Self-Feature Distillation with Uncertainty Modeling for Degraded Image Recognition

Zhou Yang[1], Weisheng Dong[1(✉)], Xin Li[2], Jinjian Wu[1], Leida Li[1], and Guangming Shi[1]

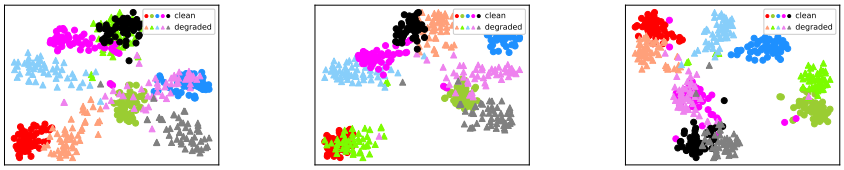
[1] School of Artificial Intelligence, Xidian University, Xi'an, China
yang_zhou@stu.xidian.edu.cn, {wsdong, jinjian.wu}@mail.xidian.edu.cn
{ldli, gmshi}@xidian.edu.cn
[2] Lane Dep. of CSEE, West Virginia University, Morgantown WV, USA
xin.li@mail.wvu.edu

Abstract. Despite the remarkable performance on high-quality (HQ) data, the accuracy of deep image recognition models degrades rapidly in the presence of low-quality (LQ) images. Both feature de-drifting and quality agnostic models have been developed to make the features extracted from degraded images closer to those of HQ images. In these methods, the l_2-norm is usually used as a constraint. It treats each pixel in the feature equally and may result in relatively poor reconstruction performance in some difficult regions. To address this issue, we propose a novel *self-feature distillation* method with *uncertainty modeling* for better producing HQ-like features from low-quality observations in this paper. Specifically, in a standard recognition model, we use the HQ features to distill the corresponding degraded ones and conduct uncertainty modeling according to the diversity of degradation sources to adaptively increase the weights of feature regions that are difficult to recover in the distillation loss. Experiments demonstrate that our method can extract HQ-like features better even when the inputs are degraded images, which makes the model more robust than other approaches.

Keywords: Robust image recognition, Self-feature distillation, Uncertainty modeling

1 Introduction

Despite rapid advances in deep learning [5,15,16,27,37,41,43], the impact of image degradation on visual recognition tasks has remained poorly understood. The good performance of deep models tested on HQ images of public datasets [8,29] often degrades dramatically in the presence of LQ images. Recent benchmark studies on the robustness of image classification [19], object detection [35], and semantic segmentation [23] models have shown that the performance of a standard neural network model is sensitive to image quality. For instance, vanilla ResNet50 [16] has a mean Corruption Error (mCE) up to 76.7% on ImageNet-C for image classification [19]; Faster-RCNN [37] with ResNet50 as the backbone



(a) Original model (b) QualNet[25] (c) Our method

Fig. 1. t-SNE feature distribution visualization on ImageNet-C validation set [19]. We trained the model on all classes in ImageNet-1K [8] but randomly selected five classes to show. We used the low-contrast degradation at severity level 3 to generate the degraded images. The colored symbols represent the feature vectors extracted from the corresponding images. Symbols with the same color are from the same class. Dot marks represent HQ features, and triangle marks indicate features from degrade images. The results of (c) show that our method can better gather the features both of degraded and high-quality images.

network has a mean Average Precision (mAP) of 18.2/36.3 on LQ/HQ images for object detection [35]; and DeepLabv3+ [5] only has a 6.6 mean Intersection over Union (mIoU) on shot noise images for semantic segmentation [23].

Naive approaches toward degraded image recognition attempt to restore corrupted images first. Indeed, various image restoration techniques including image denoising, deblurring, super-resolution, dehazing, and other image enhancement methods have been developed to improve the visual quality of degraded images. However, there is a fundamental difference between the *visual quality* and the *recognition quality* of an image - e.g., a photo with a masked face might have the highest visual quality, but its quality is deemed low under the context of face recognition. Various studies have confirmed that the improvement of visual perception can not guarantee a higher accuracy of subsequent high-level vision tasks [42,46]. Moreover, existing image restoration techniques are mostly devoted to a single type of degradation; how to restore an image from multiple-type degradation has remained an open challenge.

Current state-of-the-art in degraded image recognition tend to recognize directly from corrupted images based on statistical observations on the feature distribution in the latent space, as shown in Fig. 1. It has been found that shrinking the distribution distance between degraded/LQ features and original/HQ features is an effective way to improve the robustness of image recognition models. In recent work [46], a Feature De-drifting Module (FDM) was proposed to correct shallow pretrained layer's drifted feature response outputs. The basic idea behind FDM is to transform the task of degraded image restoration into feature-based reconstruction by deep degradation prior (DDP). Along this line of research, QualNet [25] attempted to produce HQ-like features from any LQ image via an invertible neural network [2]. Inspired by the success of knowledge distillation in network compression [6,21], we propose a approach of distilling

knowledge in the feature space, it can help the model learn the HQ-like feature so that improving the performance on corrupted images.

Another important new insight brought by this work is to recognize the *uncertainty* with the modeling of the degradation process [24]. In previous works, the estimation of HQ-like features has been deterministic, i.e., most of these methods adopted the common *MSE* loss which treats the distribution (variance) of features as a definite constant, leading to poor generalization property when the assumption of degradation process varies. To explicitly address such issue with degradation modeling (e.g., for images containing multiple-type degradation), we propose to design a new branch of a standard deep neural network for estimating the uncertainty (variance) of the feature distribution, which makes the model learn HQ-like features better. In summary, the contributions of this paper are listed as follows:

- We model the problem of degraded image recognition and propose a novel self-feature distillation approach, which can be easily applied into any recognition network and improve the performance of the model on the degraded images.
- We model the uncertainty of the various degraded features and transform the common deterministic estimation model into probabilistic uncertainty estimation. Specifically, a devoted branch, named uncertainty estimation module (UEM), is added to the network to estimate the uncertainty of the feature distribution (variance).
- Extensive experimental results on popular benchmark datasets show that our method performs much better in recognition task under multiple types of degradation than several current state-of-the-art methods.

