changes. As ImageNet (Deng et al., 2009) dataset does not come with 3D labels, we leverage a state-of-the-art depth estimator to obtain depth predictions, and then apply the methods from (Kar et al., 2022) using predicted depth to generate corruptions (See Sec. 2.1 for details). A subset of the corruptions marked in the last column are novel and commonly faced in the real world, but are not 3D based. We include them in our benchmark.

validation set. We denote the resulting set as ImageNet-3DCC (See Fig. 1 for examples). ImageNet-3DCC addresses several aspects of the real world, such as camera motion, weather, depth of field, and lighting. Figure 2 provides an overview of all corruptions in ImageNet-3DCC.

We show that the performance of the methods aiming to improve robustness of ImageNet models, including those with diverse data augmentation, reduce drastically under ImageNet-3DCC. Thus, it can serve as a challenging testbed for real-world corruptions, especially those that depend on scene geometry. It also exposes vulnerabilities that are not captured by ImageNet-2DCC, hence it can be used to better assess generalization of the existing models which may have overfitted to ImageNet-2DCC, which can be informative for development of better robustness mechanisms.

2. ImageNet-3DCC Benchmark

2.1. Corruption Types

Following (Kar et al., 2022), we define different corruption types, namely depth of field, camera motion, lighting, video, weather, and noise, resulting in 12 corruptions in ImageNet-3DCC. Most of the corruptions require an RGB image and scene depth, except the noise ones that can be generated from RGB image directly. As ImageNet does not have depth labels, we generate depth predictions using a state-of-the-art depth estimator trained on Omnidata (Eftekhar et al., 2021) dataset with 3D data augmentations (Kar et al., 2022) and consistency constraints (Zamir et al., 2020). Note that the corruptions generated using predicted depth from the state-of-the-art models (Ranftl et al., 2021; Eftekhar et al., 2021) are similar to those generated from ground truth depth, as shown in (Kar et al., 2022) (Sec. 5.2.4). Furthermore, as shown there, the 3D corruptions are also applicable to other object recognition datasets without 3D labels as well, e.g. COCO (Lin et al., 2014).

To generate the corruptions, we use the methods from (Kar et al., 2022), as explained in more detail below. Note that the semantic corruptions in (Kar et al., 2022) would require a mesh with semantic annotations to generate. We dropped them as ImageNet does not have those labels.

Depth of field corruptions create refocused images. They keep a part of the image in focus while blurring the rest. We consider a layered approach (Eftekhar et al., 2021; Barsky & Kosloff, 2008) that splits the scene into multiple layers. For each layer, the corresponding blur level is computed.

Figure 2. ImageNet-3DCC benchmark. We propose a diverse set of new corruption operations ranging from defocusing (near/far focus) to lighting changes. As ImageNet (Deng et al., 2009) dataset does not come with 3D labels, we leverage a state-of-the-art depth estimator to obtain depth predictions, and then apply the methods from (Kar et al., 2022) using predicted depth to generate corruptions (See Sec. 2.1 for details). A subset of the corruptions marked in the last column are novel and commonly faced in the real world, but are not 3D based. We include them in our benchmark.
using the pinhole camera model. The blurred layers are then composited with alpha blending. We generate near focus and far focus corruptions by changing the focus region to the near or far part of the scene.

**Camera motion** creates blurry images due to camera movement during exposure. To generate this effect, we first transform the input image into a point cloud using the depth information. Then, we define a trajectory (camera motion) and render novel views along this trajectory. As the point cloud was generated from a single RGB image, it has incomplete information about the scene when the camera moves. Thus, the rendered views will have disocclusion artifacts. To alleviate this, we apply an inpainting method from (Niklaus et al., 2019). The generated views are then combined to obtain parallax-consistent motion blur. We define XY-motion blur and Z-motion blur when the main camera motion is along the image XY-plane or Z-axis, respectively.

**Video** corruptions arise during the processing and streaming of videos. Using the scene 3D, we create a video using multiple frames from a single image by defining a trajectory, similar to motion blur. Inspired by (Yi et al., 2021), we generate average bit rate (ABR) and constant rate factor (CRF) as H.265 codec compression artifacts, and bit error to capture corruptions induced by imperfect video transmission channel. After applying the corruptions over the video, we pick a single frame as the final corrupted image.

