2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Enhanced Model Robustness to Input Corruptions by Per-corruption Adaptation of Normalization Statistics

Elena Camuffo[1,3,*], Umberto Michieli[1], Simone Milani[3], Jijoong Moon[2], Mete Ozay[1]

***Abstract*— Developing a reliable vision system is a fundamental challenge for robotic technologies (*e.g.*, indoor service robots and outdoor autonomous robots) which can ensure reliable navigation even in challenging environments such as adverse weather conditions (*e.g.*, fog, rain), poor lighting conditions (*e.g.*, over/under exposure), or sensor degradation (*e.g.*, blurring, noise), and can guarantee high performance in safety-critical functions. Current solutions proposed to improve model robustness usually rely on generic data augmentation techniques or employ costly test-time adaptation methods. In addition, most approaches focus on addressing a single vision task (typically, image recognition) utilising synthetic data. In this paper, we introduce Per-corruption Adaptation of Normalization statistics (PAN) to enhance the model robustness of vision systems. Our approach entails three key components: (i) a corruption type identification module, (ii) dynamic adjustment of normalization layer statistics based on identified corruption type, and (iii) real-time update of these statistics according to input data. PAN can integrate seamlessly with any convolutional model for enhanced accuracy in several robot vision tasks. In our experiments, PAN obtains robust performance improvement on challenging real-world corrupted image datasets (*e.g.*, OpenLoris, ExDark, ACDC), where most of the current solutions tend to fail. Moreover, PAN outperforms the baseline models by 20-30% on synthetic benchmarks in object recognition tasks.**

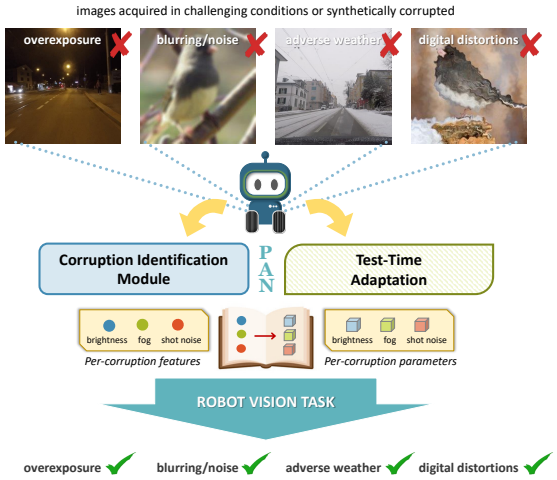


Fig. 1. Our approach enhances robot vision systems via per-corruption adaptive normalization of neural network models. This is fundamental in challenging environmental situations with corrupted images. Our proposed PAN is built on (i) a corruption identification module (CIM) that extracts per-corruption features in order to recognize the input corruption, (ii) an inexpensive test-time adaptation step to adapt model parameters to the specific corruption type, (iii) a codebook to map features to candidate parameters.

I. Introduction

A reliable perception system is one of the key components of autonomous robotics, both for outdoor (*e.g.*, autonomous driving systems) and indoor robotic systems (*e.g.*, home service robots like smart vacuum cleaner robots).

Advancements in deep learning technologies have led to the development of robust models for various robotic-related computer vision tasks, such as object recognition [1, 2], detection [3] and semantic segmentation [4, 5]. However, despite their high performance on standard benchmarks, these models often struggle with challenging environmental situations such as data corruptions [6], adversarial attacks [7], and domain shifts [8]. Factors like weather changes (*e.g.*, *snow*, *frost*, *fog*) or sensor degradation (*e.g.*, *shot noise*, *defocus blur*) experienced by robotic systems can introduce natural alterations or data corruptions [9–11]. Moreover, the data-oriented nature of deep neural networks (DNNs) and their complex architectures make them vulnerable to even minor distribution shifts, resulting in significant performance degradation. To address these challenges, researchers have introduced datasets with synthetic corruptions [6, 12], and real-world data collections acquired in heterogeneous adverse conditions, both outdoors [13–16] and indoors [17, 18]. As robotic applications increasingly adopt deep learning models, equipping mobile autonomous robots with a robust vision system is of fundamental importance, to ensure reliable navigation in any environment and guarantee high-level performance even for safety-critical functions (*e.g.*, autonomous driving, medical diagnostics, etc.). A popular strategy to enhance model robustness is through data augmentation techniques during pre-training [19–21], which aims to create a more generalizable model against corrupted images. Another approach is Test-Time Adaptation (TTA), which dynamically adjusts a pre-trained model's behaviour based on characteristics of test data [22–24], enabling it to perform better in varying conditions at test time.

In this paper, we propose PAN (Per-corruption Adaptation of Normalization statistics), a novel strategy to improve model robustness in robotic applications (Fig. 1). PAN dynamically adapts normalization layers' parameters based on the type of corruption identified in an input image using a Corruption Identification Module (CIM). Our approach is simple yet effective, compatible with various convolutional architectures, and enhances accuracy on corrupted test data without burdensome training procedures. We demonstrate the effectiveness of PAN through extensive evaluations, achieving

*Research completed during internship at Samsung R&D Institute UK.
[1]Samsung R&D Institute UK (SRUK), Communications House, South St, Staines, Surrey, United Kingdom {u.michieli,m.ozay}@samsung.com
[2]Samsung Research Korea, Seoul R&D Campus, 56, Seongchon-gil, Seocho-gu, Seoul, Rep. of Korea jijoong.moon@samsung.com
[3]University of Padova, via Gradenigo 6/b, 35129, Padova, Italy. {elena.camuffo,simone.milani}@dei.unipd.it

up to 30% relative accuracy gain compared to the state of the art. Our method is computationally and memory efficient, hence it is suitable for on-device robotic applications.

II. Related Work

Common image corruptions have various causes and occur frequently in real-world situations. These issues can be problematic for a wide variety of robot-related tasks, including localization [25], navigation [26] and vision-related tasks [9, 27] (also dealt with in this work). Considering robot vision, mainstream methods for tackling this problem can be broadly distinguished into two main categories: data augmentation and test-time adaptation methods.

Data Augmentation methods improve model performance during pre-training via data augmentation: they aim to develop a general robust model against corrupted images [19–21, 28]. Some methods improve the robustness of models by automatically searching for improved data augmentation policies among common methods [29], or applying random noise or patches to train images [30, 31]. Other approaches transform each image in a dataset by mixing it with a collection of images [21, 32] or automatically generating patterns [28], to improve model generalization by out-of-distribution examples and prevent overfitting on the training distribution. This technique is effective also on robot vision tasks other than image recognition [27]. Recent works propose mixed augmentation strategies in the frequency domain [33], as common corruptions mostly affect frequency components: APR [20] re-combines the phase spectrum of one image and the amplitude spectrum of another image, HA [19] includes hierarchical augmentations at variable frequency spectra.

