Reflash Dropout in Image Super-Resolution

Xiangtao Kong[1,2,4*] Xina Liu[1,2*] Jinjin Gu[3,1,4] Yu Qiao[1,4] Chao Dong[1,4 †]
[1]ShenZhen Key Lab of Computer Vision and Pattern Recognition, SIAT-SenseTime Joint Lab,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]The University of Sydney [4]Shanghai AI Laboratory, Shanghai, China
{xt.kong, xn.liu, yu.qiao, chao.dong}@siat.ac.cn, jinjin.gu@sydney.edu.au

Abstract

Dropout is designed to relieve the overfitting problem in high-level vision tasks but is rarely applied in low-level vision tasks, like image super-resolution (SR). As a classic regression problem, SR exhibits a different behaviour as high-level tasks and is sensitive to the dropout operation. However, in this paper, we show that appropriate usage of dropout benefits SR networks and improves the generalization ability. Specifically, dropout is better embedded at the end of the network and is significantly helpful for the multi-degradation settings. This discovery breaks our common sense and inspires us to explore its working mechanism. We further use two analysis tools – one is from a recent network interpretation work, and the other is specially designed for this task. The analysis results provide side proofs to our experimental findings and show us a new perspective to understand SR networks.

1. Introduction

Image super-resolution (SR) is a classic low-level vision task aiming at restoring a high-resolution image from a low-resolution input. Benefiting from the powerful convolutional neural networks (CNNs), deep SR networks [6–8, 23, 25, 27, 29, 57–59] can easily fit the training data and achieve impressive results in a synthetic environment. To further extend their success to real-world images, researchers begin to design blind SR methods [30], which can deal with unknown downsampling kernels or degradations. Recent advances have made significant progress by enriching the data diversity [9, 49, 54, 55] and enlarging the model capacity [33, 48], but none of them has tried to improve the training strategy. The overfitting problem will become prominent when the network scale increases significantly, resulting in a weak generalization ability. Then what kind of training strategy is suitable for the blind SR task? A simple yet surprising answer comes to our mind. It is dropout [20], which is originally designed to avoid overfitting and has been proved effective in high-level vision tasks. In this work, we will dive into the usage of dropout and reflash it in super-resolution.

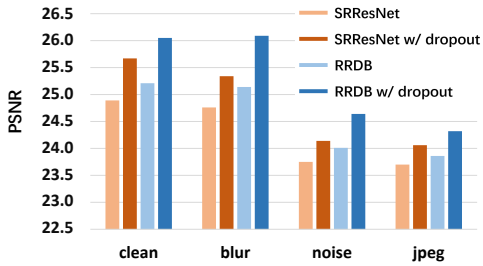


Figure 1. Dropout can significantly improve the performance of the models under the multi-degradation setting. It can even help SRResNet outperform RRDB, while the latter has ten times more parameters. There are the PSNR (dB) results of $\times 4$ SR models on Set5 with different degradations. For example, clean means input LR images without any degradations, noise means input LR images with noise.

Dropout seems to be in conflict with SR in nature. Specifically, the mechanism of dropout is to disable some units and produce a number of sub-networks randomly. Each sub-network is able to give an acceptable result. However, SR is a standard regression problem, where network features and channels all have contributions to the final output. If we randomly discard some features or pixels, the output performance will drop severely. That is why we cannot see the application of dropout in SR, as well as other low-level vision tasks. From another perspective, overfitting is not a severe problem in conventional SR tasks; thus, SR does not need dropout as well. However, this situation changes nowadays. First, overfitting has become a dominant problem for blind SR [30]. Simply increasing the data and network scale cannot continuously improve the gener-

*Equal contributions
†Corresponding author (e-mail: chao.dong@siat.ac.cn)

alization ability. Second, we have obtained a series of analysis tools in the area of network interpretation, assisting us in finding better ways of application.

To study dropout, we begin with its usage in the conventional non-blind settings. After trying different dropout strategies, we can conclude detailed guidance of using dropout in SR. With appropriate usage of dropout, the performance of SR models can improve significantly in both in-distribution (seen in the training set) and out-distribution (unseen) data. Figure 1 shows the performance before and after dropout, where the most significant PSNR gap can reach 0.95 dB. It is worth noting that dropout can help SRResNet even outperform RRDB, while the latter has ten times more parameters. More importantly, adding dropout is only one line of code and has no sacrifice on computation cost. The most appealing part of this paper does not lie in the experiments but in the following analysis. We adopt two novel interpretation tools, *i.e.*, channel saliency map and deep degradation representation [31]) to analyze the behaviour of dropout. We find that dropout can equalize the importance of feature maps, which could inherently improve the generalization ability. There are also some other interesting observations, which all support our experimental results. We believe that these analyses can help us understand the working mechanism of SR networks and inspire more effective training strategies in the future.

2. Related Work

Super-Resolution. CNN-based SR networks [6, 6–8, 18, 23, 25, 27, 29, 57–59] aim to reconstruct a high-resolution (HR) image from its low-resolution (LR) observation. These networks are usually trained in a conventional SR setting where the LR images are produced by the bicubic downsampling. However, overfitting to one degradation leads to poor performance in real-world scenarios. Recently, several works have been proposed to handle multiple degradations and even unknown degradations. Some methods try to first predict degradations explicitly or implicitly and then conditionally reconstruct according to the predicted degradation, *e.g.*, IKC [17], KernelGAN [2], and DASR [48]. These approaches rely on a predefined limited degradation model and still cannot generalize to the data that the degradation model can not cover. Some other methods try to learn end-to-end SR networks that can generalize to a large range of real-world data, *e.g.*, RealESRGAN [49] and BSRGAN [55]. These methods assume that training networks on diverse data can improve generalization capabilities and randomly generate a large amount of training data with different degradations during training. But there is no discussion under which training strategy can maximize the generalization ability. These methods still use the most straightforward direct optimization strategy.