**Weather** corruptions degrade visibility by obscuring parts of the scene due to disturbances in the medium. We define a single corruption and denote it as fog 3D to differentiate it from the fog corruption in 2DCC. We use the standard optical model for fog (Fattal, 2008; Sakaridis et al., 2018; Von Bernuth et al., 2019), similar to (Kar et al., 2022).

**Lighting** corruptions change scene illumination by modifying the original illumination. As ImageNet does not have full scene geometry information, it is not possible to perform ray-tracing. Thus, we only consider a flash corruption, where illumination decreases with increasing depth.
ImageNet-3DCC and define 5 shift intensities. We also calibrate the shift intensities so that the average SSIM (Wang et al., 2004) values of images for an ImageNet-3DCC corruption is similar to its counterpart in 2DCC. For the corruptions that do not have a counterpart in 2DCC, we adjust the distortion parameters to increase shift intensity while staying in a similar SSIM range as the others. The dataset can be accessed from the project page. Figure 3 shows example corruptions with different shift intensities.

As evaluation criteria, we follow 2DCC for compatibility and compute mean corruption error (mCE) by dividing the models errors by AlexNet (Krizhevsky et al., 2012) errors and averaging over corruptions.

3. Experiments

Models evaluated: We evaluate the robust ImageNet models (Geirhos et al., 2018; Rusak et al., 2020; Hendrycks et al., 2019; 2021) from RobustBench (Croce et al., 2020) and ImageNet-2DCC (Hendrycks & Dietterich, 2019) leaderboards. We directly use the provided weights, i.e. no training or fine-tuning is performed.

As shown in Fig. 4 (top), the performance of models degrade significantly, including those with diverse augmentations. Thus, ImageNet-3DCC can serve as a challenging benchmark for object recognition task. As expected, the general trends are similar between the two benchmarks as 2D and 3D corruptions are not completely disjoint. A similar observation was also made in (Mintun et al., 2021) even when the corruptions are designed to be dissimilar to 2DCC. Still, there are notable differences that can be informative during model development by exposing trends and vulnerabilities that are not captured by 2DCC. For example, ANT (Rusak et al., 2020) has better mCE on 2DCC compared to AugMix (Hendrycks et al., 2019), while they perform similarly on 3DCC. Likewise, combining DeepAugment (Hendrycks et al., 2021) with AugMix improved the performance on 2DCC significantly more than 3DCC.

To further understand the differences, we consider a subset of corruptions that exists in both benchmarks (e.g. 2D defocus blur vs its 3D version), namely near focus, far focus, xy-motion blur, z-motion blur, fog 3d, and flash from ImageNet-3DCC and defocus blur, motion blur, zoom blur, fog and brightness from 2DCC. We then compute the mCEs only on these subsets. The results shown in Fig. 4 (bottom) further reflects the differences where 1. all models have consistently higher normalized errors (mCEs) on 3D corruptions compared to their 2DCC counterparts and 2. certain models, e.g. AugMix and AugMix+DA, face a larger drop in performance on 3DCC compared to the other models, indicating that AugMix may be biased towards 2DCC.

Finally, in Fig. 5, we provide a detailed breakdown of performance on ImageNet-3DCC for a standardly trained ResNet-50 (He et al., 2016) from PyTorch (Paszke et al., 2019) model zoo. The performance degrades significantly from the performance on clean data, while some corruptions yield more severe errors than the others, e.g. xy-motion blur vs h265 crf. Examining this non-uniformity in performance could be informative to design better robustness mechanisms, e.g. targeted data augmentation, depending on the practical setting of interest.

4. Conclusion

We introduce ImageNet-3DCC to test model robustness against real-world distribution shifts, particularly those centered around 3D. Experiments demonstrate that ImageNet-3DCC is a challenging benchmark that exposes model vulnerabilities under real-world plausible corruptions that are not captured by 2D corruptions. We believe incorporating 3D information into benchmarking opens up a promising direction for robustness research.

Acknowledgements

This work was partially supported by the ETH4D and EPFL EssentialTech Centre Humanitarian Action Challenge Grant.

References

Barsky, B. A. and Kosloff, T. J. Algorithms for rendering depth of field effects in computer graphics. In Proceed-