Test-Time Adaptation (TTA) methods focus on resolving data distribution shift at test-time, using data from target domains [22, 23, 34]. These methods have been widely employed in robotic-related vision problems such as image registration [35], depth estimation [36] and point cloud upsampling via meta-learning [37]. There are different types of TTA methods. Some assume that target domain data can be observed simultaneously during adaptation, *e.g.*, adapting a source model on the target domain data using self-supervised loss, and employing the features obtained from the intermediate layers of the adapted model to refine the pseudo labels for the entire target dataset [38]. Other TTA methods assume that target data is received by the system in mini-batches [34, 39] and updates statistics at each iteration. For instance, AugBN [40] estimates normalization statistics of the unseen test distribution from the given test images in a mini-batch, using only one forward pass. TTA can also be applied to streams of data (sampled from a new data distribution, distinct from the source data distribution) instead of a fixed test set, in an online manner [23]. However, samples obtained at test time may come from a variety of different distributions, leading to new challenges, such as error accumulation and catastrophic forgetting [41]. To address this issue, a few methods [42, 43] investigate the continual test time adaptation problem that adapts the pre-trained source model to the continually changing test data.

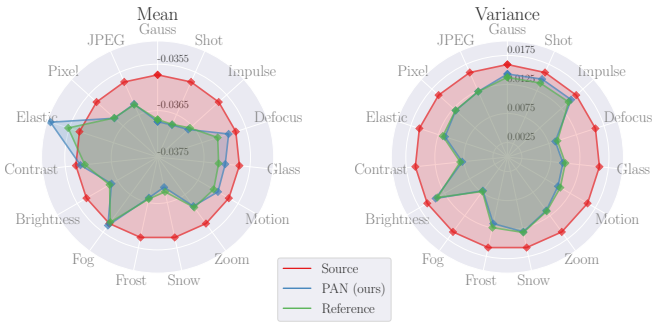


Fig. 2. Statistics estimated at normalization layers vary depending on the image corruption type, averaged on all layers (ResNet18 on ImageNet-C). Unlike classical data augmentation approaches where a single set of normalization statistics is estimated for all corruption types on a source domain (**red**), our method estimates normalization statistics for each corruption (**blue**), which are very close to the reference ones, estimated assuming that the true corruption type of the data is known (**green**).

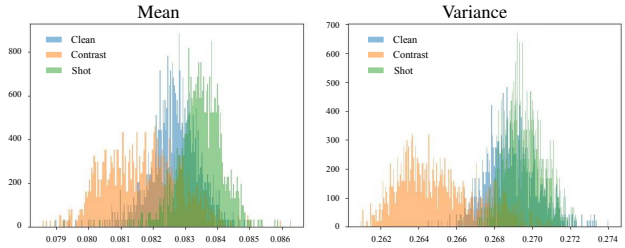


Fig. 3. Mean and variance distributions of the output of the first BN layer when encountering *clean* data, *contrast* corrupted data and *shot* noise corrupted data (ResNet18 on ImageNet-C).

SAR [24] employs an optimization scheme, which removes samples with large gradients and encourages model weights to lie in a flat minimum. NOTE [22] performs a selective mixing strategy that only calibrates the batch normalization layers statistics for detected out-of-distribution samples. ONDA [44] estimates BN statistics via running average on test batches.

III. Methodology: Per-corruption Adaptive Normalization (PAN)

Our approach builds upon the observation that statistics of BN layers in any convolutional architecture significantly differ for images corrupted according to different corruption types (Fig. 2), but are similar for images with the same corruption type (Fig. 3). Some previous work [22, 45, 46] explored adaptation of statistics of normalization layers for TTA, keeping a single set of normalization parameters for all corruptions, to build generic normalization layers to accommodate any input corruption. Instead, we build multiple sets of normalization statistics estimated for each corruption type. PAN is composed of three parts:

1) A corruption type identification module (Sec. III-B).
2) A per-corruption adaptation method for adapting statistics of BN layers to various corruption types at inference time (Sec. III-C).
3) A codebook to map the identified corruption type to the respective set of BN statistics (Sec. III-D).

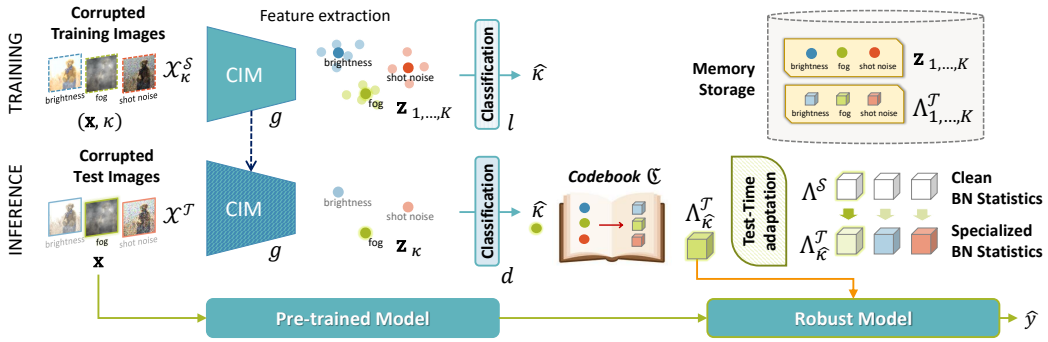


Fig. 4. The corruption identification module (Sec. III-B) is trained on corrupted training images and a set of corruption-related prototypical features $\mathbf{z}_{1,\dots,K}$ is built, by averaging features $\mathbf{z}$ relative to each corruption. Then, at *inference time*, the CIM is frozen and a *Codebook* $\mathfrak{C}$ (Sec. III-D) maps the corruption identified by the CIM to the respective corruption-specific BN parameters. Such parameters are initialized with the ones of the pre-trained downstream task model $F(\cdot)$ on clean source images $\mathcal{X}^{\mathcal{S}}$ and adapted to test images via TTA, separately for each identified corruption $\hat{\kappa}$ (Sec. III-C), obtaining a corruption-specific set $\Lambda^{\mathcal{T}}_{\hat{\kappa}}$. Finally, $\Lambda^{\mathcal{T}}_{\hat{\kappa}}$ is plugged into $F(\cdot)$ achieving enhanced robustness on downstream tasks, specifically on the identified corruption. The systems stores $\mathbf{z}_{1,\dots,K}$ and $\Lambda^{\mathcal{T}}_{1,\dots,K}$ to use and update them while doing inference.