Dropout. Dropout is a regularization technique and is first proposed to address the overfitting problem in classification networks. The key idea is to randomly drop units (along with their connections) from the neural network during training. Therefore, in the training phase, dropout makes only part of the network to be updated each time, and it is an efficient method of averaging sub-networks. Dropout follows a long line of research. A large number of variants have been developed [14, 26, 45, 46] to improve the use of dropout and to adopt dropout in different practical problems. Among them, two works are more relevant to our work. SpatialDropout [45] (channel-wise dropout) formulates a new dropout method to zero out channels from the feature map. When the input has a strong spatial correlation, this method performs better than previous dropout strategies. Different from the original method of adding dropout at the fully connected layers, DropBlock [12] applies dropout to residual blocks (behind convolution layer and skip connection) and then explores using dropout in different parts of networks.

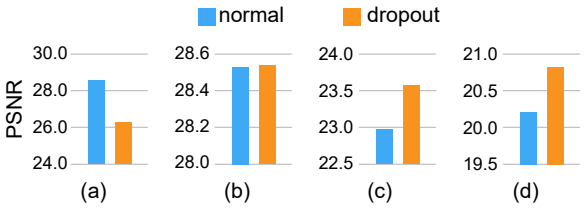


Figure 2. Performance of SRResNet with different settings on Manga109 with $\times 4$. (a) Naive applying of dropout harms SR. (b) Appropriate applying of dropout does not affect SR. (c) (d) Dropout is beneficial for SR in some situations.

Besides, to interpret the success of dropout, various works have attempted to analyze it from different perspectives [4, 10, 19, 22]. Srivastava *et al*. [41] argue that the dropout method samples from an exponential number of different "thinned" networks and approximates the effect of averaging the predictions of all these thinned networks at test time. Some other works attempt to theoretically study the generalization performance for the deep neural network with dropout. For instance, Gao *et al*. [11] point out that dropout can help to reduce the networks' Rademacher complexity. However, most of these improvements, explanations and discussions are aimed at classification tasks. Although dropout has been widely used in classification tasks, its role in super-resolution has not been explored.

3. Observation

We have made some primary attempts to adopt dropout in SR and find that the networks exhibit completely different behaviours under different settings. It is hard to reach a consistent conclusion but will motivate the following study.

Dropout is harmful for SR. This experiment is conducted under the conventional SR setting, where the only degradation is the bicubic downsampling. We adopt the

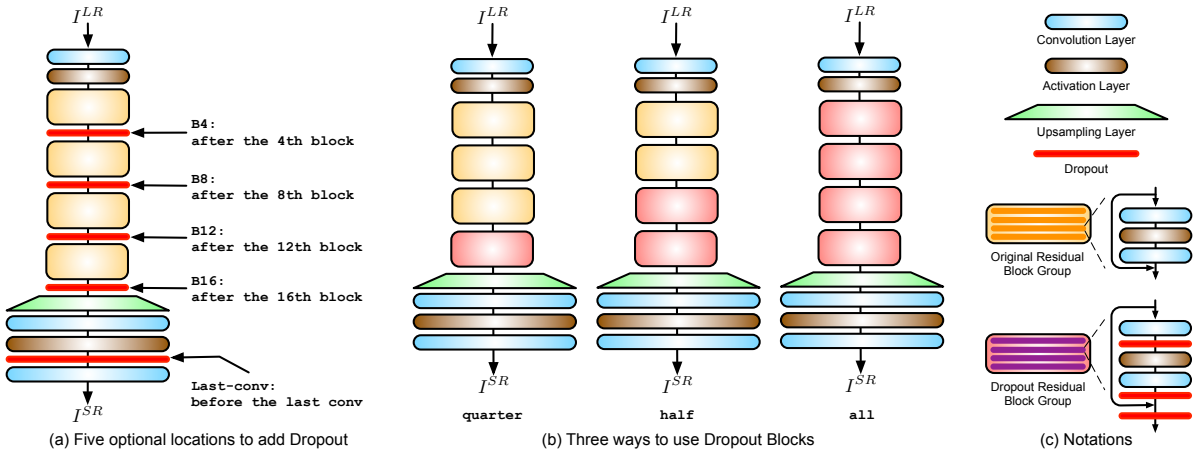


Figure 3. Different ways to apply dropout in SRResNet: (a) illustrates five optional positions where we can add a single dropout layer, marked with red layers; (b) illustrates three ways to add dropout inside the residual blocks; (c) presents the notation.

widely-used dropout strategy – channel-wise dropout [45] (randomly zero out the entire channels) after each convolution layer of SRResNet [27]. As expected, the performance drops dramatically (see Figure 2a). This result exactly conforms to our common sense. It indicates that the regression problem is different from the classification problem. In regression, each element in the network contributes to the final output, which is a continuous intensity value but not a discrete class label. More experiments in Section 5.2 show that most common dropout strategies in classification do not work well on SR.

Dropout does not affect SR. However, we find a special case that does not coincide with the above observation. Under the same setting, we add channel-wise dropout only before the last convolution layer. The final performance is not affected at all, see Figure 2b. This phenomenon is interesting. It indicates that the features in that layer can be randomly masked, which does not influence the regression results. We have also tried to discard a few features during testing, and found no apparent performance drop, see Section 6.1. What happens to those features? Does that mean the regression and classification networks have something in common? This inspires our curiosity.

