

Restoring Spatially-Heterogeneous Distortions using Mixture of Experts Network

Sijin Kim^{1,*}, Namhyuk Ahn^{1,2,*}, and Kyung-Ah Sohn^{1,2,†}

¹ Department of Computer Engineering, Ajou University

² Department of Artificial Intelligence, Ajou University
tlwlsdi0306@gmail.com, {aa0dfg, kasohn}@ajou.ac.kr

Abstract. In recent years, deep learning-based methods have been successfully applied to the image distortion restoration tasks. However, scenarios that assume a single distortion only may not be suitable for many real-world applications. To deal with such cases, some studies have proposed sequentially combined distortions datasets. Viewing in a different point of combining, we introduce a spatially-heterogeneous distortion dataset in which multiple corruptions are applied to the different locations of each image. In addition, we also propose a mixture of experts network to effectively restore a multi-distortion image. Motivated by the multi-task learning, we design our network to have multiple paths that learn both common and distortion-specific representations. Our model is effective for restoring real-world distortions and we experimentally verify that our method outperforms other models designed to manage both single distortion and multiple distortions.

1 Introduction

The image restoration task is a classic and fundamental problem in the computer vision field. It aims to generate a visually clean image from a corrupted observation. There exist various problems related to the image restoration such as super-resolution, denoising, and deblurring. For all such tasks, clean and distorted images are many-to-one mapping, so it is very challenging to develop an effective restoration algorithm. Despite the difficulties, image restoration has been actively investigated because it can be applied to various scenarios.

Recently, the performance of the image restoration methods has been significantly improved since the use of a deep learning-based approach. For example, in the super-resolution task, various methods [1–5] progressively push the performance by stacking more layers and designing novel modules. Likewise, other restoration tasks such as denoising [6–10] and deblurring [11–14] also enjoy the huge performance leap. However, most of the deep restoration models assume that the image is corrupted by a single distortion only (Figure 1b), which may not be suitable for the real scenarios. In real-world applications, there can be mixed

* indicates equal contribution.

† indicates corresponding author.

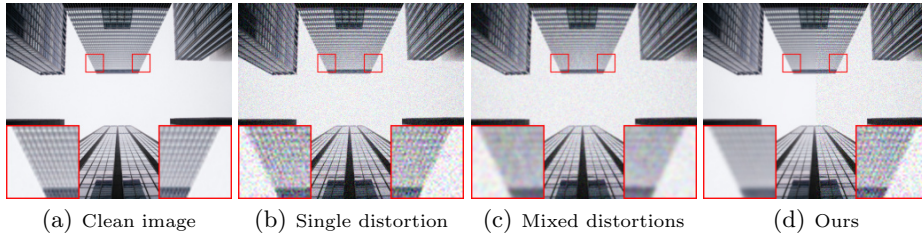


Fig. 1. Comparison of three different image distortion assumptions. **(a)** Clean image. **(b)** Single distortion (Gaussian noise). Only one corruption is applied to the image. **(c)** Mixed distortions [15]. Multiple distortions corrupt the image in sequentially (Gaussian blur and Gaussian noise), but no variation on the spatial domain. **(d)** Our proposed spatially-variant distortion. Instead of mixing in sequentially, we *spatially* combine heterogeneous distortions (left: Gaussian blur, right: Gaussian noise).

or multiple distortions in one image such as JPEG artifact of the blurry image or a photo taken on a hazy and rainy day. To cope with such a multi-distortions scenario, some studies have proposed new datasets [15, 16] and methods [17, 18] recently. They generated combined distortion datasets by overlapping multiple distortions sequentially, which makes the assumption more realistic than the single distortion datasets (Figure 1c).

Viewing in a different point of the multi-distortions, we introduce a spatially-heterogeneous distortion dataset (SHDD). In our dataset, distortions are applied in different regions of the image (Figure 1d). This concept makes our dataset a proxy environment of the real-world applications such as multi-camera systems. For example, in the case where the images are acquired from various devices or post-processors, stitching these images may produce output that has different quality regions, thus degrading the recognition performance. Because of the nature of the spatial-heterogeneity, it is crucial to catch both *what* and *where* the corruptions are, unlike the existing multi-distortion datasets [15, 16] which spread corruptions to the entire image. Recently, Ahn *et al.* [19] proposed a multi-distortion dataset similar to ours. However, corruptions of their dataset are spatially sparse thus may not be ideal to our potential applications (*i.e.* image stitching). In addition, their work is limited to recognizing the distortions.

To address the above requirements, we propose a mixture of experts with a parameter sharing network (MEPSNet) that effectively restores an image corrupted by the spatially-varying distortions. Motivated by the multi-task learning [21, 22] and the mixture of experts [23], we build our network to have a multi-expert system (Figure 2). With this approach, individual distortion can be treated as a single task, thus the model divides and distributes the task to appropriate experts internally. By doing so, experts are able to concentrate on restoring only the given single distortion. We experimentally observed that each expert learns a particular distortion distribution as well. Note that even though we build every expert to be identical, any type of the network design can be adapted thanks to the flexibility and modularity of our framework.