2 Related Works

2.1 Degraded Image Recognition

Many visual recognition tasks have achieved good performance on HQ data, even better than humans. However, in some common degradation conditions, such as noise, blur, low contrast, rain and snow, the performance of deep convolutional neural networks (DCNNs) will be greatly reduced. [45] revealed the performance degradation of standard DCNNs in the case of blurred image, and [9,10,13] showed that DCNNs are not as good as humans in the recognition tasks on distorted images. To evaluate the robustness of DCNNs, a common corruption dataset, namely, ImageNet-C, was introduced in [19] which consists of 19 corruption types. Recently, researches on robust recognition of corrupted images can be roughly divided into the following methods:

Naive Data Augmentation. Data augmentation is a simple and effective way to make the model see more augmented images, to have better generalization performance during the inference time. The first method using Reinforcement Learning to search for the optimal data augmentation strategy is AutoAugment

[7]. AugMix [20] utilizes Jensen-Shannon Divergence consistency loss, and a formulation to mix multiple augmented images. [31] adds noise to randomly selected patches in an input image. DeepAugment [18] introduces four new real-world distribution shift datasets. However, as described in [25], deep models are inclined to learn an average data distribution when using a naive data augmentation method for multiple degradation types.

Image Restoration with Recognition. Conventional methods tend to fix the recognition network parameters but focus on restoring images from the degraded ones to perform better. But [36] indicated that only using dehazing methods is of little help or even harmful to improving the performance of classification because there may still exist a distribution shift between the HQ image and the reconstructed image. Therefore, there exists some research on recognition-friendly restoration. Based on this conclusion, [30] and URIE [42] simultaneously considered image enhancement and recognition. Specifically, they used the joint loss of image restoration and classification.

Feature Reconstruction with Recognition. Some researchers approved that the essential reason for the decline of performance is the degradation of features. [25,44,46] turned to reconstructing degraded features. [44] proposed a Feature Super-Resolution Generative Adversarial Network(FSR-GAN) to produce high-resolution features from small size images and enhance the discriminatory ability of features. Deep Degradation Prior (DDP) [46] reconstructed shallow features in the network through a feature de-drifting module. QualNet [25] transformed the final feature map into an image domain by an invertible network [2] to solve the HQ-like feature. Compared to these methods, we also use HQ-degraded image pairs to train the network but focus on reducing the intra-class differences in the feature representation space and modeling the uncertainty of features under various degradation situations.

Test-Time Adaptation and Self Learning. Recently, there exist some methods aimed at facilitating robustness by test-time adaptation. BN-Adaptation [39] employed a simple recalculation of batch normalization statistics in the procedure of testing for improving robustness to data shift. Robust Pseudo-Labeling (RPL) [38] proposed an improved cross entropy loss function for test-time training to calculate the loss of predicted pseudo labels and model outputs. The pseudo labels are generated by the model itself and are employed while training the model, so it is called self-training/learning. Clearly, the above methods are time-consuming and need sufficient data for inference. Unlike these existing methods above, our approach achieves robustness without extra models and data, enjoying a better generalization property.

2.2 Uncertainty in Deep Learning

Uncertainty has been introduced into the regression task of machine learning for a long time [3,14]. Recently, modeling uncertainty in deep learning for various visual tasks has been proved to improve deep networks' performance and robustness effectively [4,12,17,24,28,40,48]. Two types of uncertainty models have been studied in the literature: one is called epistemic or model uncertainty, which

represents the uncertainty of model prediction; the other is aleatoric or data uncertainty, which characterizes the noise inherent in observation data. We focus on the latter (aleatoric/data uncertainty) in this work and model the uncertainty of the feature distribution for a variety of degraded images.(see Sec. 3.2).

3 Proposed Method

In this section, we first describe the background of robust recognition in Sec. 3.1, then introduce our uncertainty-based self-feature distillation paradigm, and discuss the modeling uncertainty in HQ-like feature estimation in Sec. 3.2. Finally, we present the proposed method and the training process in Sec. 3.3.

3.1 Problem Formulation

Generally speaking, the goal of an image recognition task is to obtain its label y from an HQ/ideal image $\boldsymbol{x}$. However, in real-world applications, due to various sources of degradation (e.g., noise interference, motion blur, and compression artifacts), we can only get the LQ/degraded image $\tilde{\boldsymbol{x}}$ instead of the HQ one. Therefore, the problem of robust or degraded image recognition is to recognize the correct class label y from the LQ observation $\tilde{\boldsymbol{x}}$.

Several prior works [25,44,46] have shown that degraded features result in significant recognition performance degradation. We also did a simple visualization of features extracted from HQ and degraded images. Fig.1(a) shows the t-SNE [33] feature embedding visualization on ImageNet-C validation set. It suggests that the feature distributions of the same class (marked by the same color) stay close in the case of HQ images (marked by dots); but become separated from each other in the presence of degradation (marked by triangles). Moreover, the separation patterns will vary from dataset to dataset. For the reason of tractability, we do not consider the issue of domain shift [32] in this paper.

This above observation inspires us to pursue a model capable of performing well under multiple types of degradation by jointly restoring the features $\boldsymbol{z}$ and estimating the label y *simultaneously*. Let $\boldsymbol{z}, \tilde{\boldsymbol{z}}$ denote the corresponding HQ and LQ features of $\boldsymbol{x}, \tilde{\boldsymbol{x}}$ respectively, we can model the estimation of y and $\boldsymbol{z}$ as a maximum a posteriori probability (MAP) estimation framework

$$\operatorname{argmax} p(\boldsymbol{z}, y \mid \tilde{\boldsymbol{x}}) = \operatorname{argmax} p(y \mid \boldsymbol{z}, \tilde{\boldsymbol{x}}) p(\boldsymbol{z} \mid \tilde{\boldsymbol{x}}), \tag{1}$$

where we have used the Bayesian formula to translate the original problem into two subproblems: image recognition $p(y \mid \boldsymbol{z}, \tilde{\boldsymbol{x}})$ and feature reconstruction $p(\boldsymbol{z} \mid \tilde{\boldsymbol{x}})$.

