MoESR: Blind Super-Resolution using Kernel-Aware Mixture of Experts

Mohammad Emad[1] Maurice Peemen[2] Henk Corporaal[1]
[1]Eindhoven University of Technology, Netherlands [2]Thermo Fisher Scientific, Netherlands
{m.emad, h.corporaal}@tue.nl maurice.peemen@thermofisher.com

Abstract

Modern deep learning super-resolution approaches have achieved remarkable performance where the low-resolution (LR) input is a degraded high-resolution (HR) image by a fixed known kernel i.e. kernel-specific super-resolution (SR). However, real images often vary in their degradation kernels, thus a single kernel-specific SR approach does not often produce accurate HR results. Recently, degradation-aware networks are introduced to generate blind SR results for unknown kernel conditions. They can restore images for multiple blur kernels. However, they have to compromise in quality compared to their kernel-specific counterparts. To address this issue, we propose a novel blind SR method called Mixture of Experts Super-Resolution (MoESR), which uses different experts for different degradation kernels. A broad space of degradation kernels is covered by kernel-specific SR networks (experts). We present an accurate kernel prediction method (gating mechanism) by evaluating the sharpness of images generated by experts. Based on the predicted kernel, our most suited expert network is selected for the input image. Finally, we fine-tune the selected network on the test image itself to leverage the advantage of internal learning. Our experimental results on standard synthetic datasets and real images demonstrate that MoESR outperforms state-of-the-art methods both quantitatively and qualitatively. Especially for the challenging $\times 4$ SR task, our PSNR improvement of 0.93 dB on the DIV2KRK dataset is substantial [1].

1. Introduction

Single image super-resolution (SISR) techniques reconstruct a high-resolution (HR) image from their degraded low-resolution (LR) counterpart. This degradation occurs due to camera sensors imperfections, sub-optimal acquisition (*e.g.* unpleasant light or camera shake) and image processing routines. Due to the large space of possible degradations, SISR is an ill-posed problem. It has a broad variety of applications, *e.g.* surveillance [23], medical imaging [32], microscopy [10] and so on.

The use of Convolutional Neural Networks (CNNs) in SISR has improved the state-of-the-art considerably. Many previous works [8, 17, 19, 30, 37, 7, 21] assume a fixed image degradation kernel (usually bicubic). These methods train a CNN on a large dataset of HR and LR images generated with this predefined kernel. However, these earlier methods perform poorly on realistic images since the real degradation process is complicated and varies from image to image. Recently, a few methods have been proposed to solve blind SR i.e. the degradation kernel is unknown. They usually estimate the degradation kernel and recover the HR image from only a single LR input.

Some blind SR methods train a single degradation-aware network (for multiple kernels) on external datasets [12, 14, 6]. However, their performance is inferior to a specialized network trained with a single kernel. In addition, these methods do not benefit from internal statistics of the test image. On the other hand, there are self-supervised methods [2, 9, 16], which train a small network at test time on the input image. There are two main drawbacks for these methods: they have longer run-time and there is limited information to learn from a single image.

To use the best aspects of specialized solutions and self-supervised methods, we propose Mixture of Experts Super-Resolution (MoESR), which uses different experts for different degradation kernels. For every input image, we predict the degradation kernel and super-resolve the LR image using the best suited kernel-specific expert. To predict the degradation kernel, we introduce an Image Sharpness Evaluator (ISE) and a Kernel Estimation Network (KEN). ISE evaluates the sharpness of the images generated by the experts. These evaluations are used by KEN to estimate the kernel and select the best pretrained expert network. Finally, the selected network is fine-tuned on the test image to leverage the advantage of internal learning. Our experimental results demonstrate that MoESR outperforms state-of-the-art blind SR methods quantitatively and qualitatively. Our contributions can be summarized as follows:

- A novel kernel-aware mixture of experts approach for

[1]Codes and datasets are available at https://github.com/memad73/MoESR

blind SR based on external and internal learning of kernel-specific expert networks (MoESR).

- A new and accurate kernel estimation method by evaluating the sharpness of images generated by the kernel-specific experts.
- We evaluate our method both on synthetic and real images and show that MoESR outperforms state-of-the-art blind SR methods in terms of quantitative metrics (PSNR/SSIM) and visual quality.

2. Related work

2.1. Degradation-aware SR networks

These methods develop a single SR network for multiple degradations. In addition to the LR image, they take an estimated blur kernel as auxiliary information. A representative degradation-aware SR method is SRMD [36]. By using dimensionality stretching it concatenates degradation information (blur kernel and noise) with the LR image which is used by a non-blind SR network. The work of Gu *et al*. [12] use an Iterative Kernel Correction (IKC) method for blur kernel estimation. The estimated kernel is used in their Spatial Feature Transform (SFT) layers that combines degradation information and the LR input. A similar approach is presented by Cornillère *et al*. [6]. They train a kernel discriminator to detect errors in the generated SR image. Their approach searches a blur kernel that minimizes these errors. Huang *et al*. [14] propose a Deep Alternating Network (DAN) that can predict the blur kernel and SR image using an unfolded end-to-end trainable model. In this model, a chain of Restorer and Estimator modules are stacked alternately, which each module restores the SR image or estimates the blur kernel based on the output of previous module. The main drawback of degradation-aware networks is that combining kernel information and LR image is not efficient since they are naturally in different domains, thus their performance is inferior to a network trained for a specific kernel. In addition, they only rely on external learning while the internal statistics of the input image can be beneficial.