A. Problem Setup: Improving Model Robustness

Image corruption: Let $F(\mathbf{x}, y; \mathcal{W})$ be a DNN model mounted on a robot for visual scene understanding. The aim of $F(\cdot)$ is to approximate ground truth labels $y \in \mathcal{Y}$ of input images $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{w \times h \times 3}$ optimizing its set of learnable parameters $\mathcal{W}$ (*e.g.*, weights and biases of the network architecture of the model). Among these parameters, we denote the set of parameters of its BN layers by $\Lambda \subset \mathcal{W}$. Samples of a source (*clean*) dataset $\mathcal{D}^{\mathcal{S}} = \{\mathcal{X}^{\mathcal{S}}, \mathcal{Y}^{\mathcal{S}}\}$ are drawn from a probability distribution $P^{\mathcal{S}}(\mathbf{x})$ on a *source* domain $\mathcal{S}$. Then, we consider a target (*corrupted*) dataset $\mathcal{D}^{\mathcal{T}} = \{\mathcal{X}^{\mathcal{T}}, \mathcal{Y}^{\mathcal{T}}\}$ of distorted images sampled from a *target* domain $\mathcal{T}$. We make a distinction between real (*endogenous*) and synthetic (*exogenous*) distortions as follows:

Endogenous distortions are natural corruptions that imply a shift in image statistics due to either inherent noise of camera sensors, deformations of objects observed in the images, or divergence of patterns of the objects. This is the most general case, where *target* test data $\mathcal{X}^{\mathcal{T}}$ cannot be parametrized by any operator. We denote a corrupted set of images presenting the same type of corruption, *e.g.*, dark images, as $\mathcal{X}^{\mathcal{T}}_u \sim P^{\mathcal{T}}_u(\mathbf{x})$, where u denotes the corruption type. The distribution of the images in the corrupted set is different from the source $P^{\mathcal{S}}(\mathbf{x})$.

Exogenous distortions are synthetic approximations of real corruptions provided by a function of clean images. They are obtained assuming that there exists an operator $C_{k,s}$ which corrupts a given set of clean images $\mathcal{X}^{\mathcal{S}}$ by $C_{k,s}(\mathcal{X}^{\mathcal{S}}) =: \mathcal{X}^{\mathcal{S}}_{k,s}$ where k denotes the synthetic corruption type and s denotes its severity level. Images of each corrupted set $\mathcal{X}^{\mathcal{S}}_{k,s}$ are sampled from $P^{\mathcal{S}}_{k,s}(\mathbf{x}) = \psi_{k,s}(P^{\mathcal{S}}(\mathbf{x}))$ as the operator $C_{k,s}$ transforms the distribution by a non-linear transformation $\psi_{k,s}(\cdot)$ according to the corresponding corruption type k and severity s. Synthetic corruptions attempt to approximate real corruptions, *i.e.*, $\psi_{k,s}(P^{\mathcal{S}}(\mathbf{x})) \approx P^{\mathcal{T}}_u(\mathbf{x})$ for some u.

B. Corruption Identification Module (CIM)

Our CIM is designed using a convolutional encoder followed by a linear classifier, taken from [47].

Architecture of the CIM. Extraction of corruption-specific features is accomplished through a DNN model $r = l \circ g$ [47], composed of a convolutional encoder $g(\cdot)$ that projects an input image $\mathbf{x}$ to a feature vector by $\mathbf{z} = g(\mathbf{x})$, and a linear layer $l(\mathbf{z})$ that outputs corruption identification probabilities.

Training: The CIM performs a corruption classification task to recognize the corruption type of input images. The CIM is trained on a set $\mathcal{D}_K := \bigcup_{\kappa=1}^{K} \mathcal{D}^{\mathcal{T}}_{\kappa}$, where each $\mathcal{D}^{\mathcal{T}}_{\kappa} = \{\mathcal{X}^{\mathcal{T}}_{\kappa}, \kappa\}$ is a dataset of images corrupted with some corruption type κ, and κ is known. Note that κ refers here to either endogenous or exogenous corruptions, depending on the available data. CIM is trained end-to-end following [47] via distance-based contrastive training using a Class Anchor Clustering (CAC) loss defined by

$$\mathcal{L}_{CAC}(\mathbf{x}, y) = \mathcal{L}_T(\mathbf{x}, y) + \lambda \mathcal{L}_A(\mathbf{x}, y), \tag{1}$$

where $\mathbf{x}$ is the input image with its label y and λ is a hyperparameter. The CAC loss aggregates two individual losses: i) a tuplet loss $\mathcal{L}_T(\mathbf{x}, y)$ [47, 48] used to minimize the distance between training samples and their ground-truth anchored class centre, and ii) an anchor loss $\mathcal{L}_A(\mathbf{x}, y)$ [47] used to maximize the distance to other anchored class centres. Thereby, the CAC loss $\mathcal{L}_{CAC}$ encourages training data to form tight and class-specific clusters, and anchored class centres to fix cluster centre positions during training.

Inference: After training the CIM model $r(\cdot)$ on $\mathcal{D}_K$, the final layer $l(\cdot)$ is removed and the feature extractor $g(\cdot)$ is used to extract q-dimensional features $\mathbf{z} \in \mathbb{R}^q$ from corrupted samples[1]. Then, prototypical features $\bar{\mathbf{z}}_\kappa = \frac{1}{h_\kappa} \sum_{i=0}^{h_\kappa} \mathbf{z}_i$ are computed from the training set, where each $\mathbf{z}_i$ is a feature vector corresponding to an image corrupted with corruption κ, and h_κ is the number of samples affected by the corruption κ. The calculated K prototypes are concatenated by $\bar{\mathbf{Z}} = [\bar{\mathbf{z}}_1^T, \bar{\mathbf{z}}_2^T, \dots, \bar{\mathbf{z}}_K^T]^T$ to construct the prototype matrix $\bar{\mathbf{Z}} \in \mathbb{R}^{K \times q}$ where $(\cdot)^T$ denotes the vector/matrix transpose.

We employ a distance-based classifier $\phi(\cdot, \cdot)$ to classify features according to their relative distance to

[1]Empirically, we found that using the features $\mathbf{z}$ instead of corruption identification probabilities generalizes better to unseen corrupted test data.

prototypical features. The classifier $\phi(\mathbf{z}, \bar{\mathbf{Z}})$ outputs $\mathbf{d} = (||\mathbf{z} - \bar{\mathbf{z}}_1||_2, \ldots, ||\mathbf{z} - \bar{\mathbf{z}}_K||_2)^T$ where $||\cdot||_2$ denotes the Euclidean norm.

The output is normalized by $\mathbf{b} = \mathbf{d} \odot (1 - \text{softmin}(\mathbf{d}))$ [47], where $\odot$ is the element-wise product, and

$$\text{softmin}(\mathbf{d})_\kappa = \frac{\exp^{-d_\kappa}}{\sum_{\kappa=1}^{K} \exp^{-d_\kappa}}, \quad \mathbf{d} = [d_\kappa]_{\kappa=1}^{K}, \tag{2}$$

is utilized to match the feature with the closest prototype. Then, the model $r' = \phi \circ g$ predicts the corruption affecting the input by

$$\hat{\kappa} = \arg\min_{\kappa}(\mathbf{b}). \tag{3}$$

C. Per-corruption Adaptation of BN Statistics

Batch Normalization (BN) [49] is a technique, used to make training of artificial neural networks faster and more stable through normalization of the layer inputs by re-centering and re-scaling. It is widely used in DNNs to mitigate the problem of internal covariate shift, where changes in the distribution of the inputs of each layer affect the learning of the network. BN is applied over a 4D input (a mini-batch of 2D inputs with additional channel dimension) [50].