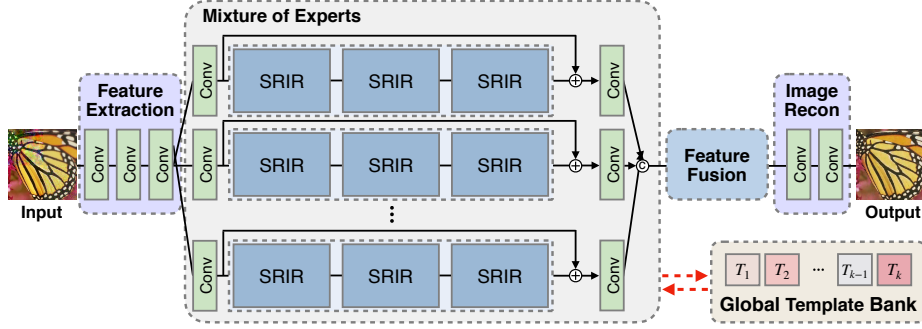


Fig. 2. Overview of the MEPSNet. Our network is composed of 1) feature extraction, 2) mixture of experts, 3) global template bank, 4) feature fusion, and 5) reconstruction modules. Here, SRIR denotes our proposed shared residual-in-residual block. \odot and \oplus symbols are concatenation and element-wise addition operations, respectively. The mixture of experts unit has several pathways, each with multiple SRIR blocks. The parameters of all SRIRs are soft-shared through the global template bank [20].

However, naively constructing the sub-networks (experts) may limit the power of using MTL. Misra *et al.* [24] investigated the trade-offs amongst different combinations of the shared and the task-specific architectures and revealed that the performance mostly depends on the tasks, not on the proportion of the shared units. Based on the above investigation, they proposed a cross-stitch unit so that the network can learn an optimal combination of shared and task-specific representations. Following the analysis of Misra *et al.* [24], we use the soft parameter sharing [20] to guide experts to learn both shared and distortion-specific information effectively. In this approach, convolutional layers of the experts only contain the coefficient vector, not the entire weights and biases. Instead, these are stored in the global template bank thus layers adaptively generate their parameters by a linear combination between the coefficient vector and the templates. It allows each expert to grasp not only the characteristics of the individual distortions but also common representation from the various corruptions automatically. In addition, the number of the parameters is decoupled to the number of the experts and we can increase the experts only using negligible additional parameters. Our experiments show that MEPSNet outperforms other restoration methods including OWAN [17] which is specifically designed network to manage multiple distortions. We summarize our contributions as follows:

- We introduce a new image distortion restoration task and dataset (**SHDD**). It simulates the cases where the various distortions are applied to different locations of an image, a very common scenario in the real world.
- We propose a novel distortion restoration method, **MEPSNet**, which successfully restores the multiple corruptions by jointly adapting two motivations from the multi-task learning and the parameter sharing approach.
- Our proposed model shows significant improvement over other image restoration networks in spatially-heterogeneous distortion environments.

2 Related Work

2.1 Image Distortion Restoration

Image distortion restoration has been a very active topic in the computer vision field for decades [25, 26]. This task aims to reconstruct a distortion-free image from a corrupted input image. Recently, deep learning-based methods show drastic performance improvement in various image restoration tasks such as image super-resolution [1, 3, 27, 28], denoising [6–10], and deblurring [11–14]. All the aforementioned methods follow the supervised approach where the model is trained on an external dataset that has clean and distorted image pairs. Thanks to the high-complexity of the deep networks, such a training scheme is powerful if the volume of the training data is large enough. However, the performance is heavily deteriorated when the dataset size becomes small [29, 30] or the dataset distributions between the training and testing are mismatched [31, 32].

2.2 Multiple Image Distortion Restoration

In real-world applications, multiple distortions can damage entire images, or only the partial regions. Restoring such images using the model trained on a single distortion dataset may produce undesirable artifacts due to the mismatched distribution. To close the gap between the real and simulated data, recent studies have proposed new datasets [15, 16, 19] and methods [17, 18] for multi-distortion restoration task. In their datasets, images are damaged with sequentially applied distortions [15, 16] or only small parts of the image are corrupted [19]. To restore multiple distortions, Yu *et al.* [15] used the toolbox that has several distortion specialized tools. Then, the framework learns to choose the appropriate tool given the current image. Similarly, path-restore [18] and OWAN [17] adopt a multi-path approach so that the models dynamically select an appropriate pathway for each image regions or distortions. Although our method is also motivated by the multi-path scheme, we have two key differences. First, our proposed network is built for restoring spatially-varying distortions. Second, by cooperatively using a mixture of experts and parameter sharing strategies, we can achieve more advanced performance than the other competitors.

2.3 Multi-task Learning

Multi-task learning (MTL) guides a model to learn both common and distinct representations across different tasks [21, 22]. One of the widely used approaches for MTL is combining a shared and task-specific modules [21]. Based on this work, numerous studies have been investigated the power of MTL in various tasks [24, 33, 34]. Among them Misra *et al.* [24] is one notable work; they proposed a cross-stitch unit to optimize the best combination settings for given tasks with a end-to-end training. Without this module, the optimal point depends on the tasks, and the searching process may become cumbersome. Hinted by this work, our network also learns to find the balance between the shared and the distortion-specific representations using the soft parameter sharing approach.

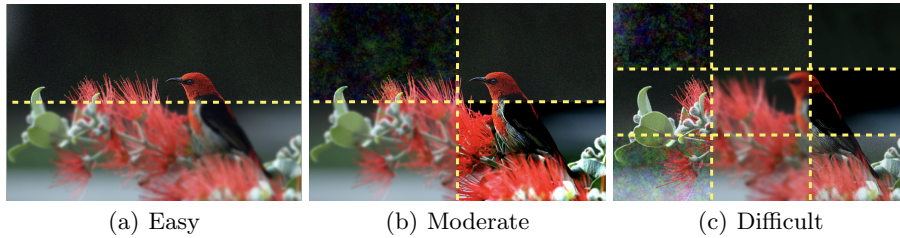


Fig. 3. Examples of our proposed spatially-heterogeneous distortion dataset (SHDD). We separate our dataset into three levels (*easy*, *moderate*, and *difficult*) according to the number of the blocks in a single image. To generate a dataset, we first split image to sub-images using the virtual perforated line (divide phase) and corrupt each region with different distortions (distort phase). Best viewed on display.