2.2. Self-supervised methods

Natural image priors such as the recurrence of small patches in a single image can be beneficial for both kernel estimation [22] and image reconstruction [11]. Self-supervised methods exploit this recurrence by training a network on patches of the LR input. ZSSR [25] trains from scratch on example patches from the input image and a downscaled version. KernelGAN [2] proposes kernel estimation based on internal learning by a Generative Adversarial Network (GAN). To jointly learn blur kernel and LR-to-HR mapping, DualSR [9] and DBPI [16] develop a dual-path architecture consisting of a downsampler, which learns the blur kernel; and an SR network. DualSR introduces a masked interpolation loss to effectively remove artifacts in the output image. Self-supervised methods suffer from insufficient training data and slow inference speed. To overcome these issues, MZSR [27] leverages meta-transfer learning to find an initial point that quickly adapts to the test image conditions. Although MZSR is fast, the performance drops significantly when the estimated blur kernel is slightly different from the ground-truth kernel.

2.3. Domain translation-based methods

These methods employ adversarial learning for the translation to HR domain by using only unpaired datasets of images. CinCGAN [33] and the improved version MCinCGAN [38] employ a two-step approach for real-world SR. They map a realistic LR input into a noise-free LR space, next the intermediate image is upsampled with a pretrained SR network. Umer *et al*. [29] propose SRResCGAN which uses adversarial training to learn the degradation settings of real-world images. They use pixel-wise supervision to train a network in the HR domain. Unlike degradation-aware and self-supervised methods, this category trains on a collection of real-world images with a similar degradation kernel. In practice, this real-world dataset is not always available. In addition, the performance of these methods usually is inferior to supervised methods. Chen *et al*. [5] reviews the existing real-world SR methods in more detail.

2.4. Mixture of Experts

Mixture of Experts (MoE) divides the problem space over multiple expert networks, where each expert handles a subset of the whole space. MoE is firstly introduced by [15]. Non-blind SR methods like [35, 20] use MoE to improve the overall quality of their model. Wang *et al*. [31] propose a mixture model of networks for blind SR, which is capable of clustering SR tasks of different degradation kernels into a set of groups. They model the degradation kernel with a latent variable, which is inferred from the input image by an encoder network. In contrast, we predict the kernel by evaluating the sharpness of upsampled images generated by the experts. In addition, our method benefits from internal learning to improve the final result.

3. Proposed method

3.1. Degradation model

The basic model that is used in the literature for blind SR is as follows:

$$I_{LR} = (I_{HR} * k) \downarrow_s + n \tag{1}$$

where $*$ denotes the 2D convolution operation, k is the blur kernel, $\downarrow_s$ is downsampling with scale s and n represents

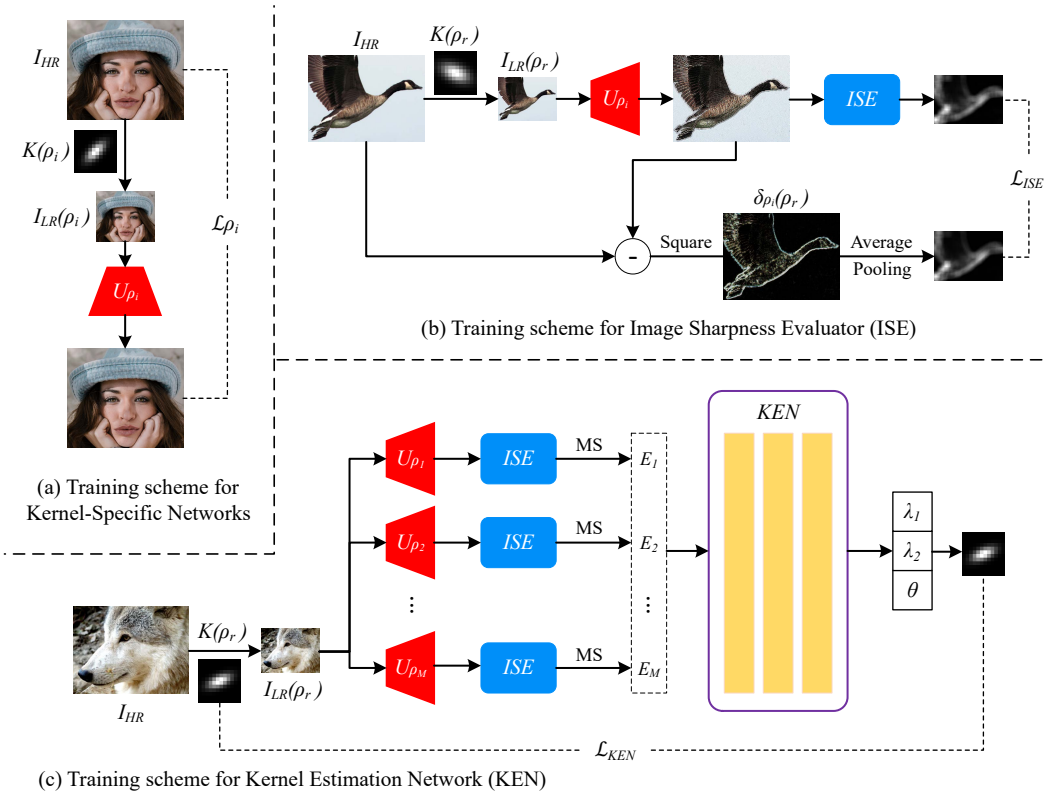


Figure 1. Block diagram of training different networks in MoESR.

additive noise. We aim to recover a high-resolution image I_{HR} from a low-resolution one I_{LR}. The noise n can be reduced by a denoiser before performing super-resolution. Consequently, blind SR mainly focuses on predicting kernel k and reconstructing image I_{HR}.