Let $\mathcal{B}$ denote a mini-batch of features, obtained using model $F(\cdot)$, and let $\mathbf{f} \in \mathcal{B} \subset \mathbb{R}^{B \times D \times L}$ be a feature map in the mini-batch, where B, D, and L denote the batch size, the depth and the size of each feature map, respectively. The mean $\mu \in \mathbb{R}^D$ and standard-deviation $\sigma \in \mathbb{R}^D$ (BN statistics) are employed per-dimension over the mini-batches channel-wise for normalizing features using

$$\text{BN}(\mathbf{f}; \mu, \sigma^2) := \gamma \frac{\mathbf{f} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \tag{4}$$

where γ and β are learnable affine parameter vectors of size D, and $\epsilon > 0$ is a small constant used for numerical stability.

Test-Time Adaptation (TTA) refers to adapting DNNs to distribution shifts, with access to only the unlabelled test samples belonging to the target domain $\mathcal{T}$ at test time. The conventional way of employing BN in test time is to set μ and σ^2 as those estimated from source data. Instead, TTA methods estimate BN statistics directly from test batches to reduce the distribution shift at test time by

$$\mu = \frac{1}{B \cdot L} \sum_{\mathbf{f} \in \mathcal{B}} \mathbf{f}, \quad \sigma^2 = \frac{1}{B \cdot L} \sum_{\mathbf{f} \in \mathcal{B}} (\mathbf{f} - \mu)^2. \tag{5}$$

This practice is simple yet effective and thus adopted in many recent TTA studies [22, 23, 42, 45, 46]. In our paper, we propose updating BN statistics via TTA, separately, per each corruption type, as described next.

Estimating statistics on test data. Let $\Lambda := (\mu, \sigma^2) \subset \mathcal{W}$ be the set of BN statistics of the model $F(\mathbf{x}, y; \mathcal{W})$. We denote the set of BN statistics obtained after training the model on the source dataset by $\Lambda^{\mathcal{S}}$. We first initialize K sets of source BN parameters $\Lambda^{\mathcal{S}}$. Then, we update each set according to the corruption type present in the input image. In the ideal case, each set is associated to a specific corruption type κ, and each corruption type is always identified correctly. Therefore, the BN statistics $\Lambda_\kappa^{\mathcal{T}}$ associated with the type κ are updated only with images corrupted with corruption type κ that belong to the test set $\mathcal{D}_\kappa^{\mathcal{T}}$. We define this ideal reference set of statistics by $\Lambda_\kappa^{\text{ref}}$. However, the target corrupted test images come without the corruption label κ, and BN parameters must be computed on the corruption type estimated by CIM ($\hat{\kappa}$).

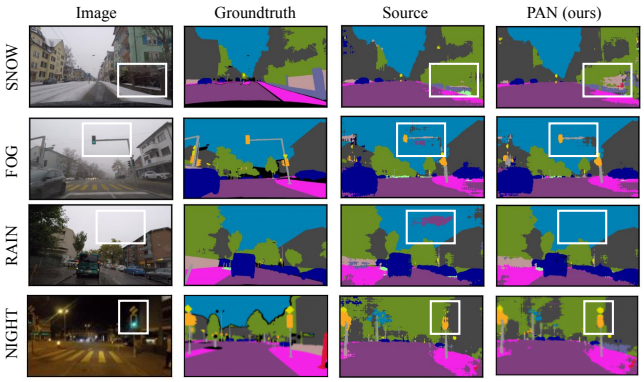


Fig. 5. Per-Corruption qualitative results of semantic segmentation using the DeepLabV2 [55] with the ACDC [15] dataset.

D. Use CIM and TTA to Improve Task Performance

When deployed on a robotic device, our system is composed of (i) a CIM module (Sec. III-B) employed to recognise the corruption type affecting the unlabelled input test image, and (ii) K sets of clean BN statistics, obtained training a model $F(\cdot)$ on clean training data (Sec. III-C). The purpose of our PAN is to improve the downstream task performance of $F(\cdot)$ by using CIM to identify the correct corruption type, update the correct set of BN parameters via TTA, and finally plug the updated set of BN parameters into the network.

Codebook mapping. In detail, at inference time, for each input test image $\mathbf{x} \in \mathcal{X}^{\mathcal{T}}$, we estimate the corruption type using the CIM by $r'(\mathbf{x}) = \hat{\kappa}$. Then, we use a *codebook* $\mathfrak{C}$ to map each estimated corruption type $\hat{\kappa}$ to a corruption-specific set of BN statistics $\Lambda_{\hat{\kappa}}^{\mathcal{T}}$ by

$$\mathfrak{C} : \hat{\kappa} \longmapsto \Lambda_{\hat{\kappa}}^{\mathcal{T}} := (\gamma_{\hat{\kappa}}, \beta_{\hat{\kappa}}). \tag{6}$$

Note that BN statistics associated with each of the K corruptions are initialized as $\Lambda^{\mathcal{S}}$, and will be assigned to $\Lambda_{\hat{\kappa}}^{\mathcal{T}}$ after they are estimated by TTA. The more CIM is able to correctly recognize the corruption (when $\hat{\kappa} = \kappa$), the more the BN statistics are specialized for such corruption and are different from the others. Fig. 2 shows that our PAN can obtain BN statistics close to reference ones $\Lambda^{\mathcal{T}} \approx \Lambda^{\text{ref}}$.

IV. Experiments

This section describes the experimental analysis of our approach. We start our investigation by presenting results spanning different real and synthetic datasets, architectures, and visual scene understanding tasks, showing that PAN improves over the baseline. Then, we examine our approach on common synthetic recognition benchmarks by comparing our PAN against common methods used to solve dataset shift problems and delving into more detailed ablation studies.

As discussed throughout the paper, our PAN provides a versatile vision system, with limited memory and storage

TABLE I

PAN IMPROVES OVER DIFFERENT ARCHITECTURES, DATASETS AND TASKS. BN PARAMETERS' SIZE IS NEGLIGIBLE WITH RESPECT TO THE SIZE OF THE WHOLE MODEL. THE NUMBER OF PARAMETERS OF EACH ARCHITECTURE IS ALSO REPORTED WITH THE PERCENTAGE USED FOR TEST-TIME ADAPTATION. REAL: NATURAL CORRUPTIONS.