Dropout is beneficial for SR. The last observation is even more interesting. We find that under the multiple-degradation setting, dropout can even benefit SR. A simple experimental setting is as follows. The training data contain enough degradations, namely Real-SRResNet. We add dropout at the second last convolution layer. The performance is tested on bicubic (seen in the training set) and nearest neighbour (unseen) downsampling dataset. From Figure 2c and 2d, we can observe that dropout improves performance in both in-distribution and out-distribution data. This indicates that dropout improves the generalization ability to some extent. Does this finding have the same theoretical interpretation as the previous one? Can we find other cases where dropout benefits SR? All the observations above can provide us with a clue to recover the effectiveness of dropout in low-level tasks. We will steadily go through this process by detailing the dropout strategies, describing the experiments and revealing the inner working mechanisms.

4. Apply Dropout in SR Network

To explore the application strategies of dropout, we borrow the successful experience from high-level vision tasks. In this section, we will systematically review the feasible implementations of dropout in previous works, and apply them in SR networks. Our study is based on two representative SR networks – SRResNet [27] and RRDB [51]. Our conclusion can be easily generalized to other CNN based SR networks [49, 55, 56], which share similar architectures. As a simple and flexible operation, dropout has many application ways. In general, the effect of dropout mainly depends on two aspects, one is the dropout position, and the other is the dropout strategy. We will discuss them as follows.

4.1. Dropout Position

We explore these potential positions for applying dropout in SR networks through analogy analysis with previous studies in high-level vision. The positions can be mainly divided into three categories. It is very helpful to refer to Figure 3 when reading the following description:

(1) Use dropout before the final output layer. Hinton *et al*. [20] first introduce dropout and apply it at the fully connected layers before the final classification layer. Similarly, we also apply the dropout layer before the output convolutional layer (from the feature channels to the RGB channels). We use `last-conv` to represent this method.

(2) Use dropout at the middle of the network. Many works also try to use dropout at the middle of the network, *e.g.*, after a special convolution layer [45] and at certain locations [13]. Without loss of generality, we split the SRResNet residual blocks (16 blocks) into four groups. Each group consists of four residual blocks. We choose `B4`, `B8`, `B12`, `B16` as representative positions, where the number indicates that dropout is added after which blocks.

(3) Use multiple dropout layers in a residual network. Ghiasi *et al*. [13] suggest that we can apply the dropout layer inside the residual block and use these "dropped residual blocks" multiple times. Figure 3c shows the detail of the "dropped residual blocks". According to their experiments, using this "dropped residual blocks" at the deep locations of the network could generate the best results. We design three different ways to employ "dropped residual blocks" in an SR network and we name them as `all-part`, `half-part` and `quarter-part`. `all-part` means all the 16 residual blocks are replaced by the "dropped residual blocks"; `half-part` means that the second half of the residual blocks are replaced while the others unchanged; and `quarter` represents only the last four residual blocks are replaced.

4.2. Dropout Dimension and Probability

In addition to the position, the dimension of dropout and the probability of dropped channels/elements are also important. Dropout was originally used for fully-connected layers [20]; thus there is no need to determine which dimension to drop. However, after being used in the convolution layers, performing dropout on different dimensions (element and channel) will bring different effects. We also involve different dropout dimensions in our study. The element-wise dropout randomly drops elements among all the feature channels, while the channel-wise dropout only randomly drops the entire channels.

Dropout probability determines the percentage of dropped elements or channels. It is reasonable that too much interference will result in a bad performance, *e.g.*, adding dropout in all blocks or a very high dropout probability. In a classification network, you can randomly drop up to 50% of the elements/channels, not affecting the final result but improving generalization performance. However, this probability may be too large for SR networks as the robustness against information disturbance is much worse than classification networks. To achieve possible benefits without damaging the network, we first test dropout with probabilities of 10%, 20% and 30%. We also include higher dropout probabilities (*e.g.*, 50% or 70%) in multi-degradation SR.

In total, we have eight optional positions, two dimensions and at least three probabilities to apply dropout in SR networks. However, most of them are harmful. Before finally determining our methods, we will study their effects, respectively. Our results indicate that the `last-conv` method with channel-wise dropout does not harm SR networks (see Sec.5.2). Therefore, we use this dropout method to exploit the benefits of dropout for multi-degradation SR.

5. Experiments

5.1. Implementation

SR Settings. There are two commonly-used settings for SR, *i.e*. the single-degradation setting [43] and the multi-degradation setting [49,55]. The most common degradation used in the single-degradation setting is the bicubic interpolation. Training and testing under this single-degradation setting can be used to study the capability or performance of the SR networks. However, SR networks have weak generalization ability under this setting because the network only needs to overfit to a specific degradation.

Unlike the above setting, the multi-degradation setting uses multiple complex degradations to simulate real-world degradations better. With this setting, the SR networks are expected to be effective in real-world scenarios. Overfitting to a specific degradation will no longer be suitable in this setting. The performance of the SR network mainly depends on its generalization ability now. We follow a successful multi-degradation setting called high-order degradation modelling, which is introduced by Wang *et al*. [49]. In their setting, complicated combinations of different degradations (*e.g.*, blurring, downsampling, noising and compression) are used, not one time, but multiple times to generate complex degradations. All the kernels, downsampling scales, noise and compression, are randomly sampled during the training process on the fly. We use the same hyperparameters as Wang *et al*. [49, 50]. As this setting is designed for real-world applications, we use the "Real" prefix to represent models trained in this way.