Literature such as [24, 25, 12] suggests that real-world image degradation can be approximated with isotropic or anisotropic Gaussian kernels. We focus on the broader applicable set of anisotropic kernels that can be considered as the combination of isotropic and motion blur kernels. An anisotropic Gaussian kernel is defined by three parameters: λ_1 and λ_2 which are the eigenvalues of the kernel and θ which is the rotation angle. The covariance matrix of the kernel is calculated as:

$$\Sigma = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \quad (2)$$

Finally, the blur kernel and the degradation model can be expressed as:

$$k = \mathrm{K}(\rho) = \mathrm{K}(\lambda_1, \lambda_2, \theta) \quad (3)$$

$$I_{LR}(\rho) = (I_{HR} * \mathrm{K}(\rho)) \downarrow_s \quad (4)$$

where ρ represents the kernel parameters and K is a function that generates 2D kernel pixels from kernel parameters.

3.2. Optimizing kernel-aware mixture of experts

Instead of training a single network for different blur kernels, a mixture of kernel-specific SR networks (experts) is used. For every image, the group of experts generates a batch of different kernel-specific SR images. Our pretrained Image Sharpness Evaluator (ISE) selects the best SR result. To interpolate the best kernel configuration between the discrete kernels, a Kernel Estimation Network (KEN) is used on the ISE outputs.

Kernel-specific experts Each expert network is trained with different blur kernel parameters. Figure 1 (a) shows the training procedure for each expert. The expert network U_{ρ_i} is trained to upsample images with kernel parameters ρ_i. The used loss function is:

$$\mathcal{L}_{\rho_i} = \|U_{\rho_i}(I_{LR}(\rho_i)) - I_{HR}\|_1 \quad , \quad i \in [1, M] \quad (5)$$

where M is the number of experts and $I_{LR}(\rho_i)$ is calculated with Equation 4. To keep the number of experts limited, we exploit the symmetries in anisotropic kernels. For example, by rotating the LR image with 90 degrees, we simulate a 90 degree rotated kernel. Due to rotation and mirror symmetry, we only require experts for kernels within a rotation angel $\theta \sim (0, \pi/4)$. More details are in supplementary material.

Image Sharpness Evaluator (ISE) In case a wrong blur kernel is used, the SR result will be either blurry or oversharpened. Our ISE module is trained to detect such blurry or oversharpened regions and predicts errors from the ground-truth image. Natural image features result in a low ISE error. The training procedure for ISE is illustrated in Figure 1 (b). We build a large set of kernels ρ_r, which is uniformly sampled from the supported parameter space. The kernels are used to generate LR images which are upsampled by the experts. $\delta_{\rho_i}(\rho_r)$ gives the pixel-wise squared error of the super-resolved images and the ground-truth defined as:

$$\delta_{\rho_i}(\rho_r) = (U_{\rho_i}(I_{LR}(\rho_r)) - I_{HR})^2 \tag{6}$$

In the end, we only need a total error estimate for the whole image. For this task, region-based error prediction is more robust than pixel-based prediction, therefore ISE should predict the average error in a region. To generate the training labels we apply average pooling with a large window size. The loss function can be mathematically represented as:

$$\mathcal{L}_{ISE} = \sum_{i=1}^{M} \|ISE(U_{\rho_i}(I_{LR}(\rho_r))) - \text{AP}(\delta_{\rho_i}(\rho_r))\|_1 \tag{7}$$

where AP is average pooling operation, e.g. 8x in our example.

Kernel Estimation Network (KEN) We apply ISE for all kernel configurations that can be simulated by our experts. Naively selecting the kernel with the smallest ISE error would be sub-optimal since we support a limited number of configurations. To interpolate the degradation kernel more accurately, we introduce a Kernel Estimation Network (KEN). Figure 1 (c) shows the structure used for training KEN. The input data preparation for the training process is similar to the training of ISE. However, now the trained ISE is used to generate mean squared errors. This can be expressed as:

$$E_i = \text{MS}(ISE(U_{\rho_i}(I_{LR}(\rho_r)))) \quad , \quad i \in [1, M] \tag{8}$$

where MS denotes the mean square operation on ISE output pixels. The error vector is input to KEN which is trained with the following loss function:

$$\mathcal{L}_{KEN} = \|\text{K}(KEN(E_1, E_2, ..., E_M)) - \text{K}(\rho_r)\|_1 \tag{9}$$

$\mathcal{L}_{KEN}$ is a pixel based loss on the kernel structure, which gives better accuracy compared to training on kernel parameters ρ. Note that, the weights of expert networks and the ISE module are fixed during training. At test time an input image I_{in} is upsampled for all kernel combinations by applying the kernel-specific experts. For each upsampled image an ISE error is computed as:

$$E_i = \text{MS}(ISE(U_{\rho_i}(I_{in}))) \quad , \quad i \in [1, M] \tag{10}$$

Then we apply KEN to obtain the kernel parameters:

$$\rho^* = KEN(E_1, E_2, ..., E_M) \tag{11}$$

3.3. Fine-tuning and Image reconstruction

After kernel parameter estimation, we can use the best kernel-specific expert U^* to reconstruct the final image. Expert selection is done by:

$$U^* = U_{\rho'} \quad \text{where} \quad \rho' = \underset{\rho \in \{\rho_1 ... \rho_M\}}{\arg\min} \|\rho - \rho^*\|_1 \tag{12}$$

The result of selected expert can be further improved by fine-tuning on the test image. This has two advantages. Firstly, internal learning can exploit feature recurrence in the image as suggested by self-supervised methods [25, 9, 16]. Secondly, the estimated parameters ρ^* will slightly differ from the parameters ρ' of the pretrained expert U^*. By fine-tuning we adjust the selected expert to the parameters ρ^*.