Model	Backbone	Dataset	Task	Real	Source	PAN (ours)	Gain (%)	Metric	Model (MB)	BN (MB)
ResNet18 [1]	—	ImageNet-C [6]	Object Recognition		31.7	39.0	23.0	CA ↑	44.6	0.04
ResNet50 [1]	—	ImageNet-C [6]	Object Recognition		46.1	47.7	3.5	CA ↑	97.7	0.20
ResNet101 [1]	—	ImageNet-C [6]	Object Recognition		53.0	55.5	4.7	CA ↑	170.3	0.40
MobileNetV3 [51]	—	ImageNet-C [6]	Object Recognition		32.9	34.4	4.6	CA ↑	9.7	0.05
ResNeXt50 [52]	—	ImageNet-C [6]	Object Recognition		49.6	51.3	3.4	CA ↑	95.7	0.26
Wide-ResNet50 [53]	—	ImageNet-C [6]	Object Recognition		49.0	50.2	2.4	CA ↑	263.0	0.26
ResNet50 [1]	—	VizWiz [17]	Object Recognition	✓	39.1	43.8	12.0	CA ↑	97.7	0.20
ResNet50 [1]	—	OpenLORIS [18]	Object Recognition	✓	42.5	43.8	3.1	CA ↑	97.7	0.20
YOLOv8n [3]	CSPNet [54]	VOC-C [12]	Object Detection		34.6	36.3	4.9	mAP^{50-95} ↑	12.1	0.04
YOLOv8n [3]	CSPNet [54]	ExDARK [16]	Object Detection	✓	39.4	40.3	2.3	mAP^{50-95} ↑	12.1	0.04
DeepLabV2 [55]	MobileNetV2 [56]	Cityscapes-C [12]	Sem. Segmentation		34.5	41.5	20.3	mIoU ↑	42.2	0.11
DeepLabV2 [55]	MobileNetV2 [56]	ACDC [15]	Sem. Segmentation	✓	37.8	40.1	6.1	mIoU ↑	42.2	0.11

requirements that can be deployed on several robotic devices for different purposes (both indoors and outdoors). For example, a robot vacuum cleaner equipped with PAN can navigate safely in low light conditions, and overexposed regions, and can promptly adapt to rapid illumination changes. In outdoor environments, we can appreciate the effects of PAN in *e.g.*, autonomous vehicles: PAN is beneficial when navigating dark environments or in the presence of extreme weather conditions, such as fog, rain, or snow.

Evaluation metrics follow recent works [6, 19, 22, 23]. Robustness to corruptions is evaluated either using the Classification Accuracy (CA %, ↑) or the mean Corruption Error (mCE, ↓), for the object recognition task. mCE calculates the average classification error of the model on each corruption type over all the severity levels, normalized by the error obtained using AlexNet [6]. We use mean Intersection over Union (mIoU %, ↑) for semantic segmentation and mean Average Precision [3] (mAP^{50-95} %, ↑) for object detection.

A. Versatility of PAN on Robotic Vision Data

We present results on various robotics datasets, employing different models and performing different tasks, in order to show the versatility of PAN to outperform source models in various setups (Tab. I). Following most of the recent approaches [6, 19, 40], we start showing experiments on datasets synthetically corrupted by exogenous distortions.

First, we experiment on the **ImageNet-C** [6] using several different convolutional architectures (with BN layers). We show that the CA of the model grows by up to 7.2% applying PAN on top of the source model, and the relative gain does not depend on the model architecture. Second, we examine the applicability of PAN to various vision tasks, *i.e.*, object recognition, detection and semantic segmentation with either synthetic or real-world corruptions. Hence, we use the **VOC-C** [12] for object detection and the **Cityscapes-C** [12] for semantic segmentation (both with synthetic corruptions).

PAN improves accuracy significantly when applied to synthetically corrupted datasets on all tasks. We note that the percentage gain of PAN depends on the utilised model, but it is not much affected by the number of the employed BN layers (that is the only part of the architecture affected by PAN, as we will see in Sec. IV-C). Despite obtaining good performance on synthetic distortions, many state-of-the-art technologies fail when applied to real-world corruptions. On the other hand, achieving good performance on these datasets is essential to deploy reliable AI systems on real robotic devices. With this aim, we devise experiments training PAN on the VizWiz-classification [17] dataset (detailed in the next subsection) and three common robotic vision benchmarks:

OpenLORIS [18]. Object Recognition Dataset (OpenLORIS-Object) is designed for incremental recognition of common objects in office or home scenarios. The dataset includes some of the common challenges that home robots usually face, *e.g.*, different illumination conditions, occlusions, camera-object distances/angles, and context information (clutters).[2]

ExDARK [16]. The Exclusively Dark dataset is a collection of 7,363 low-light images captured in 10 different conditions from very low-light environments to twilight with 12 object classes annotated with object bounding boxes. The dataset is designed for indoor and outdoor *object detection*.[2]

ACDC [14]. The Adverse Conditions Dataset with Correspondences is a popular benchmark in autonomous driving, generally used for *semantic segmentation* in adverse visual conditions. It comprises a large set of 4006 images which are evenly distributed between fog, nighttime, rain, and snow.

We note that PAN outperforms the source model also in these challenging situations, even if the gain is smaller than those obtained on synthetic corruptions due to the less neat separation of corruption types applied to the input (*e.g.*, multiple corruption types co-exist on target data). Fig. 5 shows some qualitative results of PAN obtained by performing semantic segmentation in an outdoor autonomous driving environment. PAN shows better segmentation maps with respect to baseline predictions for every corruption. Tab. I reports also the model size and the size of the BN layers for each architecture used; the latter is small and almost negligible compared to the former.

B. Comparisons with Other Approaches

Following previous works, we compare our method with state-of-the-art approaches on synthetically corrupted test datasets. We use the CIFAR10, CIFAR100 [49] and ImageNet [60] datasets. The CIFAR datasets comprise of $50,000$

[2] To fit our purpose, we use the data with the lowest severity level for training and data with higher severity levels for testing.

TABLE II

CA (%, ↑) ON **CIFAR-C** WITH RESNET18. COMPARISONS FROM [22].

	CIFAR10-C	CIFAR100-C	Avg CA
Source	57.7	33.4	45.6
BN Adapt [46]	26.6	35.0	30.8
ONDA [44]	36.4	50.4	43.4
Pseudo label [38]	24.6	33.6	29.1
TENT [23]	23.6	33.1	28.3
CoTTA [42]	24.5	35.8	30.1
LAME [34]	63.9	36.7	50.3
NOTE [22]	78.9	53.0	65.9
PAN (ours)	**80.9**	**62.1**	**71.5**

TABLE III

CA (%, ↑) ON **VIZWIZ** [17] WITH RESNET50. LEFT TO RIGHT: 6 SPECIFIC CORRUPTIONS (*blur*, *bright*, *framing*, *rotation*, *obscured*, *dark*), AVERAGE CORRUPTED, CLEAN SOURCE AND TOTAL AVERAGE; AS DEFINED IN [17, 57].