Training and Testing. We use HR images from the DIV2K [1] dataset for training. During training, L_1 loss function is adopted with Adam optimizer [24] (β_1 = 0.9, β_2 = 0.999). The batch size is 16, LR images are of size 32×32. The cosine annealing learning strategy is applied to adjust the learning rate. The initial learning rate is 2×10^{-4}. The period of cosine is 500k iterations. All models are built using the PyTorch framework [37] and trained with NVIDIA 2080Ti GPUs. For testing, we use Set5 [3], Set14 [53], BSD100 [34], Manga109 [35] and Urban100 [21] as the test sets. We mainly use PSNR to evaluate the performance of the models [15]. The way to generate LR images in different experiments is also different; we will introduce them in the corresponding sub-sections.

Models	Parm.	Set5 [3]		Set14 [53]		BSD100 [34]		Manga109 [35]		Urban100 [21]	
		clean	blur	clean	blur	clean	blur	clean	blur	clean	blur
Real-SRResNet (p=0)	1.5M	24.89	24.76	23.24	23.04	23.89	23.67	22.97	22.59	21.23	21.06
Real-SRResNet (p=0.7)	1.5M	25.67	25.34	23.74	23.44	24.18	23.89	23.58	22.98	21.58	21.31
Improvement		**+0.78**	**+0.58**	**+0.50**	**+0.39**	**+0.29**	**+0.22**	**+0.61**	**+0.39**	**+0.35**	**+0.25**
Real-RRDB (p=0)	16.7M	25.21	25.14	23.73	23.35	24.42	24.22	23.58	23.16	21.57	21.17
Real-RRDB (p=0.5)	16.7M	26.05	26.09	24.02	23.96	24.54	24.44	23.78	23.58	21.89	21.75
Improvement		**+0.84**	**+0.95**	**+0.29**	**+0.61**	**+0.12**	**+0.22**	**+0.20**	**+0.41**	**+0.32**	**+0.58**
		noise	jpeg	noise	jpeg	noise	jpeg	noise	jpeg	noise	jpeg
Real-SRResNet (p=0)	1.5M	23.75	23.70	22.51	22.31	23.01	23.03	22.15	21.75	20.82	20.59
Real-SRResNet (p=0.7)	1.5M	24.14	24.06	22.70	22.64	23.02	23.24	22.57	22.03	20.94	20.89
Improvement		**+0.39**	**+0.36**	**+0.19**	**+0.33**	**+0.01**	**+0.21**	**+0.42**	**+0.28**	**+0.12**	**+0.29**
Real-RRDB (p=0)	16.7M	24.01	23.86	22.93	22.60	23.25	23.33	22.56	22.18	21.16	20.92
Real-RRDB (p=0.5)	16.7M	24.64	24.32	23.17	22.84	23.41	23.42	22.74	22.33	21.26	21.12
Improvement		**+0.64**	**+0.46**	**+0.24**	**+0.24**	**+0.16**	**+0.10**	**+0.18**	**+0.16**	**+0.10**	**+0.19**
		b+n	b+j	b+n	b+j	b+n	b+j	b+n	b+j	b+n	b+j
Real-SRResNet (p=0)	1.5M	23.20	23.44	22.19	22.06	22.65	22.78	21.56	21.25	20.46	20.29
Real-SRResNet (p=0.7)	1.5M	23.47	23.69	22.26	22.38	22.60	22.97	21.81	21.45	20.47	20.53
Improvement		**+0.27**	**+0.25**	**+0.07**	**+0.32**	**-0.05**	**+0.19**	**+0.24**	**+0.20**	**+0.01**	**+0.23**
Real-RRDB (p=0)	16.7M	23.40	23.47	22.45	22.17	22.77	22.95	21.74	21.48	20.57	20.39
Real-RRDB (p=0.5)	16.7M	23.73	23.93	22.57	22.59	22.83	23.15	21.76	21.76	20.53	20.69
Improvement		**+0.33**	**+0.45**	**+0.12**	**+0.42**	**+0.06**	**+0.20**	**+0.02**	**+0.28**	**-0.04**	**+0.30**
		n+j	b+n+j	n+j	b+n+j	n+j	b+n+j	n+j	b+n+j	n+j	b+n+j
Real-SRResNet (p=0)	1.5M	23.17	22.75	22.01	21.74	22.67	22.39	21.37	20.82	20.41	20.09
Real-SRResNet (p=0.7)	1.5M	23.53	23.04	22.26	21.97	22.81	22.51	21.65	21.03	20.63	20.22
Improvement		**+0.36**	**+0.28**	**+0.26**	**+0.22**	**+0.15**	**+0.12**	**+0.28**	**+0.21**	**+0.22**	**+0.13**
Real-RRDB (p=0)	16.7M	23.43	22.81	22.36	21.90	22.90	22.51	21.77	21.05	20.74	20.23
Real-RRDB (p=0.5)	16.7M	23.80	23.18	22.49	22.11	22.98	22.61	21.88	21.20	20.83	20.31
Improvement		**+0.36**	**+0.37**	**+0.13**	**+0.20**	**+0.08**	**+0.10**	**+0.11**	**+0.15**	**+0.10**	**+0.08**

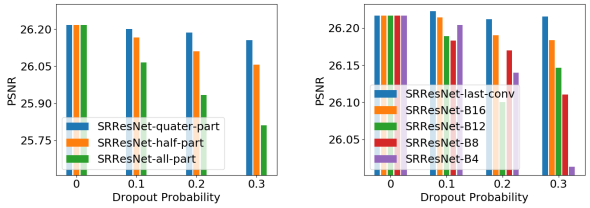
Table 1. The PSNR (dB) results of models with $\times 4$. Each of two columns gives a test set with 8 types of degradations. We apply bicubic, blur, noise and jpeg to generate the degradation, e.g. clean means only bicubic, noise means bicubic $\rightarrow$ noise, b+n+j means blur $\rightarrow$ bicubic $\rightarrow$ noise $\rightarrow$ jpeg. Red texts mean that the performance of Real-SRResNet (with dropout) is better than Real-RRDB (without dropout), half the test sets are red. p indicates the probability of channel-wise dropout using the `last-conv` method.