We use the DualSR [9] pipeline for fine-tuning. DualSR is a dual-path architecture that jointly trains an image-specific downsampler (degradation kernel) and corresponding upsampler (SR network). We replace the downsampler with our fixed estimated kernel $K(\rho^*)$ and initialize the upsampler with the pretrained expert U^*. The expert network is fine-tuned using cycle-consistency losses and masked interpolation loss as explained in [9]. We extend the architecture of DualSR to enable $\times 4$ SR. The detailed $\times 4$ architecture is available in the supplementary material section C. Finally we use the fine-tuned network to upsample the test image. The effectiveness of fine-tuning is evaluated in Section 5.2.

4. Experiments

4.1. Implementation details

Network architecture We use 25 specialized expert networks with kernel parameters in the range $\lambda_1, \lambda_2 \sim (0.6, 5)$ and $\theta \sim (0, \pi/4)$. However, by rotating and flipping the LR input, the experts can be used for 85 different kernels. For each expert, we employ a simple 12-layer convolutional network with 64 feature maps for each layer. The network takes the bicubically upsampled image as the input and predicts the residual of input and ground-truth images. For $\times 4$ SR, we use two similar networks in sequence, each upsamples the image by a scale of 2. We found that this is faster and gives better accuracy compared to direct $\times 4$ SR. Our ISE module contains a 6-layer convolutional network with 64 feature maps per layer. The stride of some intermediate

Method	DIV2KRK		Flickr2KRK		Urban100RK	
	×2	×4	×2	×4	×2	×4
Bicubic	28.73 / 0.8040	25.33 / 0.6795	28.62 / 0.7997	25.27 / 0.6758	23.76 / 0.7017	20.94 / 0.5400
SAN [7]	29.21 / 0.8232	25.67 / 0.6947	29.09 / 0.8167	25.52 / 0.6868	24.32 / 0.7298	21.25 / 0.5611
KernelGAN [2]	30.36 / 0.8669	26.81 / 0.7316	30.62 / 0.8612	26.58 / 0.7227	25.35 / 0.7870	21.91 / 0.5939
DBPI [16]	30.77 / 0.8684	26.86 / 0.7368	31.34 / 0.8866	26.27 / 0.7315	26.23 / 0.8165	22.21 / 0.6234
DualSR [9]	30.92 / 0.8728	-	31.14 / 0.8701	-	25.38 / 0.7865	-
[2]+MZSR [27]	30.61 / 0.8615	26.92 / 0.7366	30.50 / 0.8553	26.69 / 0.7266	25.82 / 0.7995	22.33 / 0.6223
[2]+USRNet [34]	27.94 / 0.8084	23.59 / 0.6736	28.43 / 0.8281	23.77 / 0.6796	22.70 / 0.7016	18.54 / 0.5042
BlindSR [6]	31.36 / 0.8720	-	31.64 / 0.8747	-	26.50 / 0.8110	-
IKC [12]	31.15 / 0.8717	27.38 / 0.7640	31.21 / 0.8688	27.33 / 0.7610	25.67 / 0.7835	23.12 / 0.6612
DAN [14]	32.56 / 0.8997	27.55 / 0.7582	32.73 / 0.8917	27.67 / 0.7575	27.04 / 0.8246	23.23 / 0.6621
MoESR w/o FT	32.01 / 0.9002	28.14 / 0.7808	31.67 / 0.8958	28.24 / 0.7800	26.18 / 0.8332	23.60 / 0.6849
MoESR	32.69 / 0.9054	28.48 / 0.7805	32.95 / 0.9056	28.57 / 0.7795	27.29 / 0.8448	23.62 / 0.6766

Table 1. Quantitative results (PSNR / SSIM) of state-of-the-art SR methods on different datasets. The best results are highlighted in red and the second best results are highlighted in blue.

layers is set to 2 and each pixel of the output has a receptive field of 49. For consistency, the average pooling (AP) unit in Figure 1 (b) has windows size of 49 and stride of 8. KEN is a simple 3-layer fully-connected network with 50 features per layer. the input of KEN is normalized to zero mean and unit variance. For all networks, convolution layers are followed by ReLU activations. In secion D of the supplementary material additional experiments on 6 and 49 experts, and furthermore 8 and 16 layer expert configurations are presented.

Training details All networks are trained on 800 images from the DIV2K [1] dataset. We train with the Adam optimizer [18] and the One-Cycle learning rate policy [26]. Firstly, we train expert networks for 3×10^4 iterations with HR patches of 128×128. Secondly, the ISE module is trained for 2×10^5 iterations with input patches of 128×128. The maximum learning rate in the one-cycle policy is set to 3×10^{-4} for expert networks and ISE. Finally, we train KEN for 10^5 iterations with a maximum learning rate of 10^{-3}. The batch size is set to 32 for all mentioned networks. At test time, we fine-tune the selected expert on the input image. The final results are achieved after 1000 iterations of fine-tuning with a single batch per iteration and maximum learning rate of 10^{-5}. We trained all networks on a single RTX 2080 Ti GPU.

4.2. Evaluation on synthetic datasets

To evaluate the quantitative accuracy of SR methods, we experiment with the DIV2KRK benchmark introduced by [2]. It contains 100 validation images from DIV2K [1] that are degraded with random anisotropic Gaussian kernels. The kernel parameters λ_1 and λ_2 are uniformly distributed in [0.6, 5] and θ in $[-\pi, \pi]$. A uniform multiplicative noise (up to 25% of each kernel pixel) is applied to deviate the kernel from a regular Gaussian. We follow the same degradation setting to generate Flickr2KRK and Urban100RK benchmarks from Flickr2K [28] (first 100 images) and Urban100 [13] datasets.