	BLR	BRT	FRM	ROT	OBS	DRK	Corr	Clean	Total
Source	39.3	32.8	36.2	26.5	23.7	40.8	33.2	42.8	39.1
AugMix [21]	41.0	34.4	39.3	31.1	26.6	42.5	35.8	46.3	42.0
APR [20]	37.6	31.3	35.7	27.0	26.0	38.0	32.6	43.2	38.9
HA [19]	37.7	32.2	35.6	26.9	27.2	40.1	33.3	41.3	38.3
DA [58]	39.3	33.8	38.1	30.9	30.2	40.8	35.5	45.6	41.1
BN Adapt [46]	25.2	21.3	23.1	18.5	20.7	24.4	22.2	32.9	27.7
PAN (ours)	**41.2**	**40.6**	**39.4**	**34.4**	**36.1**	**44.3**	**39.3**	**47.9**	**43.8**

TABLE IV

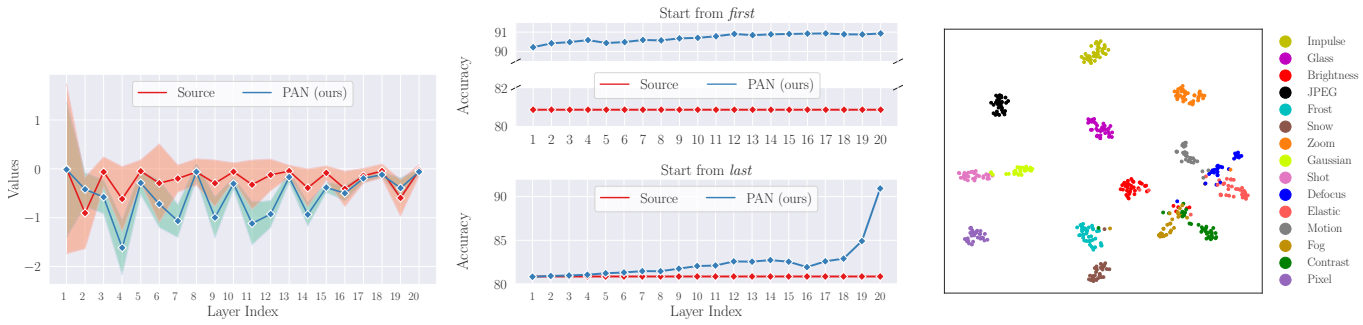
MCE (%, ↓) ON **IMAGENET-C** WITH RESNET50. WE COMPARE STATE-OF-THE-ART DATA AUGMENTATION (RESULTS FROM [19]) AND TTA (THE FIRST THREE RESULTS ARE OBTAINED FROM [40]) METHODS. $^{\heartsuit}$: PRE-TRAINED ON SAMPLES CORRUPTED WITH TARGET CORRUPTIONS. $^{\diamondsuit}$: PRE-TRAINED VIA DA+AUGMIX+HA.

	Noise	Blur	Weather	Digital	mCE
Source	80	84	77	82	80.6
Source$^{\heartsuit}$	64	59	59	64	61.4
Patch Uniform [31]	68	79	73	76	74.3
AA [29]	70	80	69	72	72.7
Random AA [29]	71	82	73	77	76.1
MBPool [59]	74	79	67	74	73.4
SIN [30]	70	80	71	73	73.3
AugMix [21]	66	71	68	69	68.4
APR [20]	65	77	61	71	68.9
HA [19]	57	70	65	69	65.8
PixMix [28]	52	79	60	69	65.8
DA [58]	46	71	62	66	62.0
PTN [45]	110	124	133	144	128.7
BN Adapt [46]	70	79	62	74	70.9
AugBN [40]	69	77	61	72	69.8
TENT [23]	104	103	84	88	94.3
SAR [24]	78	93	65	86	80.5
EATA [43]	68	83	56	70	69.2
PAN (ours)	74	76	50	67	66.0
Source$^{\diamondsuit}$	45	59	56	62	56.1
PAN$^{\diamondsuit}$ (ours)	42	51	50	52	49.4

32×32 training images. The ImageNet contains around 1.2M images belonging to 1000 different classes. Evaluation is performed on the corrupted versions of these datasets' test splits, *i.e.*, CIFAR10-C, CIFAR100-C, and ImageNet-C [6]. For those datasets, corruptions are simulated for 4 categories (*noise*, *blur*, *weather*, *digital*) with $K = 15$ corruption types, each with 5 severity levels. For a fair comparison with the prior art [6, 19, 22, 23], we adopt the ResNet18 [1] for the CIFAR datasets, and the ResNet50 [1] for the ImageNet. We use *source* pre-trained weights obtained from [61]. Results obtained using **CIFAR-C** are reported in Tab. II, comparing our PAN against several state-of-the-art TTA methods. Our approach improves the best competitor (*i.e.*, NOTE) by 2.5% and 17.2% CA on the CIFAR10-C and CIFAR100-C, respectively. Results obtained using the **ImageNet-C** are reported in Tab. IV. First, we observe that the model pre-trained on clean data (*i.e.*, Source) suffers from severe performance degradation. This degradation is attenuated by applying DA and TTA approaches. However, DA methods require re-training the model to enable robustness and train a single set of parameters for all corruption types, while TTA can be implemented a posteriori after having a trained model. PAN outperforms all compared TTA methods (by a significant margin) and most DA approaches. We observe that pre-training the whole model with the corruptions encountered in the test set (Source$^{\heartsuit}$) significantly improves the results, outperforming most corruption-agnostic data augmentation approaches. Nevertheless, PAN obtains a larger gain. We include this experiment to motivate our choice of specialising BN layer parameters to each corruption separately.

Conversely, a robust model pre-trained with heavy data augmentation improves accuracy significantly over the standard Source baseline. The best augmentation approach is obtained by combining multiple state-of-the-art pipelines (*i.e.*, Source$^{\diamondsuit}$ using DA+AugMix+HA). TTA methods achieve comparable results with the key benefit of adaptation at test time rather than having to retrain the model from scratch with additional augmentations. Our PAN can be seamlessly integrated on top of any pre-trained model. To examine this hypothesis, we employ PAN starting from pre-trained weights DA+AugMix+HA (PAN$^{\diamondsuit}$) and observe large gains by about 10% relative mCE.