5.2. How to Apply Dropout in SR Networks

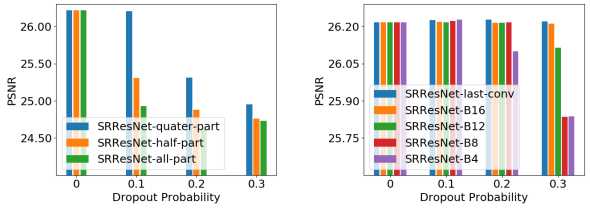
We first study the difference between the dropout methods mentioned in Section 4. We test the performance of applying dropout in different ways under the bicubic single-degradation SR setting. The results are shown in Figure 4. We can obtain the following observations. Firstly, different dropout positions will lead to completely different performances. In the case of using a single dropout layer, we can get better performance when the dropout position comes closer to the output layer. When using multiple dropout layers, we can observe larger performance drop for more dropout layers. Among them, the performance of the `last-conv` method is the best. This observation is consistent with that in classification networks. Secondly, as can be observed from Figure 4a and 4b, element-wise dropout methods tend to degrade the performance, while channel-wise dropout methods generally perform better. Thirdly, in line with expectations, a larger dropout probability will bring worse impact in most cases. In conclusion, we propose to apply channel-wise dropout before the last convolution layer. This position could be easily applied to different network structures, including the vision transformers (included in the supplementary file). We find that this simple and straightforward method can already lead to meaningful and robust results, so we adopt this method in the rest of this paper.

5.3. Dropout in Multi-Degradation SR

Haven the method of applying dropout in SR networks, we next show where we can benefit from it. Dropout is originally proposed to improve the networks' generalization ability, which perfectly matches our need in developing blind SR networks. The following experiments demonstrate that dropout does help to train a better blind SR network under the multi-degradation training setting. In this section, we follow the data generation method proposed by Wang *et al*. [51], which contains complex degradations and their diverse combinations.



(a) Channel-wise dropout



(b) Element-wise dropout

Figure 4. Applying dropout in different methods with various probabilities. The PSNR histograms are obtained by SRResNet on BSD100 with $\times 4$. The training method can be found in Section 5.1. The details of methods can be found in Section 4.

Dropout Helps Learn Better Blind SR Networks. Under the training setting of multi-degradation, the SR network needs to learn how to restore multiple different degradations simultaneously. Directly learning to restore all degradations will make the SR networks perform poorly on individual ones. However, we find that the introduction of dropout can significantly improve the performance of the SR networks under the multi-degradation setting. We test the performance of dropout in some common degradations and complex degradation combinations. Table 1 shows the quantitative comparison of Real-SRResNet and Real-RRDB. We select Gaussian blur with kernel size 21 and standard deviation 2 (denoted by "b"), bicubic downsampling, Gaussian noise with a standard deviation 20 (denoted by "n") and JPEG compression with quality 50 (denoted by "j") as testing degradations. We also include complex mixed degradations that are combined by the above components. For these mixed degradations, we synthesize them in the same order as the training method.

When trained with dropout, Real-SRResNet and Real-RRDB obtain better PNSR performance on almost all the datasets with tested degradations. The maximal improvements on PSNR are 0.78 dB for Real-SRResNet and 0.95 dB for Real-RRDB. The red texts mean the performance of Real-SRResNet (with dropout) is better than Real-RRDB. An appropriate dropout method makes Real-SRResNet have comparable performance with a much larger model Real-RRDB. *One line of code is worth a ten-fold increase in the model parameters.* Figure 5 shows that the models with dropout perform better in content reconstruction, artifact removal and denoising. The models without dropout may remove or enhance some details incorrectly.

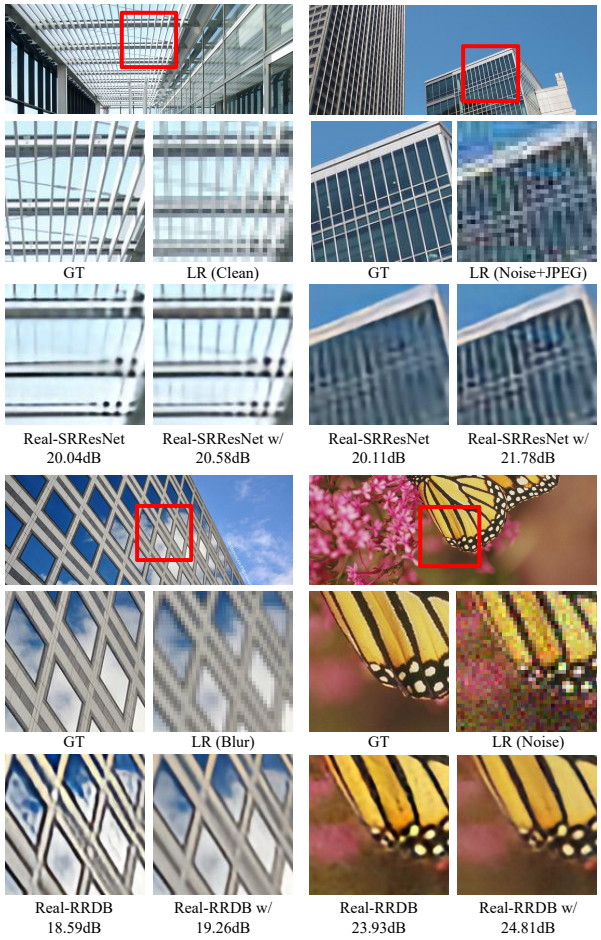


Figure 5. Visual results of representative degradations. We use "w/" to represent the model with dropout in Table 1. (Zoom in for best view)

Models	R-SRResNet (p=0 / 0.7)	R-RRDB (p=0 / 0.5)
mild	16.96 / 17.12 (+0.16)	16.76 / 17.28 (+0.52)
difficult	17.81 / 18.01 (+0.20)	17.65 / 18.15 (+0.50)
wild	17.59 / 17.76 (+0.17)	17.38 / 17.91 (+0.53)

Table 2. The quantitative comparison (average PSNR) of realistic mild/difficult/wild data in NTIRE 2018 SR challenge [44].