The PSNR and SSIM values for the Y channel are reported in Table 1. SAN [7] represents the methods trained with bicubic degradation. Although it provides outstanding performance on bicubic downsampled images, it does not perform well on realistic images with complicated degradation kernels. KernelGAN [2], DBPI [16] and DualSR [9] are methods that only rely on internal learning. They perform better than SAN, but their performance is inferior to methods like IKC [12] and DAN [14] that are trained on external data. The main reason is that the information in a single image is too limited to train an SR network from scratch. For KernelGAN, we use ZSSR [25] as the SR method. We also use the kernels estimated by KernelGAN for non-blind SR methods like MZSR [27] and USRNet [34]. MZSR starts internal learning from a transferable initial point and provides slightly better performance than KernelGAN. We observe that, the quality of USRNet drops significantly when the estimated kernel deviates from the GT kernel. DAN trains a single large model for multiple degradations on external datasets. Although it performs better than internal learning-based methods, its performance is inferior to our MoESR method.

For DIV2KRK dataset, the PSNR of MoESR is 0.13 dB and 0.93 dB higher than DAN for scales $\times 2$ and $\times 4$ respectively. This is despite the fact that each expert in MoESR is much simpler w.r.t. to DAN. It suggests that having several small networks, each specialized for a specific kernel, performs better than a single large network that supports multiple kernels. This holds even more for $\times 4$ SR where

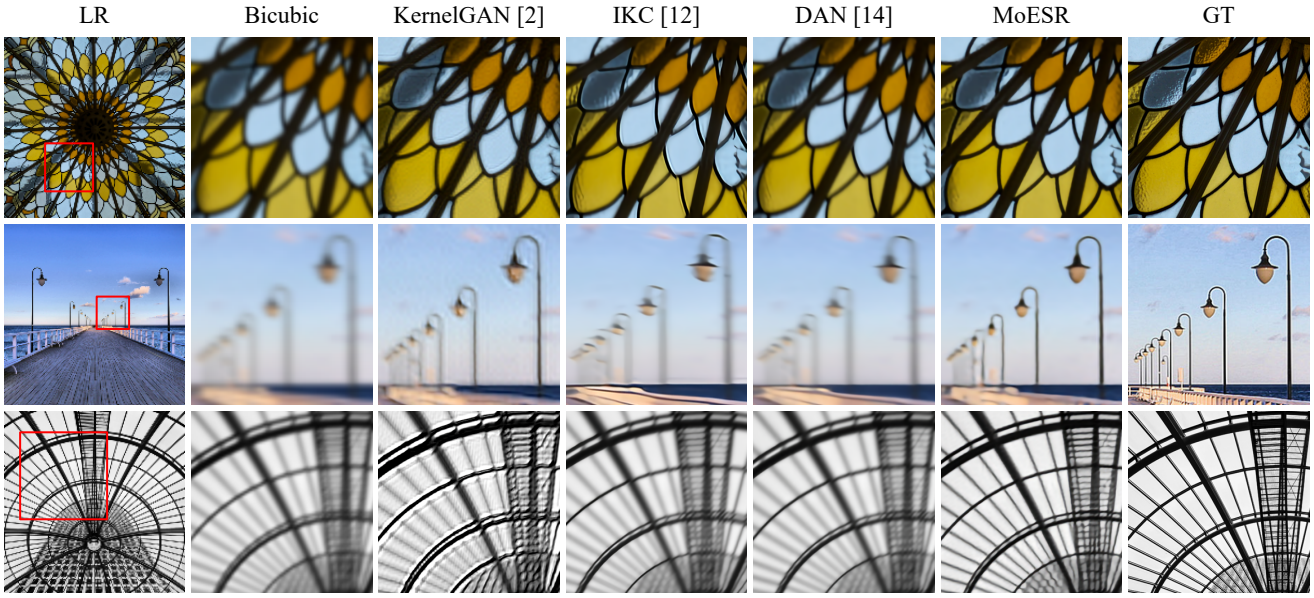


Figure 2. Visual comparison of $\times 4$ SR on synthetic images. Images are from DIV2KRK [2], Flickr2KRK and Urban100RK respectively.