We propose to use a deep learning method to solve this problem. The robust classifier (parameterized by $\boldsymbol{\Theta_1}$) can be represented as $f(\cdot; \boldsymbol{\Theta_1})$, which is expected to map the input degraded image $\tilde{\boldsymbol{x}}$ to the correct class y. $g(\cdot; \boldsymbol{\Theta_2})$ denotes the backbone network (parameterized by $\boldsymbol{\Theta_2}$) in the classifier which can reconstruct the HQ-like feature denoted by $\hat{\boldsymbol{z}}$ from $\tilde{\boldsymbol{x}}$, i.e, $\hat{\boldsymbol{z}} = g(\tilde{\boldsymbol{x}}; \boldsymbol{\Theta_2})$.

For the term $p(y \mid \boldsymbol{z}, \tilde{\boldsymbol{x}})$ in Eq. (1), since the HQ/ideal feature $\boldsymbol{z}$ is unavailable during test time, we use the HQ-like feature $\hat{\boldsymbol{z}}$ which is restored from the degraded images $\tilde{x}$ to approximate, i.e., $p(y \mid \boldsymbol{z}, \tilde{\boldsymbol{x}}) \approx p(y \mid \hat{\boldsymbol{z}}, \tilde{\boldsymbol{x}})$. Note that $\hat{\boldsymbol{z}} = g(\tilde{\boldsymbol{x}}; \boldsymbol{\Theta_2})$, so we have $p(y \mid \hat{\boldsymbol{z}}, \tilde{\boldsymbol{x}}) = p(y \mid \tilde{\boldsymbol{x}})$. Taking the logarithm of Eq. (1) and rewrite the formulation, we have

$$\log[p(\boldsymbol{z}, y \mid \tilde{\boldsymbol{x}})] \approx \log[p(y \mid \tilde{\boldsymbol{x}})] + \log[p(\boldsymbol{z} \mid \tilde{\boldsymbol{x}})]. \tag{2}$$

In this way, using deep learning to maximize the likelihood term $\log[p(\boldsymbol{z}, y|\tilde{\boldsymbol{x}})$ becomes the following objective function

$$(\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}) = \underset{\boldsymbol{\Theta_1}, \boldsymbol{\Theta_2}}{\operatorname{argmin}} L_1(y, f(\tilde{\boldsymbol{x}}; \boldsymbol{\Theta_1})) + L_2(\boldsymbol{z}, g(\tilde{\boldsymbol{x}}; \boldsymbol{\Theta_2})), \tag{3}$$

where L_1 is the loss function of classification, commonly using cross entropy loss. And L_2 loss aims at gathering features extracted from HQ and LQ images. In our experiments, we train the whole classifier by using multitask learning strategy [25,34]. To better optimize the joint loss function in Eq. (3) through deep neural networks, we present a novel self-feature distillation method with uncertainty learning next.

3.2 Self-Feature Distillation with Uncertainty Modeling

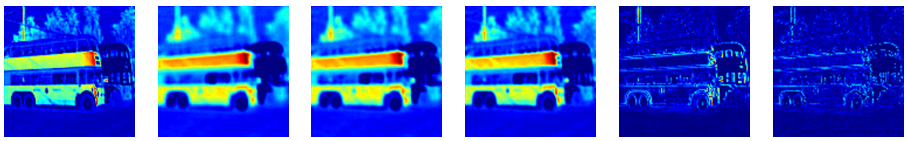
Based on the above discussions, the objective of the second term in Eq. (3) is to obtain the HQ-like feature $\hat{\boldsymbol{z}}$ from the degraded image $\tilde{\boldsymbol{x}}$. To achieve this goal, we propose a self-feature distillation framework for estimating HQ-like features. Specifically, as shown in Fig. 3, we employ a pre-trained model on HQ data as the baseline network. During training, both of the HQ images and the simulated degraded images are input into the backbone network to extract features, respectively. Through the feature distillation, their features ($\boldsymbol{z}$ and $\hat{\boldsymbol{z}}$) are expected to be close and have a more robust classification performance.

Due to multiple types of degradation and ill-posed nature of feature restoration problems, it is difficult to learn the HQ-like feature, especially in the texture or edge regions (see in Figs. 2(e) and 2(f)). The current state-of-the-art method QualNet [25] chose to transform features into the image domain by an invertible neural network [2]. We opt to tackle this problem from a different perspective: due to the diversity of degradation, data uncertainty often inevitably leads to feature uncertainty.

Assuming that each feature map extracted from the corresponding image observes a Gaussian distribution with mean $\hat{\boldsymbol{z}}_{\boldsymbol{i}}$ and standard deviation $\boldsymbol{\theta_i}$, to better quantify aleatoric/data uncertainty in feature reconstruction, we can formulate the observation model with the estimated HQ-like feature $\hat{\boldsymbol{z}}_{\boldsymbol{i}}$ and the target HQ feature $\boldsymbol{z_i}$ as a Gaussian likelihood function

$$\boldsymbol{z_i} = \hat{\boldsymbol{z}}_{\boldsymbol{i}} + \epsilon \boldsymbol{\theta_i}, \tag{4}$$

where ϵ denotes the normal distribution with zero-mean and unit-variance.