3 Spatially-Heterogeneous Distortion Dataset

In this section, we introduce a novel spatially-heterogeneous distortion dataset (SHDD). Our proposed dataset is designed to simulate the scenario where the image is corrupted by spatially varying distortions. To implement this idea, we synthetically generate corrupted images using *divide-and-distort* procedure. That is, we divide clean images into the multiple blocks (divide), and corrupt each block with selected distortions (distort).

In *divide* phase, we split images according to the virtual horizontal or vertical lines (Figure 3). These lines are randomly arranged so as to prevent the model from memorizing the position of the resulting regions. We create three levels of difficulties (easy, moderate, and difficult), by varying the number of split regions. The reason for creating a multi-level dataset is two-fold. First, we consider the relationship between the restoration hardness and the number of regions presented in a single image. Second, we would like to explore the robustness of the model by training on one level and evaluating it on others. In *distort* stage, we corrupt each block with randomly selected distortion. We use **1**) Gaussian noise, **2**) Gaussian blur, **3**) f-noise, **4**) contrast change, and **5**) identity (no distortion). Note that we include *identity* to the distortion pool. By including it, we can measure the generalizability of the model in depth since deep restoration methods tend to over-sharpen or over-smooth when the input is already of high-quality [30]. In addition, it simulates more realistic cases where the real-world scenarios suffer very often (*i.e.* stitching clean image to the corrupted ones).

We build our SHDD based on the DIV2K dataset [35]. It has 800 and 100 images for training and validation respectively. We use half of the DIV2K valid set as validation of the SHDD and the rest of half for testing. For each of the high-quality images, we generate 12 distorted images (training dataset: $9,600 = 800 \times 12$ images) to cover data samples as densely as possible since SHDD is inherently sparse due to the spatially-varying distortions. We set {easy, moderate, difficult}-levels by chopping each image {2, 3, 4}-times (Figure 3).

The distortions used in our SHDD are carefully selected following the recent image distortion datasets [36, 37]. These reflect the real-world scenario, especially for image acquisition and registration stage. When applying the distortions, we randomly sample its strength from following ranges: **1)** [0.005, 0.02]-variances for Gaussian white noise, **2)** [1.0, 2.5]-variances for Gaussian blur, **3)** [6.0, 10.0]-scales for f-noise (pink noise), and **4)** [25.0, 40.0]-levels for contrast change. We implement the corruptions using scikit-image library [38]. The detailed generation procedure can be founded in our officially released code (Section 5.1).

4 Our Method

Our proposed *mixture of experts with a parameter sharing network* (MEPSNet) is composed of five parts: feature extraction, mixture of experts, template bank, feature fusion, and reconstruction blocks (Figure 2). In Section 4.1, we show an overview of our proposed method. Then, we describe the multi-expert architecture and the feature fusion module in Section 4.2 and 4.3.

4.1 Model Overview

Let’s denote \mathbf{X} and \mathbf{y} as a distorted and a clean image, respectively. Given the image \mathbf{X} , a feature extractor f_{ext} computes the intermediate feature \mathbf{F}_0 as:

$$\mathbf{F}_0 = f_{ext}(\mathbf{X}). \quad (1)$$

To extract informative features, the extraction module has multiple convolutional layers (we use three layers) and their dimensions are gradually expanded up to 256 unlike the recent image restoration methods [3, 5]. We observed that the capacity of the extraction module makes the impact to the performance. We conjecture that it is due to the usage of multiple distortion-specialized experts (Section 4.2). With this concept, it is crucial to extract informative shared representation to encourage the individual experts concentrate solely on their own goal. Extracted intermediate feature \mathbf{F}_0 is then fed into the mixture of experts module which outputs a concatenated feature \mathbf{F}_D as in below.

$$\mathbf{F}_D = [f_{exp}^k(\mathbf{F}_0)], \text{ for } k = 1 \dots N \quad (2)$$

Here, N is the number of experts, f_{exp} and $[.]$ denote the expert branch and the channel-wise concatenation respectively. With this deep feature \mathbf{F}_D , we finally generate restored image $\hat{\mathbf{y}}$ by Equation 3. To guide the reconstruction module to gather multiple information adequately, we attach the attentive feature fusion module f_{fuse} before the image reconstruction unit (Section 4.3).

$$\hat{\mathbf{y}} = f_{recon}(f_{fuse}(\mathbf{F}_D)). \quad (3)$$

We optimize our MEPSNet using a pixel-wise L_2 loss function. While several criteria for training restoration network have been investigated [39, 40], we observed that there is no performance gain of using other loss functions in our task. More detailed training setups are shown in Section 5.1.