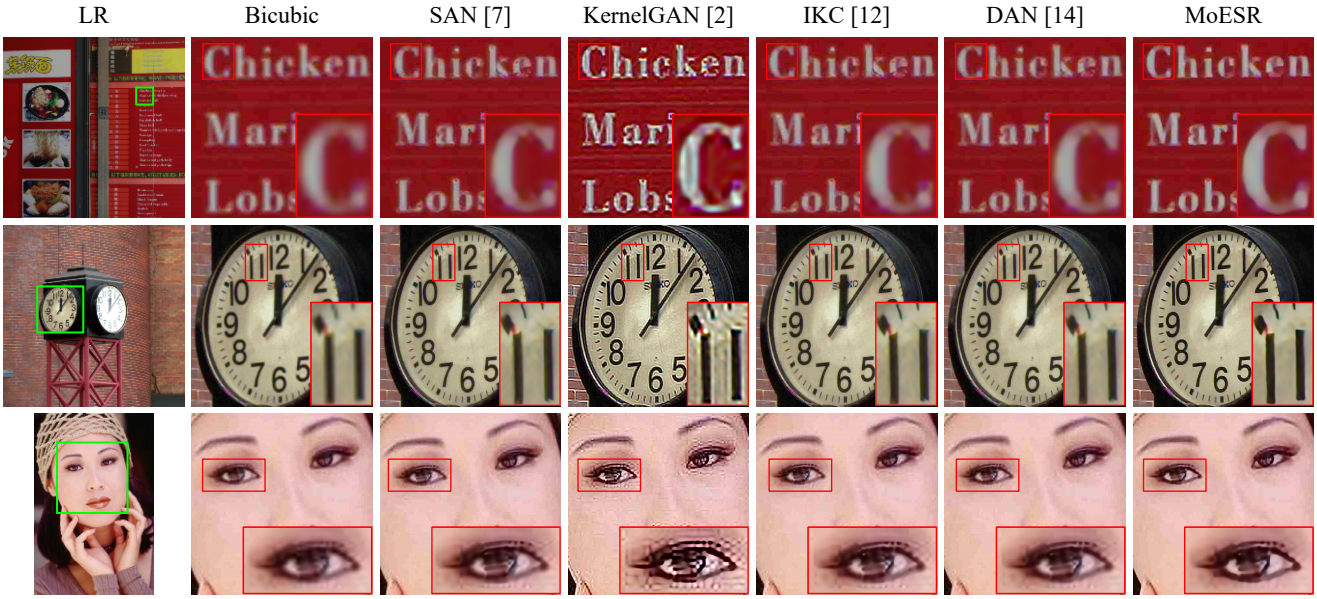


Figure 3. Visual comparison of $\times 4$ SR on real-world images. The first two images are from RealSR [4] and the last image is from Set5 [3]. High quality images are available in the supplementary material.

the LR to HR conversion is more complicated. This conversion especially for multiple kernels is difficult to learn by a single network. Part of our improved performance is due to the use of internal learning during fine-tuning. The quality difference between MoESR and DAN for $\times 4$ Urban100RK is smaller. This is because the smaller Urban100RK images contain less information to support internal learning. We report the MoESR quality without fine-tuning (MoESR w/o FT), for which it is still able to outperform several state-of-the-art methods, especially in terms of SSIM. Note that fine-tuning degradation-aware networks (e.g DAN) at the test time is time-consuming and impractical due to their complex architecture and large number of network parameters.

Visual comparison of state-of-the-art blind SR methods is shown in Figure 2. Our MoESR reconstructs the edges clearly and generates sharper and cleaner images compared to DAN and IKC. Results generated by the combination of KernelGAN and ZSSR are over-sharpened and contain severe artifacts.

Method	$\times 2$	$\times 4$
ZSSR [25]	32.44	27.53
DualSR [9]	32.72	-
USRNet [34]	32.23	27.80
DAN [14]	32.55	26.93
MoESR	33.35	28.83

Table 2. PSNR values on DIV2KRK when ground-truth kernels are given.

4.3. Evaluation on real-world images

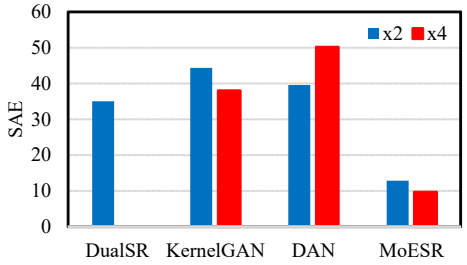
To evaluate the generalization of MoESR to real-world SR, we conduct experiments on real images. As it is mentioned by previous work [25, 36, 14], adding a small amount of Gaussian noise to the LR image during training improves the performance of SR methods on real images significantly. We add noise to the downsampled input image during fine-tuning step. This way, we do not need to retrain the experts and we can control the intensity of denoising for each image by adjusting the standard deviation of added noise. In this experiment, we fixed the standard deviation of noise to 10 for all images.

Figure 3 shows visual comparison of SR methods on real images. The results generated by KernelGAN are too sharp and suffer from unpleasant artifacts. The results of IKC and DAN are cleaner but they are still blurred and contain some artifacts. However, MoESR generates realistic sharp images with less artifacts. For example, in the last image (*woman*) there are some slight compression artifacts in the LR image. These artifacts are highlighted by all KernelGAN, IKC and DAN methods. However, MoESR reduces the artifacts without smoothing the output image. It indicates that MoESR can be adapted to different degradation settings at the test time.

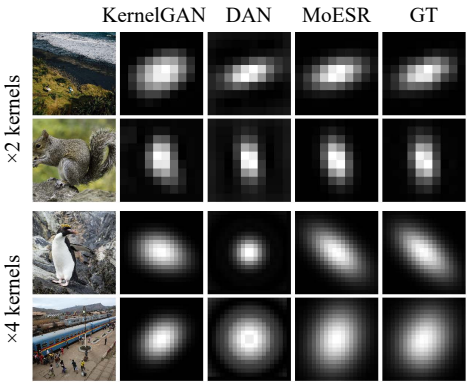
5. Discussion

5.1. Analysis on estimated kernels

Inaccurate kernel estimation can lead to oversmoothing or oversharpening of the super-resolved image. To evaluate the accuracy of our kernel estimation, we calculate the Sum Absolute Error (SAE) between the predicted and GT kernels. Since the kernels of DAN are compressed by Principle Component Analysis (PCA), we firstly convert them into the original kernel space using transpose of PCA matrix. Figure 4 (a) shows the results on DIV2KRK dataset. It indicates that for both $\times 2$ and $\times 4$ SR, the error of kernels estimated by MoESR is significantly lower than other methods. Examples of estimated kernels are shown in Figure 4 (b). Although KernelGAN and DAN can estimate the kernels properly for $\times 2$ SR, they tend to fail for $\times 4$ SR. This explains the large performance gap between MoESR and other methods for $\times 4$ SR in Table 1.