(a) HQ (b) LQ (c) DDP (d) Ours (e) Diff.DDP (f) Diff.Ours

Fig. 2. (a), (b): The HQ and LQ feature (size: 112×112) extracted from clean and Gaussian blur image respectively. (c), (d): The HQ-like feature reconstructed by DDP [46] and our method. (e), (f): The normalized absolute difference map between HQ and HQ-like feature. **Best viewed in color.**

Conventional feature distillation methods commonly use *MSE* loss for deterministic estimation. Obviously, the MSE loss can be interpreted as a Gaussian likelihood function with a constant variance in Eq. (4), assuming that the variance of the difference signals between the HQ-feature $\boldsymbol{z}_i$ and restored HQ-like feature $\hat{\boldsymbol{z}}_i$ are constant. However, as shown in Fig. 2(e) and Fig. 2(f), we can see the spatial variation of the difference map, implying that the variances in the texture and edge areas vary across the feature map. Therefore, the stationary assumption of the variances of the Gaussian likelihood function for each pixel in the feature map is invalid.

Instead of assuming a constant variance, we proposed to estimate the restored HQ-like feature $\hat{\boldsymbol{z}}_i$ and their uncertainty (i.e., the variances $\boldsymbol{\theta_i}$) simultaneously. For a given LQ image $\tilde{\boldsymbol{x}}_{\boldsymbol{i}}$, to restore the corresponding HQ feature $\boldsymbol{z_i}$, a Gaussian distribution is assumed for representing the likelihood function by

$$p(\boldsymbol{z_i} \,|\, \tilde{\boldsymbol{x}}_{\boldsymbol{i}}, \boldsymbol{\theta_i}) = \frac{1}{\sqrt{2\pi}\boldsymbol{\theta_i}} exp(-\frac{||\boldsymbol{z_i} - g(\tilde{\boldsymbol{x}}_{\boldsymbol{i}}; \boldsymbol{\Theta_2})||^2}{2{\boldsymbol{\theta_i}}^2}), \tag{5}$$

where $g(\tilde{\boldsymbol{x}}_{\boldsymbol{i}}; \boldsymbol{\Theta_2}) = \hat{\boldsymbol{z}}_{\boldsymbol{i}}$ denotes the HQ-like feature (mean) and $\boldsymbol{\theta_i}$ is the uncertainty (variance). Both of them are learned by DCNNs respectively.

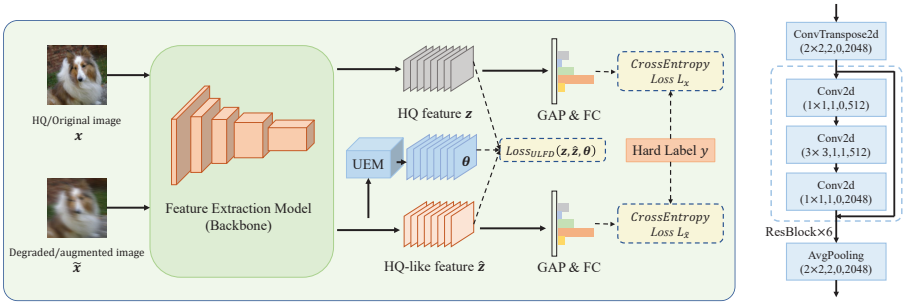
Based on the observation that the uncertainty $\boldsymbol{\theta}$ is generally sparse in the feature map, as shown in Fig. 2(e) and Fig. 2(f), we propose to impose **Jeffrey's prior** [11]: $p(w) \propto \frac{1}{w}$ on uncertainty estimation $\boldsymbol{\theta_i}$, which can be expressed as

$$\begin{aligned} p(\boldsymbol{z_i}, \boldsymbol{\theta_i} \,|\, \tilde{\boldsymbol{x}}_{\boldsymbol{i}}) &= p(\boldsymbol{z_i} \,|\, \tilde{\boldsymbol{x}}_{\boldsymbol{i}}, \boldsymbol{\theta_i}) p(\boldsymbol{\theta_i}) \\ &= \frac{1}{\sqrt{2\pi}\boldsymbol{\theta_i}} exp(-\frac{||\boldsymbol{z_i} - g(\tilde{\boldsymbol{x}}_{\boldsymbol{i}}; \boldsymbol{\Theta_2})||^2}{2{\boldsymbol{\theta_i}}^2}) \frac{1}{\boldsymbol{\theta_i}}. \end{aligned} \tag{6}$$

Then the log-likelihood function with Jeffrey's prior can be formulated as follows

$$\log p(\boldsymbol{z_i}, \boldsymbol{\theta_i} \,|\, \tilde{\boldsymbol{x}}_{\boldsymbol{i}}) = -\frac{||\boldsymbol{z_i} - g(\tilde{\boldsymbol{x}}_{\boldsymbol{i}}; \boldsymbol{\Theta_2})||^2}{2{\boldsymbol{\theta_i}}^2} - \log {\boldsymbol{\theta_i}}^2. \tag{7}$$

To implement the above idea, we add a new branch (UEM), as highlighted by the blue color in Fig. 3(a), at the end of the backbone network to estimate the uncertainty. It follows that the problem of maximum-likelihood estimation



(a) The proposed self-feature distillation with uncertainty learning method. (b) UEM

Fig. 3. System overview. (a) Both of the HQ features $\boldsymbol{z}$ and the estimated HQ-like features $\hat{\boldsymbol{z}}$ are extracted from backbone network. The HQ-like features are input into the uncertainty estimation branch to estimate the uncertainty (variance) $\boldsymbol{\theta}$ of the feature distribution in a variety of degradation. (b) The architecture of our uncertainty estimation module. The numbers in parentheses represent the kernel size, stride, padding, and output channels respectively. Note that our UEM represents a clever use of ResNet [16] for variance/uncertainty estimation.

in Eq. (7) can be translated into the following uncertainty learning-based feature distillation (ULFD) loss function,

$$L_{ULFD} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{||\boldsymbol{z_i} - g(\tilde{\boldsymbol{x_i}}; \boldsymbol{\Theta_2})||^2}{2{\boldsymbol{\theta_i}}^2} + \log \boldsymbol{\theta_i}^2\right), \tag{8}$$

where N is the number of samples in a minibatch of training dataset. As both the HQ image and the corresponding degraded image are input into the backbone network, we can obtain the HQ feature $\boldsymbol{z_i}$ and restored HQ-like features $\hat{\boldsymbol{z}}_{\boldsymbol{i}} = g(\tilde{\boldsymbol{x}}_{\boldsymbol{i}}; \boldsymbol{\Theta_2})$ with estimated uncertainty $\boldsymbol{\theta_i}$ through the uncertainty loss function.