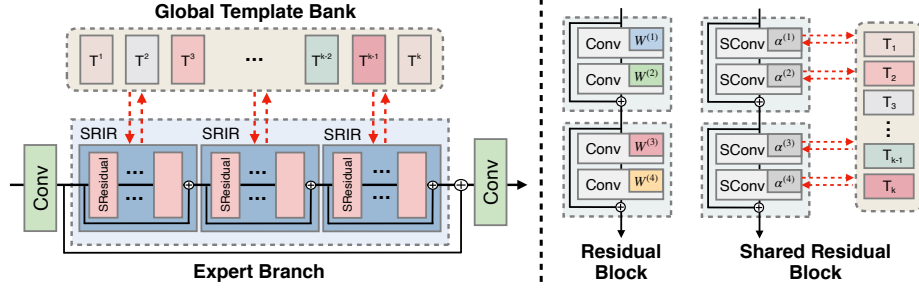


Fig. 4. Our proposed mixture of experts module. **(Left)** Expert branch with parameter sharing. Experts are shared using global template bank (other branches are omitted). In each pathway, there exist three shared residual-in-residual (SRIR) units which have several shared residual blocks. **(Right)** Comparison of the standard and our shared residual blocks. While convolutional layers of the standard block have their own parameters $\{W^{(1)}, \dots, W^{(4)}\}$, ours only have coefficient. Instead, weights are adaptively generated using coefficients $\{\alpha^{(1)}, \dots, \alpha^{(4)}\}$ and templates $\{T^1, \dots, T^k\}$.

4.2 Mixture of Parameter Shared Experts

Mixture of experts module. In our network, this module is the key component to successfully restore heterogeneous distortions. As shown in Figure 2, multiple branches, dubbed as *experts*, are positioned in between the feature extraction and the feature fusion blocks. Each expert has the same structure, which consists of three contiguous shared residual-in-residual (SRIR) units and few convolutional layers (green boxes in Figure 2) that envelope the SRIR blocks. Following the prior works [3–5], we use a long skip connection to bridge across multiple intermediate features for stable training. The SRIR is composed of multiple residual blocks [3, 41] as shown in Figure 4 (left). We also employ additional shortcut connection between the residual blocks to further stabilize the training [5]. Note that the structure of the experts is not restricted to be identical; they could be the networks with different receptive fields [42] or disparate operations [17]. However, we choose to set all the experts to have same structure considering both the simplicity and the performance.

In contrast to the conventional mixture of experts [23], our experts module does not have an external gating network. Instead, distilled information adaptively selects their importance themselves by the self-attention scheme (Section 4.3). Since we do not attach additional gating mechanism, now the formulation of our mixture of experts module is related to the multi-branch networks [43, 44]. However, we observed that the vanilla multi-branch network requires very careful tuning to stabilize the training and even shows degraded performance when we increase the number of the branches or the depths of each branch. We hypothesize that the degradation issue of using multi-branch system arises due to the *isolated branch* structure. That is, no bridge exists between the branches, thus experts (branches) learn all the representations on their own way without referring others. This is inefficient since some information are sufficient to be

extracted once and shared to others. To mitigate such issue, we employ soft parameter sharing scheme to the mixture of experts module.

Soft parameter sharing. We use this scheme [20] to guide the experts in acquiring both shared and distortion-specific information effectively. Contrary to the hard parameter sharing (*i.e.* recursive network), parameters of the layer are generated by a linear combination of the template tensors in the bank. We set the bank as global (Figure 4, left) so that all the SRIRs are shared altogether. The SRIR has shared residual blocks (SResidual) which communicate with a template bank. The SResidual is composed of several shared convolutional layers (SConv), and the parameters of the SConv are adaptively generated through the template bank (Figure 4, right). In detail, a standard convolutional layer stores weights $W \in \mathbb{R}^{C_{in} \times C_{out} \times S \times S}$ (S is kernel size). In contrast, our SConv only contains a coefficient vector $\alpha \in \mathbb{R}^K$, where K is the number of templates in the bank. Instead, the global template bank holds all the weights as $\{\mathbf{T}_1, \dots, \mathbf{T}_K\}$ where \mathbf{T}_k is the $[C_{in} \times C_{out} \times S \times S]$ -dimensional tensor. By referring these templates, each layer generates their adaptive weight \tilde{W} as

$$\tilde{W} = \sum_{j=1}^K \alpha_j \mathbf{T}_j. \quad (4)$$

Jointly using a parameter sharing and the mixture of experts provides two advantages: First, the number of the parameters is determined by the number of templates K , not the experts. Second, it improves the restoration performance compared to the model without a parameter sharing. In detail, we share the parameters not only within the experts but also between the branches. This allows every expert to jointly optimize the common representations from various distortions while each expert produces their specialized features. We can also interpret the benefit of the parameter sharing as in the multi-task learning literature. In a multi-task learning context, finding a good balance between the task-specific and the shared representations is cumbersome job and moreover, the optimal point depends on the tasks themselves [24]. To find the best combination without human-laboring, they share the intermediate representations using a cross-stitch unit [24]. Our approach has an analogous role and motivation to them but we tackle this issue using a parameter sharing scheme.

4.3 Attentive Feature Fusion

As described in Section 4.2, each expert branch generates their specific high-level features. Our attentive feature fusion module takes concatenated features \mathbf{F}_D , which is the output of the mixture of experts module, and fuses this information via channel-wise attention mechanism [5, 45]. With given feature $\mathbf{F}_D \in \mathbb{R}^{C \times H \times W}$, we first apply global average pooling to make C -dimensional channel descriptor $\mathbf{F}_{CD} \in \mathbb{R}^C$ as in below.

$$\mathbf{F}_{CD} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \mathbf{F}_D(i, j), \quad (5)$$

where $\mathbf{F}_D(i, j)$ denotes the (y, x) position of the feature \mathbf{F}_D . With \mathbf{F}_{CD} , we calculate the scaling vector \mathbf{S} using a two-layer network followed by a simple gating scheme. Then, \mathbf{F}_F is produced by multiplying the scaling vector \mathbf{S} and the feature \mathbf{F}_D in a channel-wise manner via Equation 6. Finally, the reconstruction block receives this feature and generates a restored image $\hat{\mathbf{y}}$.