(a) SAE of estimated kernels



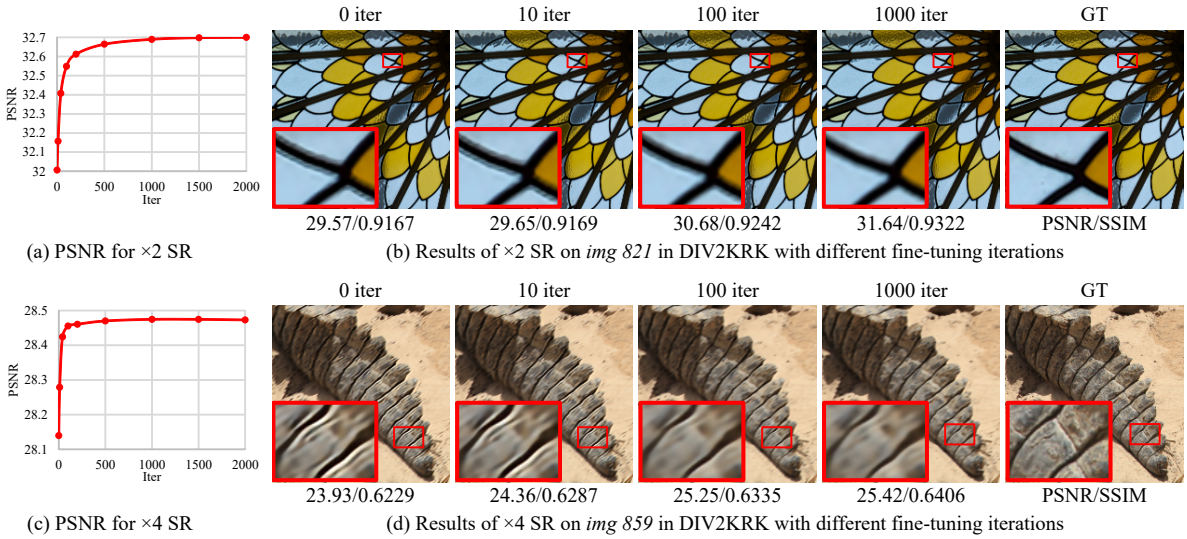
(b) Examples of estimated kernels

Figure 4. Sum Absolute Error (SAE) and visual comparison of estimated kernels for DIV2KRK.

We also evaluate MoESR in non-blind setting when GT kernels are provided. To do that, we bypass the kernel estimation and only fine-tune the best kernel-specific expert with GT kernel. For DAN, as suggested in [14], we run only one forward propagation of Restorer module. The PSNR results are shown in Table 2. Note that MoESR can further improve when the GT kernel is provided. The performance gap versus others for GT kernels suggests that the superiority of MoESR is not only due to better kernel estimation, but also exploiting internal and external learning.

5.2. Analysis on number of fine-tuning iterations

By fine-tuning, we adjust the selected expert network to the estimated kernel and also leverage the advantage of internal learning. To evaluate the effectiveness of fine-tuning, we plot average PSNR on DIV2KRK dataset with respect to the number of fine-tuning iterations. Figure 5 (a) and (c) shows that the PSNR firstly increases with higher iterations and then converges after around 1000 iterations. For $\times 4$ SR it converges even faster because the LR image is smaller and there is less information achievable from internal learning. As shown in Figure 5 (b) and (d), fine-tuning eliminates the artifacts generated due to inaccurate kernel estimation and in fact, it makes the model robust to kernel estimation errors. Fine-tuning with 1000 iterations on DIV2KRK brings an PSNR improvement of 0.68 dB and 0.34 dB for $\times 2$ and



(a) PSNR for ×2 SR

(b) Results of ×2 SR on *img 821* in DIV2KRK with different fine-tuning iterations

(c) PSNR for ×4 SR

(d) Results of ×4 SR on *img 859* in DIV2KRK with different fine-tuning iterations

Figure 5. Quantitative and qualitative results with different fine-tuning iterations for $\times 2$ and $\times 4$ SR.

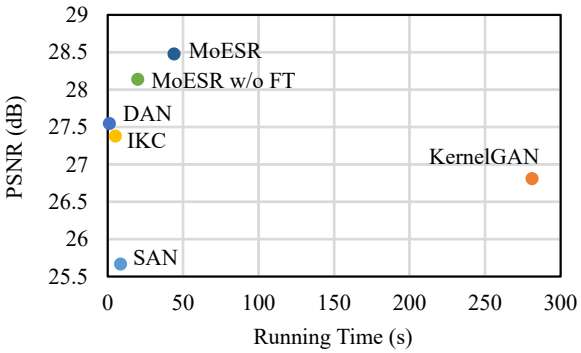


Figure 6. The average PSNR and running time for $\times 4$ SR on DIV2KRK. All methods are evaluated on the same platform with a single RTX 2080 Ti GPU.

$\times 4$ SR respectively. If time is a limitation, the number of iterations can be decreased. For example fine-tuning with 100 iterations gives a PSNR of 32.55 dB and 28.46 dB for $\times 2$ and $\times 4$ SR respectively.

5.3. Limitations

This work mainly focused on SR quality instead of run-time efficiency. The overall MoESR processing time on single images from DIV2KRK is about 34 and 44 seconds for $\times 2$ and $\times 4$ SR respectively. This includes the time of kernel estimation, fine-tuning (1000 iterations) and final image reconstruction. Figure 6 shows the tradeoff between accuracy and run-time for $\times 4$ SR. MoESR is faster than self-supervised methods like KernelGAN [2] but it is slower than degradation-aware networks such as IKC [12] and DAN [14]. In Section 5.2, we demonstrate that MoESR can operate without fine-tuning, which decreases the execution time considerably at the cost of a slight quality loss. Alternatively, the number of experts could be reduced to decrease the execution time as suggested in supplementary material section D.1. We are aware that our method performs sub-optimal when there is substantial noise in the input image. These scenarios require a denoising step.