Apparently, the learned variances $\boldsymbol{\theta_i}$ can be regarded as a confidence score measuring the closeness between the restored HQ-like feature $\hat{\boldsymbol{z}}_i$ and HQ feature $\boldsymbol{z}_i$. For those $\hat{\boldsymbol{z}}_i$ far away from $\boldsymbol{z}_i$, the network will estimate larger variances to reduce the error term $\frac{||\boldsymbol{z_i} - \hat{\boldsymbol{z}}_i||^2}{\boldsymbol{\theta}_i^2}$, instead of overfitting to those erroneous regions. When $\hat{\boldsymbol{z}}_{\boldsymbol{i}}$ is easy to learn, the second term $\log \boldsymbol{\theta_i}^2$ plays a major role in loss function, and the network tends to make $\boldsymbol{\theta_i}$ smaller. It plays a role similar to the attention mechanism, enabling the network to focus on the hard samples [22] in the training set.

3.3 Architecture and Training Strategy

The overall flowchart of our method is shown in Fig. 3(a). Note that we attempt to reconstruct the deep semantic feature rather than the features in the shallow layer (DDP [46]) because it has proved to be more helpful in improving the

accuracy of the classifier (see Sec. 4.5 for details). Therefore, we use a well-pretrained standard deep neural network on high-quality images as the baseline model and add an uncertainty estimation module (UEM) at the end of the backbone network to estimate the variance.

The detailed design of the UEM is shown in Fig. 3(b). Since the final feature map has a lower resolution, we first introduce a transposed convolution layer to expand the spatial resolution, which is similar to the role of a decoder. Then six residual blocks are used to learn the uncertainty (variance). Similar to the bottleneck architecture mentioned in [16], we use 1×1 convolution in the residual block to reduce the parameters and the dimension of the final feature maps. Finally, an average pooling layer is used to keep the output dimension consistent with the original feature dimension. To stabilize the training, we estimate $\boldsymbol{\sigma_i} = \log \boldsymbol{\theta_i}^2$ in this branch. So the uncertainty learning-based feature distillation loss function in Eq. (8) can be reformulated as

$$L_{ULFD} = \frac{1}{N}\sum_{i=1}^{N}(exp(-\boldsymbol{\sigma_i})||\boldsymbol{z_i} - g(\tilde{\boldsymbol{x}}_{\boldsymbol{i}};\boldsymbol{\Theta_2})||^2 + 2\boldsymbol{\sigma_i}). \tag{9}$$

To sum it up, We train our network by a multitask learning strategy [25] with the joint loss function of uncertainty and recognition in Sec. 3.1 and Sec. 3.2 as

$$\begin{aligned} L = &\frac{1}{N}\sum_{i=1}^{N} L_{CE}(y_i, f(\boldsymbol{x_i};\boldsymbol{\Theta_1})) + L_{CE}(y_i, f(\tilde{\boldsymbol{x}}_{\boldsymbol{i}};\boldsymbol{\Theta_1})) \\ &+ \lambda\cdot\frac{1}{N}\sum_{i=1}^{N}[exp(-\boldsymbol{\sigma_i})||\boldsymbol{z_i} - g(\tilde{\boldsymbol{x}}_{\boldsymbol{i}};\boldsymbol{\Theta_2})||^2 + 2\boldsymbol{\sigma_i}], \end{aligned} \tag{10}$$

where L_{CE} represents the *Cross-Entropy* loss and λ is the hyperparameter.

4 Experiments

Simulations and dataset. In our experiments, we have simulated the corruption described in the common dataset ImageNet-C [19] to generate the degraded images and evaluated the model's robustness. ImageNet-C contains 15 corruption types (Gaussian/shot/impulse noise, glass/motion/defocus/zoom blur, contrast, elastic, JPEG, pixelate, frost, fog, snow, and brightness) in 4 categories for training and 4 corruption types (speckle noise, Gaussian blur, spatter, and saturate) as holdout corruptions. Every corruption consists of 5 severity levels. To measure the performance of the network under these degradation conditions, the mean Corruption Error (mCE) [19] is a commonly used metric. All the mCE results in our experiments were normalized.

Training setting. Every training image pair in most of our experiments contains an original clean and a corresponding degraded image generated by a uniformly sampled type from the 15 corruption types mentioned above. We trained several architectures such as ResNet50 [16] and ResNeXt101[47] with ImageNet-1K [8], because they are commonly used in recognition tasks. We employed Adam

[26] as the optimizer with initial learning rate 0.001, and it was divided by 10 after 5k, 12.5k, 25k iterations. Our model was trained for 40k iterations (about 10 epochs) with batch size of 256 per iteration. The hyperparameter λ in Eq. (10) was set to 0.1 according to the ablation study described in Sec. 4.5.

4.1 Comparison with Sate-of-the-art Methods

To demonstrate the effectiveness of our method on the degraded image domain generalization, we have compared our method with the state-of-the-art methods on ImageNet-C, such as DDP [46], URIE [42], KD VID [1], and QualNet[25]. The experimental setup was consistent with those described in the relevant papers. Through careful experiments, we reproduced the results similar to those in their paper. In the proposed self-feature distillation network, the uncertainty estimation module (UEM) is used to improve the robustness of the model in a variety of degradation. To demonstrate the effectiveness of our UEM, we modify the network into a deterministic model by removing the UEM branch and use the common MSE loss for training.

Table 1 shows that our approach performs better than these related works on the ImageNet-C test set. *HQ, seen, unseen* represent the top-1 classification accuracy on HQ images, 15 types of corrupted images which are seen in the training set, and 4 unknown types of corrupted images during training, respectively. *Ours w/o UEM* means the deterministic version of our method. We use two types of classification neural networks, ResNet50 [16] and ResNeXt101-32x8d [47]. Table 2 shows the detailed performance for model robustness in four degradation cases which are unknown in training. It is worth noting that our method has less performance degradation on clean images and is more robust than other methods, from which we can verify the superiority of our method.