$$\begin{aligned}\mathbf{S} &= \sigma(W_2 \cdot \delta(W_1 \cdot \mathbf{F}_{CD})), \\ \mathbf{F}_F &= \mathbf{S} \cdot \mathbf{F}_D.\end{aligned}\tag{6}$$

Here, $\sigma(\cdot)$ and $\delta(\cdot)$ are sigmoid and ReLU respectively, and $\{W_1, W_2\}$ denotes the weight set of convolutional layers. With this attentive feature fusion module, diverse representations inside of the \mathbf{F}_D are adaptively selected. Unlike ours, previous mixture of experts methods [23] generate attention vector using the external network. However, we observed that such design choice does not work well in our task. Related to the isolation issue of the vanilla multi-branch network, as described in Section 4.2, we suspect that isolated external gating network cannot judge how to select features from the multiple experts adequately. In contrast, our fusion module is based on the self-attention [5, 45, 46]. With this concept, attentive feature fusion unit is now closely linked to the main expert module so that is able to decide which feature to take or not more clearly.

5 Experiments

5.1 Implementation Details

We train all the models on moderate level of SHDD. The reason for using single level only for training is to measure the generalizability of the model by evaluating on unseen (easy and difficult) cases. In each training batch, 16 patches with a size of 80×80 are used as input. We train the model for 1.2M iterations using ADAM optimizer [47] with settings of $(\beta_1, \beta_2, \epsilon) = (0.9, 0.99, 10^{-8})$, and weight decay as 10^{-4} . We initialize the network parameters following He *et al.* [48]. The learning rate is initially set to 10^{-4} and halved at 120K and 300K iterations. Unless mentioned, our network consists of three experts, each of which has three SRIRs. We choose the number of SResidual blocks in SRIR to 12 and the number of the templates K as 16. We release the code and dataset on <https://github.com/SijinKim/mepsnet>.

5.2 Comparison to the Other Methods

Baseline. we use following deep restoration methods: DnCNN [10], VDSR [2], OWAN [17] and RIDNet [6]. OWAN is proposed to restore multiple distortions while others are for a single distortion. We modify VDSR by stacking convolutional layers four times than the original ones to match the number of the parameters to the others. For OWAN and RIDNet, we use author’s official code.

