Suppressing Model Overfitting for Image Super-Resolution Networks

Ruicheng Feng[1], Jinjin Gu[2], Yu Qiao[1,3], Chao Dong[1]
[1]ShenZhen Key Lab of Computer Vision and Pattern Recognition, SIAT-SenseTime Joint Lab,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
[2]The School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen
[3]The Chinese University of Hong Kong
{rc.feng, yu.qiao, chao.dong}@siat.ac.cn, jinjingu@link.cuhk.edu.cn

Abstract

Large deep networks have demonstrated competitive performance in single image super-resolution (SISR), with a huge volume of data involved. However, in real-world scenarios, due to the limited accessible training pairs, large models exhibit undesirable behaviors such as overfitting and memorization. To suppress model overfitting and further enjoy the merits of large model capacity, we thoroughly investigate generic approaches for supplying additional training data pairs. In particular, we introduce a simple learning principle MixUp [42] to train networks on interpolations of sample pairs, which encourages networks to support linear behavior in-between training samples. In addition, we propose a data synthesis method with learned degradation, enabling models to use extra high-quality images with higher content diversity. This strategy proves to be successful in reducing biases of data. By combining these components – MixUp and synthetic training data, large models can be trained without overfitting under very limited data samples and achieve satisfactory generalization performance. Our method won the second place in NTIRE2019 Real SR Challenge.

1. Introduction

Since the seminal work of employing convolution neural networks (CNNs) for single image super-resolution (SISR) [11, 12], a constantly growing flow of deep learning based methods with different network architectures [13, 21, 24, 22, 37, 18, 44, 43, 3] and training strategies [40, 34, 5, 16] have been proposed to achieve substantial progress in state-of-the-art performance. These methods are usually trained and tested using thousands of high-quality images. Therefore, overfitting is rarely observed when training models with such abundant image pairs. These image pairs are usually generated by pre-defined downsampling methods, such as bicubic. Beyond those pre-defined degraders, in the recent work [7, 6] real captured low-high resolution image pairs are used to train SR models under realistic application settings. However, the amount of such data is often limited (e.g., only 60 image pairs in NTIRE19 Real SR Challenge [1]) because of the high cost of collection and preprocessing of data. This leads to severe overfitting problem for recent deep SR networks. Specifically, the network tends to memorize the training images and generalizes poorly to the test set. For instance, as shown in Figure 1, large models trained on a small dataset quickly deteriorate their generalization performance (see the lower curves). The overfitting problem has largely limited the usage of the advanced SR methods in real-world applications.

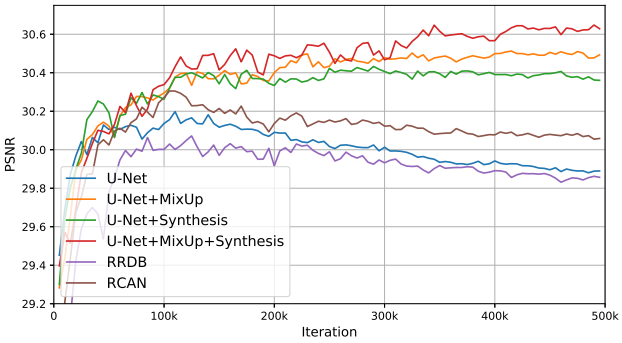


Figure 1: Convergence curves of RRDB[40], RCAN[43], the proposed U-Net and its variants with different data augmentation techniques. The original large models suffer from different degrees of overfitting, while same models trained with either MixUp, data synthesis, or both can achieve satisfactory performance without overfitting.

As an important issue, overfitting has attracted increasingly research interests in high-level vision tasks, such as image classification [10, 15, 20, 8, 39], visual tracking [9, 14], etc. However, overfitting in low-level tasks has received relatively less attention. Due to the different characteristics of low-/high-level tasks, most existing methods that

are suitable for high-level tasks cannot be directly applied to low-level tasks. For example, some network regularization methods, such as weight decay and dropout, do not work effectively for low-level networks. In addition, some popular data augmentation techniques such as label smoothing are also infeasible for low-level tasks as they only work with one-hot labels. In low-level vision community, only limited augmentation methods (e.g., random crop, rotation and flipping) are investigated, which is far from sufficiency for real-world applications.

In this paper, we study the overfitting problem for SR. First, we adopt a simple yet effective data augmentation method called *MixUp* [42] in SR. MixUp uses convex combinations of samples rather than samples themselves to train the SR model. It normalizes neural networks to support simple linear behavior in-between training samples, and leads to better generalization performance (see orange curve in Figure 1). Second, we propose a data synthesis approach with a learned degradation mapping. Concretely, we use deep networks to learn the degradation mapping first, and synthesize new training samples using extra high-quality images. This synthesis strategy reduces the bias of the data by introducing content diversity into the training set (see green curve in Figure 1). The SR models trained with the synthetic data are expected to provide better generalization performance on image contents that do not exist in the original small dataset. By combining the above components – MixUp and synthetic training data, we are able to suppress model overfitting in SR under very limited training samples. Extensive experiments show that either MixUp, data synthesis, or both can suppress model overfitting and encourage better generalization (see upper curves in Figure 1).