Results on Unseen Degradations. Theoretically, the test data listed in Table 1 may be included during training. To better show the generalization ability improvement after applying the proposed dropout method, we also test the networks using the degradations that are unseen for the networks. As shown in Table 2, dropout also shows superiority when testing on realistic mild/difficult/wild data in NTIRE 2018 SR challenge [44]. It proves that dropout could improve the performance on realistic and unseen degradations.

We show more results in the supplementary material, including more results of applying dropout with different probabilities and positions under the multi-degradation set-

ting, more visual effect and the performance of applying dropout in a transformer network called SwinIR [28].

6. Interpretation

After getting the above interesting results, we are very curious about what happens after applying dropout and how dropout improves the network generalization ability. Next, we investigate the dropout method through the lens of network interpretation and visualization.

6.1. Dropout Helps Prevent Co-adapting

Dropout is designed to relieve the overfitting problem by preventing co-adapting in high-level vision tasks [20]. Many tasks have benefited from using dropout. Does co-adapting exist in SR tasks? Are some features more important for reconstruction than others? In this section, we investigate these problems, and find that dropout can help SR networks to prevent co-adapting. The first auxiliary tool we introduce is the channel saliency map (CSM).

Saliency methods [16, 32, 38–40, 42] are widely used in network interpretation research, which aim at highlighting the important decisive factors of the final output. We want to use our CSM method to study different channels' contributions to the final result. It is very similar to the previous saliency methods, but we focus on the feature channels. For an input image I, let $F : \mathbb{R}^{h\times w} \rightarrow \mathbb{R}^{sh\times sw}$ be an SR network with the SR factor s, $F(I)$ be the model output and $F_m(I)$ be the intermediate features at layer m. Similar to LAM [16], a recent work of localizing important pixels to the SR network output, our goal is to find important feature channels. One common method to implement attribution analysis is to calculate the gradient of the output value. Here, we use the summation of image gradient as the attribution target, denoted as $D(I) = \sum \nabla I$. The gradient $\frac{\partial D(I)}{\partial F_m(I)}$ reflects the changes of $D(I)$ caused by each element in $F_m(I)$, denoted as $Grad_{F_m}(I)$. The higher the gradient is, the more influential the element is. Note that $Grad_{F_m}(I)$ has the same size as $F_m(I)$ and also consists of multiple channels. We remove the sign in $Grad_{F_m}(I)$ through an absolute value operation and normalize all its elements to $[0, 1]$, as we only need the relative magnitude instead of the real values. We visualize each channel in $Grad_{F_m}(I)$ to obtain channel saliency maps. Figure 6 shows the relationship between PSNR decrease and saliency maps. When we mask different feature maps, we can get different saliency maps and PSNR values. Low PSNR value is corresponding to bright saliency map. In the visualization results, a brighter pixel (larger intensity) indicates a larger influence w.r.t. the SR results. It shows some features are significantly more important than others.

A commonly used method called channel ablation [36] (or filters ablation [52]) also speaks to the same thing. In practice, we directly ablate an entire feature channel and see what would happen. We can obtain the importance of each channel by measuring the performance drop once the channel is ablated. For intermediate features $F_m(I)$ with c channels, we have c different choices to zero out an entire channel and then get c ablated results. We use $F'_m(I)$ to indicate a ablated result. To ensure that the total energy of this layer remains unchanged after ablation, each $F'_m(I)$ is normalized with $\frac{Sum(F_m(I))}{Sum(F'_m(I))}$, where $Sum()$ means summing up all pixel values. The amplified intermediate features will continue to participate in forwarding calculation until the final output is obtained. The sharp decrease of PSNR means that the ablated channel contributes more to the output image. A more important channel will correspond to a brighter feature map in Figure 6, this correspondence is coincide with conclusions we have obtained with CSM, that

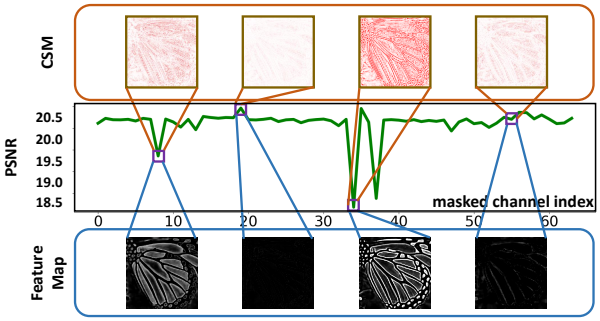


Figure 6. The relationship of PSNR changes, CSM and feature maps. The PSNR of SRResNet without dropout decreases in varying degrees with the ablation of individual channels.