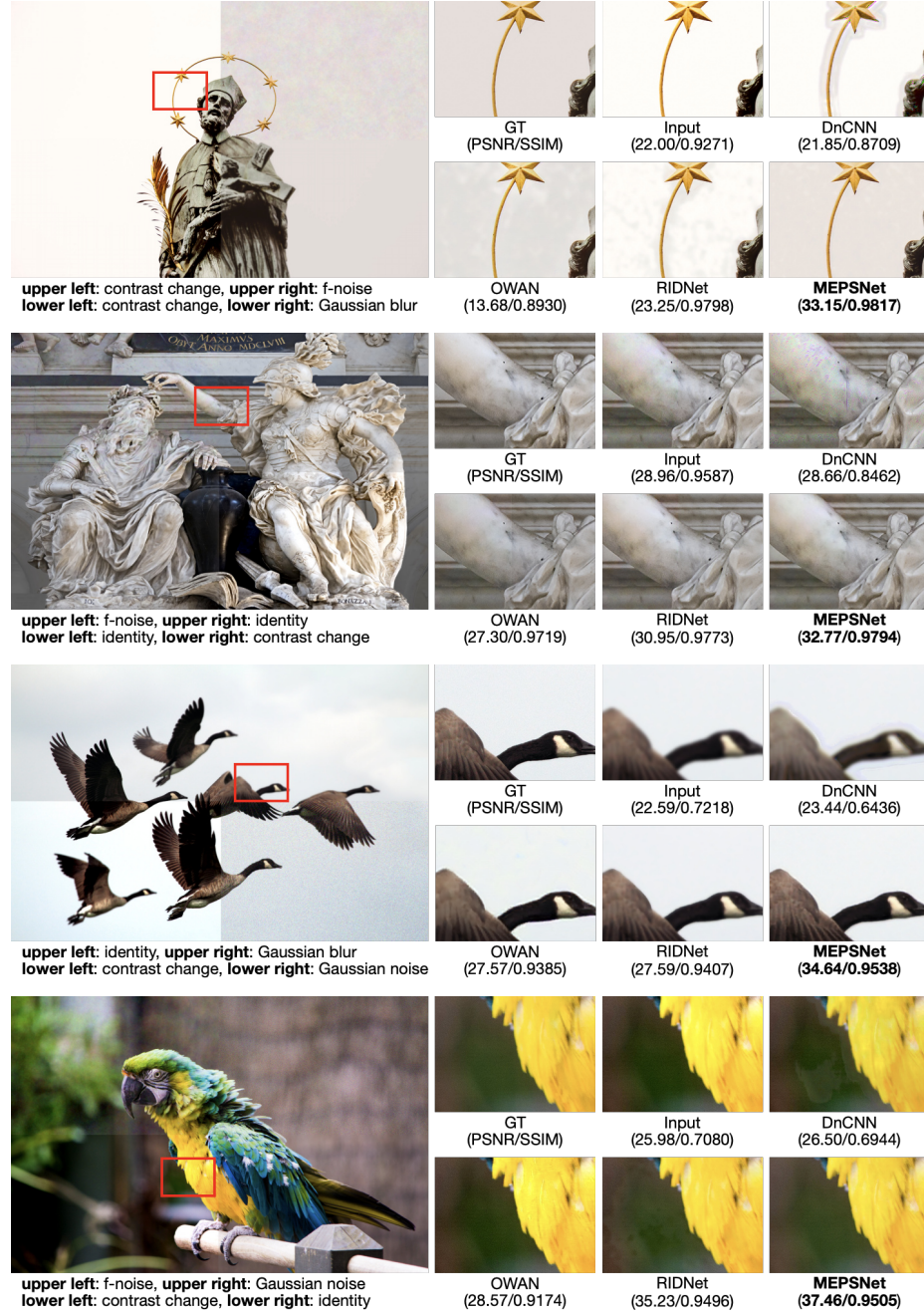


Fig. 5. Qualitative comparison of the MEPSNet and other restoration methods.

Table 1. Quantitative comparison (PSNR / SSIM) on SHDD in three levels to the deep learning-based restoration methods.

Method	Levels of SHDD		
	Easy	Moderate	Difficult
DnCNN [10]	25.29 / 0.7110	25.54 / 0.7354	26.70 / 0.7723
VDSR [2]	27.34 / 0.7709	25.73 / 0.7701	25.95 / 0.7760
OWAN [17]	30.95 / 0.9181	29.77 / 0.9112	29.27 / 0.9098
RIDNet [6]	34.19 / 0.9361	32.94 / 0.9317	32.30 / 0.9282
MEPSNet (ours)	34.23 / 0.9369	33.47 / 0.9331	32.71 / 0.9284

Table 2. Quantitative comparison (mAP) on object detection and semantic segmentation tasks. We use faster R-CNN [49] and mask R-CNN [33] to measure mAP for object detection and instance segmentation, respectively.

Task	Clean	Distorted	DnCNN	VDSR	OWAN	RIDNet	MEPSNet
Detection	40.2	26.4	26.8	25.6	28.6	29.5	29.5
Segmentation	37.2	24.4	24.8	23.6	26.5	27.4	27.5

Evaluation on SHDD. We compare the MEPSNet to the baselines on SHDD using pixel-driven metrics such as PSNR and SSIM. Table 1 shows the quantitative comparison on the different levels of the SHDD test set. In this benchmark, our proposed method consistently outperforms the others. For example, the performance gain of the MEPSNet in moderate level is +0.53 dB PSNR compared to the second best method, RIDNet. In addition, MEPSNet achieves the best performance on the unseen settings as well, and especially shows the superior PSNR on difficult level, +0.41 dB to the second best. It should be noted that OWAN [17] is also devised for the multi-distortion restoration. However, their performance is much lower than both ours and RIDNet. We conjecture that isolating all the operation layers and attention layer results in degraded performance. On the other hand, ours can fully enjoy the effect of using multi-route by sharing the parameters altogether. Figure 5 shows the qualitative results of our model. For the contrast change distortion (1st and 4th rows), the other methods create unpleasant spots (OWAN, RIDNet) or regions (DnCNN) while ours successfully reconstructs the original color. Similarly, MEPSNet effectively restores the other corruptions, such as f-noise (2nd row) or Gaussian blur (3rd row).

Evaluation on image recognition tasks. To further compare the performance of our method, we use image recognition tasks: object detection and semantic segmentation. To be specific, we distort images of COCO dataset [50] with same protocols of SHDD. Then, we restore distorted images using the trained models on SHDD. We evaluate mean average precision (mAP) score using faster R-CNN [49] and mask R-CNN [33] for detection and segmentation respectively. As in Table 2, mAPs of the distorted images are significantly lower than the clean

Table 3. Ablation study. ME and PS denote the mixture of experts and parameter sharing, respectively. Using both modules dramatically improves the performance of the baseline and successfully suppressing the number of the parameters.

ME	PS	# Experts	# Params.	PSNR / SSIM
		1	3.9M	29.36 / 0.8835
	✓	1	1.9M	33.55 / 0.9322
✓	✓	3	2.2M	34.29 / 0.9353
✓	✓	5	2.6M	34.39/0.9362

Table 4. Effect of the multi experts and parameter sharing under the layer constraint scenario. We force the number of the residual (or SResidual) blocks in entire mixture of experts module as 36. That is, expert has 36 blocks for single expert case (1st, 3rd rows), whereas 12 blocks for each when using three experts (2nd, 4th rows).

# Blocks	# Experts	PS	PSNR / SSIM
36	1		29.36 / 0.8835
	3		32.21/0.9226
	1	✓	33.55 / 0.9322
	3	✓	33.78/0.9334

cases. Restored results with our proposed MEPSNet show the best mAP than the other methods and RIDNet [6] is the only method comparable to ours.

5.3 Model Analysis

In this section, we dissect our proposed MEPSNet through the internal analysis. Unless mentioned, we set the MEPSNet to have three SRIRs each of which includes twelve SResidual blocks. We trained our model using 48×48 input patches.

Ablation study. In Table 3, we analyze how the mixture of experts (ME) and the parameter sharing (PS) affect the restoration performance. First, using PS (2nd row) outperforms the baseline (1st row) by a huge margin only using half of the parameters. We hypothesize that the PS through the global template bank successfully guides the model to combine low- and high-level features internally. The advantages of combining the multiple features are also verified in recent restoration methods [4, 51], and network with PS (via template bank) enjoy the fruitful results by an alternative implementation of the feature aggregating.