We have also compared feature distribution visualization results between our method and the *SOTA* method QualNet. Specifically, we randomly selected five classes of images in ImageNet-1K [8] validation set and used the low contrast degradation method in [19] with severity level 3 to generate corresponding corrupted ones. Both of them were input into the well-trained classifier in turn, and their logits were extracted for t-SNE visualization. Figs. 1(b) and 1(c) show that our method can better gather the features of both LQ and HQ images, and thus can improve the robustness of the classifier on degraded images.

4.2 Robustness of Using Naive Augmented Data

Naive data augmentation is a technique that synthesizes augmented images from the original ones and then trains the network with the original and augmented images, which are expected to improve the recognition accuracy and model robustness. The main difference between our proposed framework and the naive augmentation training is that we add self-feature distillation operation and uncertainty estimation branch. Therefore, our method can be easily adopted in naive data augmentation training.

Table 1. The top-1 accuracy on HQ ImageNet-1K[8] validation set, 15 types seen corrupted and 4 types unseen corrupted images in ImageNet-C[19] validation set. Each corruption type contains 5 severity levels. The mean Corruption Error(mCE) is the normalized average error rate at all severity levels of the 15 known corruptions (less is better). "Ours w/o UEM" means the UEM branch is removed and trained using MSE loss. **The best results are in bold.**

Methods	Architecture	HQ ↑	Seen ↑	Unseen ↑	mCE ↓
Vanilla [16]	ResNet50	**76.82%**	39.17%	47.11%	76.5%
DDP [46]		72.15%	48.21%	50.73%	62.78%
URIE [42]		73.80%	55.10%	56.50%	55.70%
KD VID [1]		74.85%	-	-	51.29%
QualNet50 [25]		75.43%	61.08%	58.10%	50.34%
Ours w/o UEM		75.81%	61.65%	60.23%	49.50%
Ours		76.23%	**63.44%**	**62.90%**	**46.37%**
Vanilla [47]	ResNeXt101	**79.68%**	47.08%	55.53%	69.76%
QualNet101 [25]		77.81%	65.47%	63.28%	42.61%
Ours w/o UEM		78.35%	66.81%	65.30%	41.23%
Ours		79.04%	**69.16%**	**67.83%**	**39.50%**

Table 2. Top-1 accuracy on 4 unseen corruptions in ImageNet-C [19] validation set. **The best results are in bold.**

Methods	Architecture	Top-1 Accuracy ↑			
		Speckle-Noise	Gaussian-Blur	Spatter	Saturate
Vanilla [16]	ResNet50	35.49%	49.16%	41.87%	61.92%
QualNet50 [25]		63.50%	52.59%	54.56%	61.75%
Ours w/o UEM		65.25%	55.39%	56.33%	63.95%
Ours		**66.44%**	**58.59%**	**58.65%**	**67.92%**
Vanilla [47]	ResNeXt101	47.92%	57.94%	48.72%	67.52%
QualNeXt101 [25]		64.21%	57.24%	62.48%	69.19%
Ours w/o UEM		68.70%	61.25%	60.37%	70.86%
Ours		**71.23%**	**64.87%**	**63.04%**	**72.18%**

In this experiment, the input image pair contains an original clean image and the corresponding augmented one generated by three popular data augmentation methods - i.e., Augmix [20], DeepAugment [18] and DeepAugment+Augmix [18], instead of the simulated degraded images of 15 corruption types. Table 3 shows the results of combining our framework with augmented data compared to the naive methods. Through adding self-feature distillation with uncertainty estimation, and jointly training the whole network, the model can indeed increase the clean accuracy and robustness.

4.3 Comparison with Test-Time Adaptation Methods

As the test-time adaptation methods described in Sec. 2.1 require many test images, it is unrealistic in practical applications. To explore the impact of insufficient samples in the test set on those methods, we constructed a tiny subset by randomly selecting 500 images from 50000 images in ImageNet validation set for testing. For BNAdapt [31], the batch size of the test set changed from 256

Table 3. Top-1 accuracy on HQ images and mCE on ImageNet-C validation set for other data augmentation methods. All methods are based on ResNet50 architecture. For Augmix [20], DeepAugment [18] and DeepAugment+Augmix [18], we choose to retrain the network to make a fair comparison. "*+Ours*" means we use the corresponded augmented images in our framework to train the model.

Methods	Top-1 Accuracy on HQ ↑	mCE ↓
DeepAugment[18]	74.60%	60.31%
DeepAugment+**Ours**	**74.83%**	**59.04%**
Augmix[20]	75.38%	65.30%
Augmix+**Ours**	**75.40%**	**64.37%**
DeepAugment+Augmix[18]	73.64%	53.50%
DeepAugment+Augmix+**Ours**	**73.81%**	**52.47%**

to 32. We trained the ResNet50 model with augmented images described in Sec. 4.2 for a fair comparison.

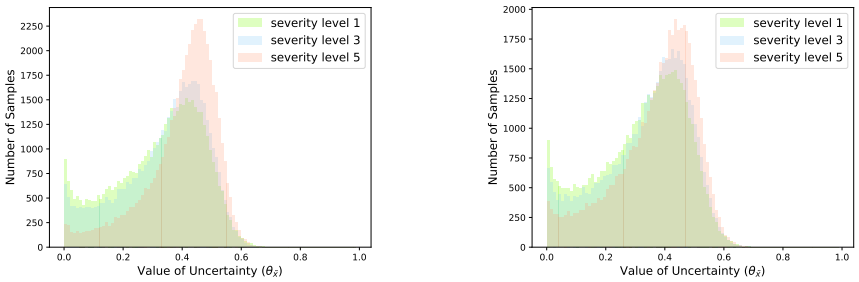
Table 4 demonstrates the average top-1 accuracy performance of test-time adaptation methods plummet when the number of test samples decreases. Clearly, test-time adaptation is time-consuming and needs to be trained separately on each corrupted type in ImageNet-C validation set. In contrast, our method neither needs to use the test set for training nor introduces any extra computational cost during inference. From the experimental results, we have also found that when the severity of degradation increases, the recognition accuracy of the self-learning method will be worse and worse due to the unreliable pseudo labels.