Finally, we analyze some per-corruption results on the **VizWiz** [17, 62] dataset in Tab. III. The VizWiz is a recently proposed benchmark with images affected by natural corruptions. We use the classification split (built of 8,900 images taken by blind people labelled with 200 categories, *i.e.*, a subset of the ImageNet label set) to test, and the other images with corruption labels for training. We observe that competing data augmentation and TTA methods that work well for synthetic corruptions fail or bring small improvements under this setting. PAN outperforms all the approaches on every per-corruption classification metric defined in [17, 62]. Moreover, by analyzing the results obtained on each corruption separately, we can note that the ROT images are the hardest to be correctly classified, and PAN improves CA by 7.9%. A large improvement occurs on BRT images, where we can observe a change in lighting condition similar to *contrast* in the ImageNet-C dataset, where BN statistics were consistently shifted from the base ones (Fig. 3) implying a wider margin of improvement for PAN. Overall, the contribution of PAN is major when the statistics of the BN layers of the network are shifted with respect to the source ones (this is the case of varying light/weather conditions, rotations and digital transformations) and minor elsewhere (*e.g.*, on noisy or blurred samples). These results suggest that PAN contributes to increasing the reliability of robot vision models: i) indoor robots can promptly adapt to instantaneous light changes and continue their navigation safely, and ii) outdoor autonomous vehicles can be more reliable in adverse weather conditions.



(a) (b) (c)

Fig. 6. **a:** Statistics of normalization layers are different throughout the network (e.g. for ResNet18 on CIFAR10-C). y-axis denotes the estimated mean μ values and the variance σ^2 is depicted by shadowed region. On average, per-corruption statistics identified by our approach (**blue**) differ significantly from source ones (**red**). **b:** Change of CA for corrupted data for different normalization layers being adapted. We adapt all layers, one at a time adding the next layer progressively, starting from the first layer (**top**) or from the last layer (**bottom**). **c:** t-SNE of per-corruption features produced by $g(\cdot)$. Different corruption types are clearly separated. Clusters of representations of similar corruption types are closer to each other (*e.g.*, *Shot* and *Gaussian* noises; *Fog* and *Contrast*; *Motion* and *Defocus* blur).

C. Analyses and Ablation Studies

Per-corruption statistics estimated using our approach are significantly different from those found using the model pre-trained on source domain data. We have analyzed how these statistics vary according to the input corruption type in Fig. 2. Also, in Fig. 6a, we show that our method estimates diverse layer-wise normalization statistics with respect to the pre-trained source model. The highest divergence is shown in the first layers of the DNN model, as shown next.

Selective adaptation of statistics estimated at certain layers could be beneficial depending on the target hardware constraints. In this case, the most sensitive normalization parameters are those closer to the input, with the very first layer alone already accounting for 99.2% of the total accuracy gain between the pre-trained source model and our PAN, as shown in the top plot of Fig. 6b. Conversely, adapting only the last normalization layers, we do not obtain large gains until we include the initial normalization layers (bottom plot of Fig. 6b). This behaviour is expected as most of the variability on corrupted images affects the high-frequency components that are captured by the first layers of the network [19, 63]. On the other hand, we accounted for all layers as considering them brings a bigger performance growth.

CIM's performance is confirmed by Fig. 6c, showing that the module is able to learn highly distinguishable features relative to different corruption types. Well-clustered features imply that ground truth corruption types are more easily recognized at inference time. In terms of overhead, the CIM only adds a minimal 0.06 ms inference time per image (on ImageNet-C), increasing the overall inference time of PAN from 1.54 to 1.60 ms on an NVIDIA GeForce RTX 2080 Ti.

TTA performance depends on the CIM. Indeed, good performance of the CIM implies that each set of BN statistics is obtained performing TTA on a set of images, corrupted with the same corruption type. When the majority of images used to obtain $\Lambda_{\hat{\kappa}}^{\mathcal{T}}$ is corrupted with corruption type κ, we obtain a set of BN statistics $\Lambda_{\hat{\kappa}}^{\mathcal{T}}$ which results in being close to the set $\Lambda_{\kappa}^{\text{ref}}$. Hence, the normalization statistics identified by our *codebook* $\mathfrak{C}$ are close to the reference BN statistics $\Lambda_{\kappa}^{\text{ref}}$. This can be noticed in Fig. 2, where we can also appreciate that PAN's BN statistics (*i.e.*, $\Lambda_{\kappa}^{\mathcal{T}}$) are much closer to the per-corruption reference BN statistics compared to the pre-trained source ones (*i.e.*, $\Lambda^{\mathcal{S}}$).

V. Conclusion

In this paper, we investigated the robustness of vision models in challenging environments, where acquired images are subject to several types of quality degradation. Our evaluation on images with natural distortions exposes the limitations of existing approaches that have mostly focused on synthetically corrupted data, emphasising the need for solutions to improve model robustness in practical real-world scenarios. Our method (PAN) identifies the corruption present in the target sample and uses such information to adapt batch normalization layers of downstream vision models to enhance their resilience. PAN can be seamlessly plugged on top of any convolutional architecture employed for both indoor and outdoor robot systems, accomplishing many robot vision tasks (*e.g.*, object recognition, detection, semantic segmentation).

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[2] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do imagenet classifiers generalize to imagenet?," in *ICML*, 2019.

[3] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," `https://github.com/ultralytics/ultralytics`, 2023.

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[5] E. Camuffo, U. Michieli, and S. Milani, "Learning from mistakes: Self-regularizing hierarchical representations in point cloud semantic segmentation," *TMM*, 2023.

[6] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *ICLR*, 2019.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[8] D Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, et al., "The many faces of robustness: A critical analysis of out-of-distribution generalization," in *ICCV*, 2021.

[9] U. Michieli and M. Ozay, "Online continual learning for robust indoor object recognition," in *IROS*, 2023.

[10] Aakanksha and A. N. Rajagopalan, "Improving robustness of semantic segmentation to motion-blur using class-centric augmentation," in *CVPR*, 2023.

[11] F. Barbato, U. Michieli, M. K. Yucel, P. Zanuttigh, and M. Ozay, "A modular system for enhanced robustness of multimedia understanding networks via deep parametric estimation," in *ACM MMSys*, 2024.

[12] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel, "Benchmarking robustness in object detection: Autonomous driving when winter is coming," *ArXiv:1907.07484*, 2020.

[13] C. Sakaridis, D. Dai, and L. Van Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *ICCV*, 2019.

[14] C. Sakaridis, D. Dai, and L. Van Gool, "Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," *TPAMI*, 2020.

[15] C. Sakaridis, D. Dai, and L. Van Gool, "ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding," in *ICCV*, 2021.

[16] Y. P. Loh and C. S. Chan, "Getting to know low-light images with the exclusively dark dataset," *CVIU*, 2019.

[17] R. A. Bafghi and D. Gurari, "A new dataset based on images taken by blind people for testing the robustness of image classification models trained for imagenet categories," in *CVPR*, 2023.

[18] Q. She, F. Feng, X. Hao, Q. Yang, C. Lan, V. Lomonaco, X. Shi, Z. Wang, Y. Guo, Y. Zhang, F. Qiao, and R. H. M. Chan, "OpenLORIS-Object: A robotic vision dataset and benchmark for lifelong deep learning," in *ICRA*, 2020.