Simultaneously applying PS and ME additionally gives dramatic improvements (2nd vs. 3rd rows). Even though we triple the number of experts, the total number of parameters is marginally increased by only 15%, thanks to the parameter sharing scheme. Increasing the number of the experts to five (4th row) further boosts the performance as well. However, unless we share the parameters, using five experts increases about 40% of the parameters compared to the single expert network due to the additional coefficients and extra burden to the

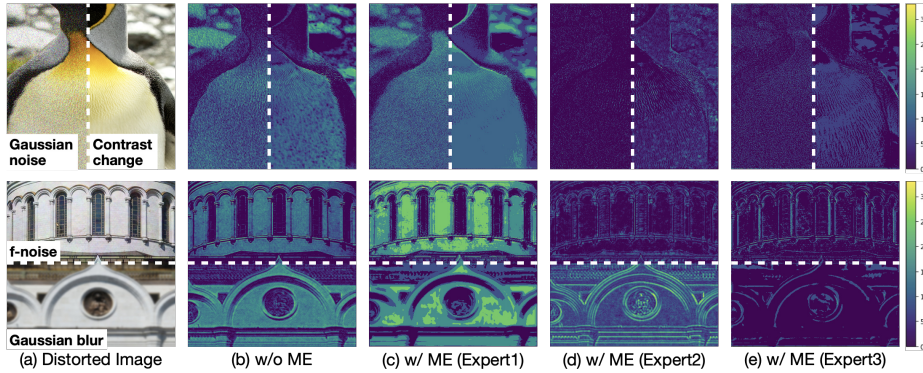


Fig. 6. Visualization of the extracted feature maps from the mixture of experts module. **(a)-upper** Distorted image by Gaussian noise (left) and contrast change (right). **(a)-lower** Distorted image by f-noise (top) and Gaussian blur (bottom). **(b)** Generated feature map of the single expert module, without mixture of experts. **(c-e)** Feature maps produces by three different experts when using multi-expert system.

fusion module. Considering the trade-off between the number of the parameters and the performance, we choose to use three experts for the final model.

To analyze the impact of the mixture of experts and parameter sharing more clearly, we conduct an experiment based on the layer constraint setting as in Table 4. In this scenario, the number of the residual (or SResidual) blocks in the entire mixture of experts module is fixed to 36. Without a multi-expert (1st, 3rd rows), models are three times deeper than the others (2nd, 4th rows). However, single expert models result in degraded performance than the multi-expert. It may contradict the recent trends of single distortion restoration task [3, 5]: deeper network is better than the shallow one. Such a result may indicate that it is necessary to view multi-distortion restoration task on a different angle to the single distortion restoration literature.

Feature visualization. Figure 6 shows the output feature map of the mixture of experts module. The model without a mixture of experts (Figure 6b) struggles to recover all the distortions simultaneously while ours separates the role to each other (Figure 6c-e). For example, expert 1 (c) produces coarse and large activations, implying that it mainly deals with contrast change and partially reduces the color tone of the f-noise and Gaussian noise. On the other hand, expert 2 (d) concentrates on recovering edges for the Gaussian blur. The expert 3 (e) also focuses on the primitives but finer elements than the expert 2.

Effect of the number of layers and templates. In Figure 7a, we fix the number of the experts to three and vary the number of the SResidual blocks for each of the expert. Not surprisingly, we can stack more layers without a sudden increase in the number of the parameters. The performances are consistently improved except the 45 blocks case may due to the unstable training of the extremely deep network. Increasing the number of the templates also gives

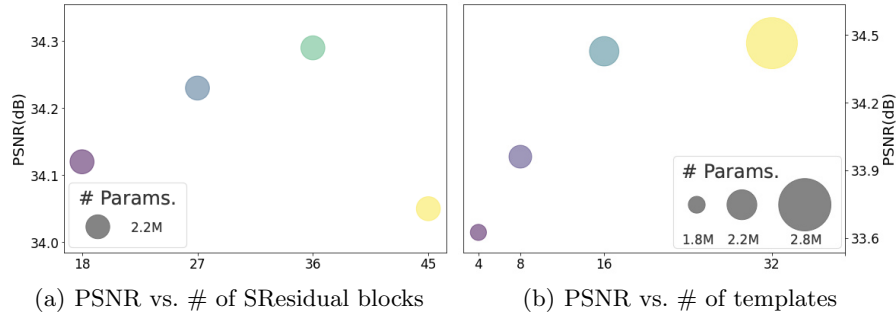


Fig. 7. Effect of the number of the SResidual blocks and the templates. **(a)** Varying the number of the blocks of each expert. The number of the experts are fixed as three for all the cases. With parameter sharing, they all have similar parameters. **(b)** Increasing the number of the templates in the global bank from 4 to 32.

the progressive gains as show in Figure 7b. With diverse templates, layers can generate more complex and advanced weight combinations so that it is possible to restore complicated distortion patterns.

6 Conclusion

In this paper, we have presented the spatially-heterogeneous distortion dataset (SHDD) and the mixture of experts with a parameter sharing network (MEPSNet) for effective distortion restoration. The proposed SHDD assumes the cases where the multiple corruptions are applied to the different locations. To appropriately handle the above scenario, our method is motivated by the analysis from the multi-task learning contexts [24]. By jointly utilizing the mixture of experts scheme [23] and the parameter sharing technique [20], MEPSNet outperforms the other image restoration methods on both the pixel-based metrics (PSNR and SSIM) and the indirect measures (image detection and segmentation). As future work, we plan to integrate the spatially-heterogeneous and the sequentially-combined distortions [15] concepts to further reduce the disparity between the simulated and the real-world environments.