4.4 Contributions of Uncertainty Learning

To illustrate how uncertainty learning works for each image $\tilde{\boldsymbol{x}}$, we averaged the learned feature uncertainty (variance) map $\boldsymbol{\theta}$ in the spatial and channel dimensions (i.e., $\theta_{\tilde{x}} = \frac{1}{CHW}\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{w=1}^{W}\boldsymbol{\theta}_{c,h,w}$), which can represent the uncertainty of this sample. We calculated the uncertainty of each sample in five severity levels in ImageNet-C [19]. As shown in Fig. 4(a), the estimated uncertainty $\theta_{\tilde{x}}$ is closely related to the severity levels of corruption. The more serious the image degradation is, the more samples are difficult to recognize,

Table 4. Comparing our method with test-time adaptation methods when the number of samples in the test set changes. *Original set* and *Subset* represent the average top-1 accuracy on original ImageNet-C validation set and the constructed subset, respectively.

Methods	Original set	Subset
Vanilla[16]	39.2%	42.3%
DeepAugment+Augmix[18]	58.1%	61.4%
DeepAugment+Augmix+BNAdapt[31]	65.7%	60.2%
DeepAugment+Augmix+RPL[38]	**67%**	62.1%
DeepAugment+Augmix+**Ours**	59.3%	**62.7%**



(a) spatter(62.59%) (b) motion-blur(68.08%)

Fig. 4. The predicted uncertainty value $\theta_{\tilde{x}}$ of each degraded image in different severity levels of spatter and motion-blur degradation. The ordinate represents the number of samples corresponding to the value of uncertainty. For better view, we only selected severity levels of 1, 3 and 5. The number in parentheses indicates the top-1 accuracy on the corresponding degradation. **Best viewed in color.**

and the larger the corresponding $\theta_{\tilde{x}}$ value is. This is also similarly observed in DUL[4]. Comparing Figs. 4(a) and 4(b), we can observe that the recognition accuracy of the model for spatter degradation is lower than that of motion blur, and the corresponding number of samples with large uncertainty value (hard samples) is more. These experimental results support, as described in Sec. 3.2, that the uncertainty measures the difficulty of HQ-like features reconstruction for recognition. It makes the classifier pay close attention to the hard samples so that it can improve the performance on corrupted data with high severity level.

4.5 Ablation Study

In this section, we first discuss the choice of shallow features or deep semantic features for reconstruction. Through comparative experiments, we find that the top-1 accuracy on corrupted data by shallow feature reconstruction is 4.5 % lower than that of deep semantic features. Therefore, we choose to restore deep semantic features in our method. Then we conducted several ablation studies to investigated which modules significantly contribute to performance improvement. In our experiments, We roughly divide our method in two modules: *self-feature distillation, uncertainty estimation* and verify their impact separately.

Without adding any modules means, we use degraded images to fine-tune the model and only optimize it through cross entropy loss. Just adding the self-feature distillation module denotes the deterministic version as described in Sec. 4.1. Adding both self-feature distillation and uncertainty estimation modules represents the proposed method in Fig. 3(a) where we simultaneously learn HQ-like features (mean) and its uncertainty(variance) through joint loss function in Eq. (10). All models are trained and tested under the ResNet50 architecture. We use 15 types of corruptions in [19] for training. The results are shown in Tab. 5.

Table 5. Ablation study on our proposed module. "Clean" and "mCE" indicate the top-1 clean accuracy (%) and mean Corruption Error on 15 corruption types, respectively.

Self-feature distillation		✓	✓
Uncertainty modeling			✓
Clean↑	75.11%	75.81%	**76.23%**
mCE↓	51.31%	49.50%	**46.37%**

c

Table 6. The hyperparameter λ and different type of uncertainty in our method.

(a) Top-1 accuracy on ImageNet-C validation set of different λ.

λ	0	0.01	0.1	1
Top-1 Acc.↑	58.43%	62.37%	**63.44%**	62.78%

(b) The choice of estimating sample-wise or spatial-wise uncertainty.

sample-wise	mCE: 46.93%
spatial (Ours)	**mCE: 46.37%**

We have also studied the impact on the recognition accuracy of the hyperparameter value λ. We empirically select four values for training, respectively. The results are shown in Table 6(a). Based on the above results, we finally choose $\lambda = 0.1$. We also compared the mCE value of estimating a sample-wise (with dimensions of $B \times 1 \times 1 \times 1$) and spatial-wise (ours) uncertainty. The results are shown in Table 6(b) (lower is better). We can see that the performance of the sample-wise uncertainty is slightly weaker than the spatial uncertainty that we adopted. This is because the sample-wise uncertainty cannot focus on the difficult regions in the feature, resulting in slightly inferior feature detail recovery.

5 Conclusion

This paper has presented a new paradigm dedicated to making recognition models perform better in the presence of various corruptions. Through self-feature distillation with uncertainty learning, our method is capable of gathering both clean and distorted features, so that the model improves the recognition robustness effectively. The advantages of our method have been verified throughout experiments in various settings. We hope that our method can be extended to other recognition applications with low-quality/degraded images.

Acknowledgement: This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0101400 and the Natural Science Foundation of China under Grant 61991451, Grant 61632019, Grant 61621005, and Grant 61836008.