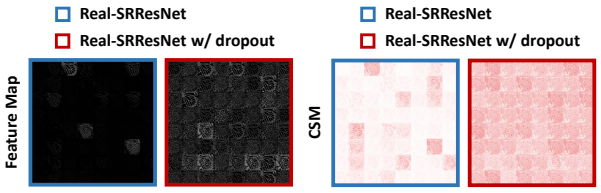


Figure 7. The comparison of feature maps and CSM between Real-SRResNet without dropout and Real-SRResNet with dropout. The features are from the layer where we add dropout.

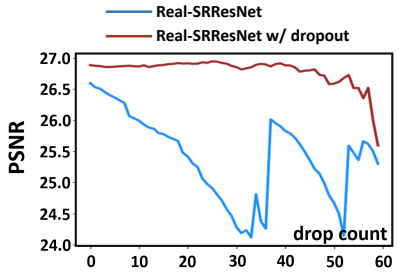


Figure 8. This figure shows PSNR results of channel ablated in turn (from zeroing out one channel to zeroing out 64 channels).

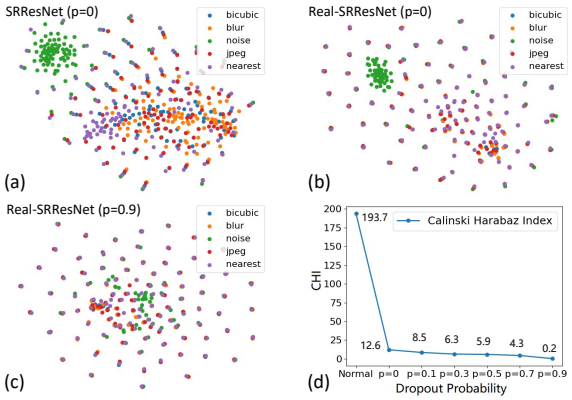


Figure 9. The DDR clusters of SRResNet and Real-SRResNet with different dropout probabilities. $p = 0$ means the networks without dropout. The last subfigure is CHI. $Normal$: SRResNet ($p = 0$). With the dropout probability increases, the cluster distributions of different degraded data are more unanimous.

is, some features are more important than others. Besides, recent works [47, 52] also point some features (filters) are more important.

Then we will show that dropout could prevent co-adapting. In other words, dropout could equalize the importance. First, we visualize the feature maps and CSM comparison in Figure 7. The feature maps and CSM are equalized after adding dropout, it illustrates that dropout could equalize the importance of features. To further prove that, we also zero out each channel in turn and linearly scale the rest features with $\frac{Sum(F_m(I))}{Sum(F'_m(I))}$. Figure 8 shows that the PSNR values of Real-SRResNet without dropout would decrease severely with more channels being ablated, but the performance of Real-SRResNet with dropout keeps unchanged. For a model with dropout, PSNR no longer depends on several specific channels. Even one-third channels of the network are enough to maintain performance. It also show that dropout could equalize the channel importance.

The above experiments demonstrate that dropout can help SR networks to prevent co-adapting.

6.2. Dropout Helps Improve Generalization Ability

The most direct strategy to evaluate generalization ability is to test models in a wide range of data, as described in Section 5.3. It is hard to predict the model's generalization performance for images and degradations that have not been tested – maybe the model happens to perform well on the tested data. However, there are also methods to evaluate generalization ability from the view of interpreting networks' behaviours.

In low-level vision, Liu *et al.* [31] present a concept called deep degradation representation (DDR). Here, we will refer to Figure 9 when introducing DDR. Each point in Figure 9a, 9b and 9c represents an input sample (128×128 image). There are 500 points in each sub-figure. These samples are produced from five degradations, and each degradation corresponding to the same 100 images. DDR reveals that SR networks could classify the inputs to different "degradation semantics". For example, in Figure 9a, points with different colors indicate the inputs with different degradations. Inputs with same degradations (points with same colors) will be clustered. If the obtained clusters are well divided, the network tends to only process specific degradation clusters and ignore other clusters, resulting in poor generalization performance. If the clustering trend is weak, the network has handled all the inputs well. For example, as can be observed from the comparison of Figure 9a and Figure 9b, the clustering degree of the original SRResNet without dropout is larger than Real-SRResNet. This illustrates that a network that has seen more degradations has more remarkable generalization ability.

When it comes to dropout, the cluster distributions of different degraded data for Real-SRResNet ($p = 0.9$, Figure 9c) are closer than Real-SRResNet ($p = 0.1$, Figure 9b). Besides directly observing distribution maps, we could also use Calinski-Harabaz Index (CHI) [5] to measure the separation degree of clusters. Lower CHI means weaker clustering degree, which also indicates better generalization ability. In Figure 9d, one can observe that CHI decreases with the dropout probability increases. It demonstrates that dropout improves the generalization ability of the SR network. This phenomenon is a mutual corroboration with our testing results in a wide range of data. Another interesting observation is that the distribution of samples with noise (the green points in Figure 9) is always the most different one. Reflected in the restoration performance mentioned in Section 5.3, the performance obtained on noisy data is also far from that on clean.

7. Conclusion

In this work, we explore the usage and working mechanism of dropout in SR task. Specifically, we discover that adding dropout using `last-conv` method can significantly improve the network performance in the multi-degradation setting. As for the working mechanism, we find that dropout indeed improves the representation ability of channels and the generalization ability of networks. This is a mutual corroboration of our experimental results. We believe that this work will bring a new perspective to SR tasks and help us better understand network behaviours.

Acknowledgements. This work is partially supported by the National Natural Science Foundation of China (61906184), the Joint Lab of CAS-HK, the Shenzhen Research Program (RCJC20200714114557087), the Shanghai Committee of Science and Technology, China (Grant No. 21DZ1100100).