Acknowledgement. This research was supported by the National Research Foundation of Korea grant funded by the Korea government (MSIT) (No. NRF-2019R1A2C1006608), and also under the ITRC (Information Technology Research Center) support program (IITP-2020-2018-0-01431) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

1. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: European conference on computer vision, Springer

- (2014) 184–199
2. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 1646–1654
 3. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. (2017) 136–144
 4. Ahn, N., Kang, B., Sohn, K.A.: Fast, accurate, and lightweight super-resolution with cascading residual network. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 252–268
 5. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 286–301
 6. Anwar, S., Barnes, N.: Real image denoising with feature attention. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 3155–3164
 7. Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., Barron, J.T.: Unprocessing images for learned raw denoising. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 11036–11045
 8. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: 2012 IEEE conference on computer vision and pattern recognition, IEEE (2012) 2392–2399
 9. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing* **27** (2018) 4608–4622
 10. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing* **26** (2017) 3142–3155
 11. Kumar, N., Nallamothu, R., Sethi, A.: Neural network based image deblurring. In: 11th Symposium on Neural Network Applications in Electrical Engineering, IEEE (2012) 219–222
 12. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 3883–3891
 13. Zhang, J., Pan, J., Ren, J., Song, Y., Bao, L., Lau, R.W., Yang, M.H.: Dynamic scene deblurring using spatially variant recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 2521–2529
 14. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
 15. Yu, K., Dong, C., Lin, L., Change Loy, C.: Crafting a toolchain for image restoration by deep reinforcement learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 2443–2452
 16. Liu, X., Suganuma, M., Luo, X., Okatani, T.: Restoring images with unknown degradation factors by recurrent use of a multi-branch network. *arXiv preprint arXiv:1907.04508* (2019)
 17. Suganuma, M., Liu, X., Okatani, T.: Attention-based adaptive selection of operations for image restoration in the presence of unknown combined distortions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 9039–9048

18. Yu, K., Wang, X., Dong, C., Tang, X., Loy, C.C.: Path-restore: Learning network path selection for image restoration. *arXiv preprint arXiv:1904.10343* (2019)
19. Ahn, N., Kang, B., Sohn, K.A.: Image distortion detection using convolutional neural network. In: 2017 4th IAPR Asian Conference on Pattern Recognition (ACPR), IEEE (2017) 220–225
20. Savarese, P., Maire, M.: Learning implicitly recurrent cnns through parameter sharing. *arXiv preprint arXiv:1902.09701* (2019)
21. Caruana, R.: Multitask learning. *Machine learning* **28** (1997) 41–75
22. Ruder, S.: An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017)
23. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. *Neural computation* **3** (1991) 79–87
24. Misra, I., Shrivastava, A., Gupta, A., Hebert, M.: Cross-stitch networks for multi-task learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 3994–4003
25. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing* **16** (2007) 2080–2095
26. Roth, S., Black, M.J.: Fields of experts. *International Journal of Computer Vision* **82** (2009) 205
27. Dong, C., Loy, C.C., Tang, X.: Accelerating the super-resolution convolutional neural network. In: European conference on computer vision, Springer (2016) 391–407
28. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 4681–4690
29. Feng, R., Gu, J., Qiao, Y., Dong, C.: Suppressing model overfitting for image super-resolution networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2019) 0–0
30. Yoo, J., Ahn, N., Sohn, K.A.: Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. *arXiv preprint arXiv:2004.00448* (2020)
31. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 1712–1722
32. Cai, J., Zeng, H., Yong, H., Cao, Z., Zhang, L.: Toward real-world single image super-resolution: A new benchmark and a new model. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 3086–3095
33. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. (2017) 2961–2969
34. Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. (2018) 1930–1939
35. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2017) 126–135
36. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 586–595

37. Ponomarenko, N., Jin, L., Ieremeiev, O., Lukin, V., Egiazarian, K., Astola, J., Vozel, B., Chehdi, K., Carli, M., Battisti, F., et al.: Image database tid2013: Peculiarities, results and perspectives. *Signal processing: Image communication* **30** (2015) 57–77
38. Van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T.: scikit-image: image processing in python. *PeerJ* **2** (2014) e453
39. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2017) 624–632
40. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence* **38** (2015) 295–307
41. 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*. (2016) 770–778
42. Wang, Y., Tao, X., Qi, X., Shen, X., Jia, J.: Image inpainting via generative multi-column convolutional neural networks. In: *Advances in neural information processing systems*. (2018) 331–340
43. Ren, H., El-Khamy, M., Lee, J.: Image super resolution based on fusing multiple convolution neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. (2017) 54–61
44. Du, X., El-Khamy, M., Lee, J., Davis, L.: Fused dnn: A deep neural network fusion approach to fast and robust pedestrian detection. In: *2017 IEEE winter conference on applications of computer vision (WACV)*, IEEE (2017) 953–961
45. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2018) 7132–7141
46. Woo, S., Park, J., Lee, J.Y., So Kweon, I.: Cbam: Convolutional block attention module. In: *Proceedings of the European conference on computer vision (ECCV)*. (2018) 3–19
47. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
48. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision*. (2015) 1026–1034
49. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems*. (2015) 91–99
50. 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*, Springer (2014) 740–755
51. Tong, T., Li, G., Liu, X., Gao, Q.: Image super-resolution using dense skip connections. In: *Proceedings of the IEEE International Conference on Computer Vision*. (2017) 4799–4807