References

1. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9163–9171 (2019) 10, 11
2. Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E.W., Klessen, R.S., Maier-Hein, L., Rother, C., Köthe, U.: Analyzing inverse problems with invertible neural networks. arXiv preprint arXiv:1808.04730 (2018) 2, 4, 6
3. Bishop, C.M., Qazaz, C.S.: Regression with input-dependent noise: A bayesian treatment. Advances in neural information processing systems pp. 347–353 (1997) 4
4. Chang, J., Lan, Z., Cheng, C., Wei, Y.: Data uncertainty learning in face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5710–5719 (2020) 4, 13
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018) 1, 2
6. Cho, J.H., Hariharan, B.: On the efficacy of knowledge distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4794–4802 (2019) 2
7. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501 (2018) 4
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 1, 2, 9, 10, 11
9. Dodge, S., Karam, L.: Understanding how image quality affects deep neural networks. In: 2016 eighth international conference on quality of multimedia experience (QoMEX). pp. 1–6. IEEE (2016) 3
10. Dodge, S., Karam, L.: A study and comparison of human and deep learning recognition performance under visual distortions. In: 2017 26th international conference on computer communication and networks (ICCCN). pp. 1–7. IEEE (2017) 3
11. Figueiredo, M.A.: Adaptive sparseness using jeffreys prior. In: NIPS. pp. 697–704 (2001) 7
12. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Insights and applications. In: Deep Learning Workshop, ICML. vol. 1, p. 2 (2015) 4
13. Geirhos, R., Temme, C.R.M., Rauber, J., Schütt, H.H., Bethge, M., Wichmann, F.A.: Generalisation in humans and deep neural networks. arXiv preprint arXiv:1808.08750 (2018) 3
14. Goldberg, P.W., Williams, C.K., Bishop, C.M.: Regression with input-dependent noise: A gaussian process treatment. Advances in neural information processing systems **10**, 493–499 (1997) 4
15. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017) 1
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 1, 8, 9, 10, 11, 12
17. He, Y., Zhu, C., Wang, J., Savvides, M., Zhang, X.: Bounding box regression with uncertainty for accurate object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2888–2897 (2019) 4

18. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8349 (2021) 4, 11, 12
19. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:1903.12261 (2019) 1, 2, 3, 9, 10, 11, 12, 13
20. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. arXiv preprint arXiv:1912.02781 (2019) 4, 11, 12
21. Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., Choi, J.Y.: A comprehensive overhaul of feature distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1921–1930 (2019) 2
22. Huang, Y., Shen, P., Tai, Y., Li, S., Liu, X., Li, J., Huang, F., Ji, R.: Improving face recognition from hard samples via distribution distillation loss. In: European Conference on Computer Vision. pp. 138–154. Springer (2020) 8
23. Kamann, C., Rother, C.: Benchmarking the robustness of semantic segmentation models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8828–8838 (2020) 1, 2
24. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? arXiv preprint arXiv:1703.04977 (2017) 3, 4
25. Kim, I., Han, S., Baek, J.w., Park, S.J., Han, J.J., Shin, J.: Quality-agnostic image recognition via invertible decoder. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12257–12266 (2021) 2, 4, 5, 6, 9, 10, 11
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 10
27. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012) 1
28. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. arXiv preprint arXiv:1612.01474 (2016) 4
29. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014) 1
30. Liu, D., Wen, B., Liu, X., Wang, Z., Huang, T.S.: When image denoising meets high-level vision tasks: A deep learning approach. arXiv preprint arXiv:1706.04284 (2017) 4
31. Lopes, R.G., Yin, D., Poole, B., Gilmer, J., Cubuk, E.D.: Improving robustness without sacrificing accuracy with patch gaussian augmentation. arXiv preprint arXiv:1906.02611 (2019) 4, 11, 12
32. Luo, Y., Zheng, L., Guan, T., Yu, J., Yang, Y.: Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2507–2516 (2019) 5
33. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008) 5
34. Mao, C., Gupta, A., Nitin, V., Ray, B., Song, S., Yang, J., Vondrick, C.: Multitask learning strengthens adversarial robustness. In: Computer Vision–ECCV 2020: 16th European Conference. pp. 158–174. Springer (2020) 6

35. Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A.S., Bethge, M., Brendel, W.: Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv preprint arXiv:1907.07484 (2019) 1, 2
36. Pei, Y., Huang, Y., Zou, Q., Lu, Y., Wang, S.: Does haze removal help cnn-based image classification? In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 682–697 (2018) 4
37. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28**, 91–99 (2015) 1
38. Rusak, E., Schneider, S., Gehler, P., Bringmann, O., Brendel, W., Bethge, M.: Adapting imagenet-scale models to complex distribution shifts with self-learning. arXiv preprint arXiv:2104.12928 (2021) 4, 12
39. Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., Bethge, M.: Improving robustness against common corruptions by covariate shift adaptation. Advances in Neural Information Processing Systems **33** (2020) 4
40. Shi, Y., Jain, A.K.: Probabilistic face embeddings. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6902–6911 (2019) 4
41. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 1
42. Son, T., Kang, J., Kim, N., Cho, S., Kwak, S.: Urie: Universal image enhancement for visual recognition in the wild. In: European Conference on Computer Vision. pp. 749–765. Springer (2020) 2, 4, 10, 11
43. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10781–10790 (2020) 1
44. Tan, W., Yan, B., Bare, B.: Feature super-resolution: Make machine see more clearly. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3994–4002 (2018) 4, 5
45. Vasiljevic, I., Chakrabarti, A., Shakhnarovich, G.: Examining the impact of blur on recognition by convolutional networks. arXiv preprint arXiv:1611.05760 (2016) 3
46. Wang, Y., Cao, Y., Zha, Z.J., Zhang, J., Xiong, Z.: Deep degradation prior for low-quality image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11049–11058 (2020) 2, 4, 5, 7, 8, 10, 11
47. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1492–1500 (2017) 9, 10, 11
48. Zafar, U., Ghafoor, M., Zia, T., Ahmed, G., Latif, A., Malik, K.R., Sharif, A.M.: Face recognition with bayesian convolutional networks for robust surveillance systems. EURASIP Journal on Image and Video Processing **2019**(1), 1–10 (2019) 4