6. Conclusion

In this paper, we proposed MoESR, a kernel-aware mixture of experts approach for blind SR that exploits both external and internal learning. We employ a group of expert networks each specialized for a specific kernel. For every image, using ISE and KEN modules, the blur kernel is estimated by evaluating the sharpness of images super-resolved by experts. To reduce the impact of small difference between training and test data, the selected expert is fine-tuned on the test image. Quantitative results and visual comparison of generated images demonstrate the superiority of MoESR for $\times 2$ and $\times 4$ SR. We also show that our proposed kernel estimation is far more accurate than other blind SR methods, especially for $\times 4$ where other methods struggle to predict a correct kernel. MoESR focuses mainly on correctly estimating the degradation kernel and therefore it scales-up effectively to 4x without feature hallucination. These properties make MoESR very applicable to the domains of surveillance, medical imaging and microscopy. Our future work should evaluate more advanced SR networks to improve the SR performance. On the other hand, a reduction of compute time per image is very desirable.

Acknowledgment This work was supported by the EDL (`https://efficientdeeplearning.nl`) research programme, which is financed by the Dutch Research Council (NWO) domain Applied and Engineering Sciences.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017.

[2] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. In *Advances in Neural Information Processing Systems*, pages 284–293, 2019.

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.

[4] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3086–3095, 2019.

[5] Honggang Chen, Xiaohai He, Linbo Qing, Yuanyuan Wu, Chao Ren, and Ce Zhu. Real-world single image super-resolution: A brief review. *arXiv preprint arXiv:2103.02368*, 2021.

[6] Victor Cornillère, Abdelaziz Djelouah, Wang Yifan, Olga Sorkine-Hornung, and Christopher Schroers. Blind image super resolution with spatially variant degradations. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH ASIA)*, 38(6), 2019.

[7] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11065–11074, 2019.

[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[9] Mohammad Emad, Maurice Peemen, and Henk Corporaal. Dualsr: Zero-shot dual learning for real-world super-resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1630–1639, 2021.

[10] Linjing Fang, Fred Monroe, Sammy Weiser Novak, Lyndsey Kirk, Cara R Schiavon, B Yu Seungyoon, Tong Zhang, Melissa Wu, Kyle Kastner, Alaa Abdel Latif, et al. Deep learning-based point-scanning super-resolution imaging. *Nature Methods*, 18(4):406–416, 2021.

[11] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. In *2009 IEEE 12th international conference on computer vision*, pages 349–356. IEEE, 2009.

[12] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1604–1613, 2019.

[13] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015.

[14] Yan Huang, Shang Li, Liang Wang, Tieniu Tan, et al. Unfolding the alternating optimization for blind super resolution. *Advances in Neural Information Processing Systems*, 33, 2020.

[15] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[16] Jonghee Kim, Chanho Jung, and Changick Kim. Dual back-projection-based internal learning for blind super-resolution. *IEEE Signal Processing Letters*, 27:1190–1194, 2020.

[17] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[20] Ding Liu, Zhaowen Wang, Nasser Nasrabadi, and Thomas Huang. Learning a mixture of deep networks for single image super-resolution. In *Asian Conference on Computer Vision*, pages 145–156. Springer, 2016.

[21] Yiqun Mei, Yuchen Fan, Yuqian Zhou, Lichao Huang, Thomas S Huang, and Honghui Shi. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5690–5699, 2020.

[22] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–952, 2013.

[23] Pejman Rasti, Tonis Uiboupin, Sergio Escalera, and Gholamreza Anbarjafari. Convolutional neural network super resolution for face recognition in surveillance monitoring. In *International conference on articulated motion and deformable objects*, pages 175–184. Springer, 2016.

[24] Gernot Riegler, Samuel Schulter, Matthias Ruther, and Horst Bischof. Conditioned regression models for non-blind single image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 522–530, 2015.

[25] Assaf Shocher, Nadav Cohen, and Michal Irani. Zero-shot super-resolution using deep internal learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3118–3126, 2018.

[26] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019.

[27] Jae Woong Soh, Sunwoo Cho, and Nam Ik Cho. Meta-transfer learning for zero-shot super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3516–3525, 2020.

[28] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017.

[29] Rao Muhammad Umer, Gian Luca Foresti, and Christian Micheloni. Deep generative adversarial residual convolutional networks for real-world super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 438–439, 2020.

[30] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[31] Yifan Wang, Lijun Wang, Hongyu Wang, Peihua Li, and Huchuan Lu. Blind single image super-resolution with a mixture of deep networks. *Pattern Recognition*, 102:107169, 2020.

[32] Chenyu You, Guang Li, Yi Zhang, Xiaoliu Zhang, Hongming Shan, Mengzhou Li, Shenghong Ju, Zhen Zhao, Zhuiyang Zhang, Wenxiang Cong, et al. Ct super-resolution gan constrained by the identical, residual, and cycle learning ensemble (gan-circle). *IEEE Transactions on Medical Imaging*, 39(1):188–203, 2019.

[33] Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 701–710, 2018.

[34] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3217–3226, 2020.

[35] Kai Zhang, Baoquan Wang, Wangmeng Zuo, Hongzhi Zhang, and Lei Zhang. Joint learning of multiple regressors for single image super-resolution. *IEEE Signal processing letters*, 23(1):102–106, 2015.

[36] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3271, 2018.

[37] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301, 2018.

[38] Yongbing Zhang, Siyuan Liu, Chao Dong, Xinfeng Zhang, and Yuan Yuan. Multiple cycle-in-cycle generative adversarial networks for unsupervised image super-resolution. *IEEE transactions on Image Processing*, 29:1101–1112, 2019.