[19] M. K. Yucel, R. G. Cinbis, and P. Duygulu, "Hybridaugment++: Unified frequency spectra perturbations for model robustness," in *ICCV*, 2023.

[20] G. Chen, P. Peng, L. Ma, J. Li, L. Du, and Y. Tian, "Amplitude-phase recombination: Rethinking robustness of convolutional neural networks in frequency domain," in *ICCV*, 2021.

[21] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "AugMix: A simple data processing method to improve robustness and uncertainty," *ICLR*, 2020.

[22] T. Gong, J. Jeong, T. Kim, Y. Kim, J. Shin, and S.-J. Lee, "NOTE: Robust continual test-time adaptation against temporal correlation," in *NeurIPS*, 2022.

[23] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," in *ICLR*, 2021.

[24] S. Niu, J. Wu, Y. Zhang, Z. Wen, Y. Chen, P. Zhao, and M. Tan, "Towards stable test-time adaptation in dynamic wild world," in *ICLR*, 2023.

[25] J. Wang, M. R. U. Saputra, C. Xiaoxuan Lu, N. Trigoni, and A. Markham, "Rada: Robust adversarial data augmentation for camera localization in challenging conditions," in *IROS*, 2023.

[26] M. H. Haider, Z. Wang, A. Aman Khan, H. Ali, H. Zheng, S. Usman, R. Kumar, M. Usman Maqbool Bhutta, and P. Zhi, "Robust mobile robot navigation in cluttered environments based on hybrid adaptive neuro-fuzzy inference and sensor fusion," *Journal of King Saud University - Computer and Information Sciences*, 2022.

[27] Z. Chen, Z. Ding, J. M. Gregory, and L. Liu, "Ida: Informed domain adaptive semantic segmentation," in *IROS*, 2023.

[28] D. Hendrycks, A. Zou, M. Mazeika, L. Tang, B. Li, D. X. Song, and J. Steinhardt, "Pixmix: Dreamlike pictures comprehensively improve safety measures," *CVPR*, 2021.

[29] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *CVPR*, 2019.

[30] E. Rusak, L. Schott, R. S. Zimmermann, J. Bitterwolf, O. Bringmann, M. Bethge, and W. Brendel, "A simple way to make neural networks robust against diverse image corruptions," in *ECCV*, 2020.

[31] R. G. Lopes, D. Yin, B. Poole, J. Gilmer, and E. D. Cubuk, "Improving robustness without sacrificing accuracy with patch gaussian augmentation," *ArXiv:1906.02611*, 2019.

[32] H. Wang, C. Xiao, J. Kossaifi, Z. Yu, A. Anandkumar, and Z. Wang, "Augmax: Adversarial composition of random augmentations for robust training," in *NeurIPS*, 2021.

[33] E. Camuffo, U. Michieli, J J Moon, D Kim, and M Ozay, "Fft-based selection and optimization of statistics for robust recognition of severely corrupted images," in *ICASSP*, 2024.

[34] M. Boudiaf, R. Mueller, I. B. Ayed, and L. Bertinetto, "Parameter-free online test-time adaptation," in *CVPR*, 2022.

[35] W. Zhu, Y. Huang, D. Xu, Z. Qian, W. Fan, and X. Xie, "Test-time training for deformable multi-scale image registration," in *ICRA*, 2021.

[36] E. Yi and J. Kim, "Test-time synthetic-to-real adaptive depth estimation," in *ICRA*, 2023.

[37] A. Hatem, Y. Qian, and Y. Wang, "Test-time adaptation for point cloud upsampling using meta-learning," in *IROS*, 2023.

[38] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *ICML*, 2013.

[39] M. Zhang, S. Levine, and C. Finn, "Memo: Test time robustness via adaptation and augmentation," *NeurIPS*, 2022.

[40] A. Khurana, S. Paul, P. Rai, S. Biswas, and G. Aggarwal, "SITA: Single Image Test-time Adaptation," *CVPR*, 2023.

[41] E. Camuffo and S. Milani, "Continual learning for lidar semantic segmentation: Class-incremental and coarse-to-fine strategies on sparse data," in *CVPRW*, 2023.

[42] Q. Wang, O. Fink, L. Van Gool, and D. Dai, "Continual test-time domain adaptation," in *CVPR*, 2022.

[43] S. Niu, J. Wu, Y. Zhang, Y. Chen, S. Zheng, P. Zhao, and M. Tan, "Efficient test-time model adaptation without forgetting," in *ICML*, 2022.

[44] M. Mancini, H. Karaoguz, E. Ricci, P. Jensfelt, and B. Caputo, "Kitting in the wild through online domain adaptation," in *IROS*, 2018.

[45] Z. Nado, S. Padhy, D. Sculley, A. D'Amour, B. Lakshminarayanan, and J. Snoek, "Evaluating prediction-time batch normalization for robustness under covariate shift," *ArXiv:2006.10963*, 2020.

[46] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge, "Improving robustness against common corruptions by covariate shift adaptation," *NeurIPS*, 2020.

[47] D. Miller, N. Suenderhauf, M. Milford, and F. Dayoub, "Class anchor clustering: A loss for distance-based open set recognition," in *WACV*, 2021.

[48] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *NeurIPS*, 2016.

[49] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *University of Toronto*, 2009.

[50] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[51] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for mobilenetv3," in *ICCV*, 2019.

[52] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017.

[53] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv: 1605.07146*, 2017.

[54] C.-Y. Wang, H.-Y. Mark Liao, I.-H. Yeh, Y-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *CVPRW*, 2019.

[55] M. Weber, H. Wang, S. Qiao, J. Xie, M. D. Collins, Y. Zhu, L. Yuan, D. Kim, Q. Yu, D. Cremers, L. Leal-Taixe, A. L. Yuille, F. Schroff, H. Adam, and L.-C. Chen, "DeepLab2: A TensorFlow Library for Deep Labeling," *arXiv: 2106.09748*, 2021.

[56] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018.

[57] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, "Vizwiz grand challenge: Answering visual questions from blind people," *CVPR*, 2018.

[58] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer, "The many faces of robustness: A critical analysis of out-of-distribution generalization," in *ICCV*, 2021.

[59] R. Zhang, "Making convolutional networks shift-invariant again," in *ICML*, 2019.

[60] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., "Imagenet large scale visual recognition challenge," *IJCV*, 2015.

[61] "Torchvision: Pytorch's computer vision library," `https://github.com/pytorch/vision`, 2016.

[62] T.-Y. Chiu, Y. Zhao, and D. Gurari, "Assessing image quality issues for real-world problems," *CVPR*, 2020.

[63] Z. Sun, M. Ozay, and T. Okatani, "Improving Robustness of Feature Representations to Image Deformations using Powered Convolution in CNNs," in *CVPR*, 2017.