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 (CVPR) Workshops*, July 2017. 4

[2] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. *arXiv preprint arXiv:1909.06581*, 2019. 2

[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, 5

[4] Xavier Bouthillier, Kishore Konda, Pascal Vincent, and Roland Memisevic. Dropout as data augmentation. *arXiv preprint arXiv:1506.08700*, 2015. 2

[5] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974. 8

[6] Haoyu Chen, Jinjin Gu, and Zhi Zhang. Attention in attention network for image super-resolution. *arXiv preprint arXiv:2104.09497*, 2021. 1, 2

[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. 1, 2

[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. 1, 2

[9] Ruicheng Feng, Jinjin Gu, Yu Qiao, and Chao Dong. Suppressing model overfitting for image super-resolution networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[10] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 2

[11] Wei Gao and Zhi-Hua Zhou. Dropout rademacher complexity of deep neural networks. *Science China Information Sciences*, 59(7):1–12, 2016. 2

[12] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *arXiv preprint arXiv:1810.12890*, 2018. 2

[13] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *arXiv preprint arXiv:1810.12890*, 2018. 4

[14] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *International conference on machine learning*, pages 1319–1327. PMLR, 2013. 2

[15] Jinjin Gu, Haoming Cai, Haoyu Chen, Xiaoxing Ye, Jimmy Ren, and Chao Dong. Pipal: a large-scale image quality assessment dataset for perceptual image restoration. In *European Conference on Computer Vision*, pages 633–651. Springer, 2020. 4

[16] Jinjin Gu and Chao Dong. Interpreting super-resolution networks with local attribution maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9199–9208, 2021. 7

[17] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1604–1613, 2019. 2

[18] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3012–3021, 2020. 2

[19] David P Helmbold and Philip M Long. Surprising properties of dropout in deep networks. In *Conference on Learning Theory*, pages 1123–1146. PMLR, 2017. 2

[20] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 1, 3, 4, 7

[21] 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. 4, 5

[22] Prateek Jain, Vivek Kulkarni, Abhradeep Thakurta, and Oliver Williams. To drop or not to drop: Robustness, consistency and differential privacy properties of dropout. *arXiv preprint arXiv:1503.02031*, 2015. 2

[23] 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. 1, 2

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[25] Xiangtao Kong, Hengyuan Zhao, Yu Qiao, and Chao Dong. Classsr: A general framework to accelerate super-resolution networks by data characteristic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12016–12025, 2021. 1, 2

[26] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Ultra-deep neural networks without residuals. In *Int. Conf. on Learning Representations, arXiv, Toulon, France*, page 1605, 2017. 2

[27] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, 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*, pages 4681–4690, 2017. 1, 2, 3

[28] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *IEEE International Conference on Computer Vision Workshops*, 2021. 7

[29] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Com-*

puter Vision and Pattern Recognition (CVPR) Workshops, July 2017. 1, 2

[30] Anran Liu, Yihao Liu, Jinjin Gu, Yu Qiao, and Chao Dong. Blind image super-resolution: A survey and beyond. *arXiv preprint arXiv:2107.03055*, 2021. 1

[31] Yihao Liu, Anran Liu, Jinjin Gu, Zhipeng Zhang, Wenhao Wu, Yu Qiao, and Chao Dong. Discovering" semantics" in super-resolution networks. *arXiv preprint arXiv:2108.00406*, 2021. 2, 8

[32] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777, 2017. 7

[33] Zhengxiong Luo, Yan Huang, Shang Li, Liang Wang, and Tieniu Tan. Unfolding the alternating optimization for blind super resolution. *arXiv preprint arXiv:2010.02631*, 2020. 1

[34] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001. 4, 5

[35] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017. 4, 5

[36] Ari S Morcos, David GT Barrett, Neil C Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018. 7

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 4

[38] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017. 7

[39] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 7

[40] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 7

[41] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 2

[42] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017. 7

[43] 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. 4

[44] Radu Timofte, Shuhang Gu, Jiqing Wu, Luc Van Gool, Lei Zhang, Ming-Hsuan Yang, Muhammad Haris, et al. Ntire 2018 challenge on single image super-resolution: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 6

[45] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 648–656, 2015. 2, 3, 4

[46] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066. PMLR, 2013. 2

[47] Longguang Wang, Xiaoyu Dong, Yingqian Wang, Xinyi Ying, Zaiping Lin, Wei An, and Yulan Guo. Exploring sparsity in image super-resolution for efficient inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4917–4926, 2021. 8

[48] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Jungang Yang, Wei An, and Yulan Guo. Unsupervised degradation representation learning for blind super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10581–10590, 2021. 1, 2

[49] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *International Conference on Computer Vision Workshops (ICCVW)*, 2021. 1, 2, 3, 4

[50] Xintao Wang, Ke Yu, Kelvin C.K. Chan, Chao Dong, and Chen Change Loy. BasicSR: Open source image and video restoration toolbox. https://github.com/xinntao/BasicSR, 2020. 4

[51] 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)*, pages 0–0, 2018. 3, 5

[52] Liangbin Xie, Xintao Wang, Chao Dong, Zhongang Qi, and Ying Shan. Finding discriminative filters for specific degradations in blind super-resolution. *Advances in Neural Information Processing Systems*, 34, 2021. 7, 8

[53] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010. 4, 5

[54] Jaejun Yoo, Namhyuk Ahn, and Kyung-Ah Sohn. Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8375–8384, 2020. 1

[55] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *arxiv*, 2021. 1, 2, 3, 4

[56] Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3096–3105, 2019. 3

[57] 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. 1, 2

[58] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018. 1, 2

[59] Hengyuan Zhao, Xiangtao Kong, Jingwen He, Yu Qiao, and Chao Dong. Efficient image super-resolution using pixel attention, 2020. 1, 2