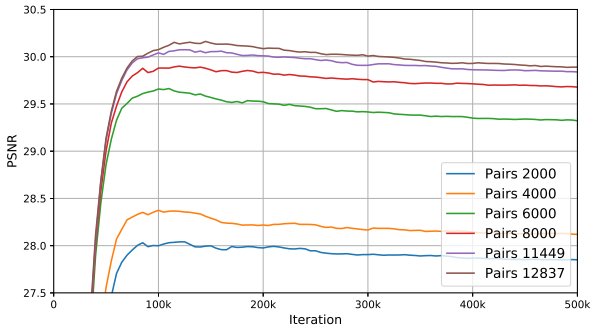
We summarize our contributions as follows: (1) We introduce the MixUp technique into SR for data augmentation. Experiments demonstrate that MixUp could significantly reduce the overfitting problem. (2) We propose a new data synthesis method to suppress model overfitting in SR. It uses the learned degradation mapping to synthesize more training pairs with additional high-quality images. (3) With the proposed data augmentation and data synthesis methods, we construct a network of a general U-Net shape [32] which encourages better generalization ability and achieves satisfactory performance without overfitting. Our method won the second place in NTIRE 2019 Real SR Challenge.

2. Related Work

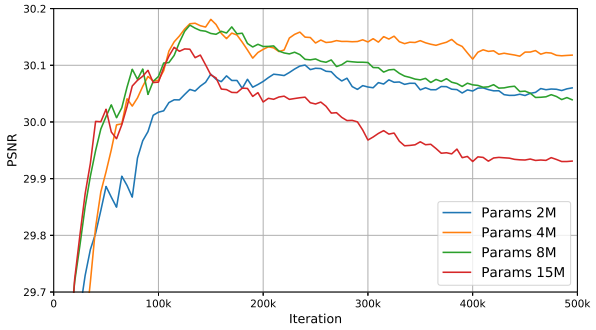
Image super-resolution Recently, learning-based methods have achieved dramatic advantages against the model based methods. With the seminal exploration of employing deep learning in SR task [11, 12], the variational approaches with deep neural networks have been dominated single image SR. Dong *et al.* [13] propose to use a deeper network with low-resolution image as input to learn the SR mapping. Kim *et al.* [21] propose VDSR – a very deep network with residual learning and show the performance improvement by using deep networks. Ledig *et al.* [25] introduce residual blocks into SR network and propose SRResNet, which makes it possible to train deeper networks. Lim *et al.* [26] further expand the network size and improve the residual block by removing the Batch Normalization Layers. Zhang *et al.* [43] propose a deep network with dense connection and Wang *et al.* [40] propose to use residual in residual dense block to improve the training stability and network size. Zhang *et al.* [44] propose residual channel attention blocks and indicate that deeper networks may be easier to achieve better performance than wider networks. As can be seen, most recently successful SR methods employ very deep networks with a large number of parameters, which leads to a high risk of overfitting.

Data augmentation. The method of choice to train on similar but different examples to the training data is known as data augmentation [35]. The most common methods of data augmentation include some basic image processing operations, e.g., random scale, random crop, horizontal/vertical flip and image affine transformation. In addition to the basic image processing operations, Zhong *et al.* [45] propose to augment data by randomly erasing part of the image. Inoue [20] propose to synthesize a new sample from one image by overlaying another image randomly chosen from the training data. Zhang *et al.* [42] propose to synthesize new samples using the linear combination of training samples. DeVries *et al.* [10] improves regularization of networks by masking out square region of training images. Geirhos *et al.* [15] reduces bias toward textures by introducing stylized image data for training. Cubuk *et al.* [8] presents AutoAugment to learn the best augmentation policies from data. Besides, Generative adversarial networks (GANs) have also been used for the purpose of generating additional data [29, 27, 46, 4, 36, 31]. Most of the existing data augmentation methods are proposed and studied for high-level tasks, and there exists few work to study the effects of different data augmentation methods on the low-level task such as SR.

NTIRE 2019 Real Super-Resolution Challenge. This work is initially developed to participate in the NTIRE2019 Real Super-Resolution Challenge [1]. The challenge aims to offer an opportunity for academic and industrial attendees to focus on Super-Resolution applications in real-world scenario. In the challenge, a novel dataset of LR real images with HR real references, where the sizes of LR images are same as its HR counterparts, is provided to challenge participants. These images are collected in natural environments, including indoor and outdoor environments. Different from most SISR tasks [12, 26] using pre-defined degraders, images from this dataset are captured by DSLR cameras, and therefore facilitate researches for real-world applications.



(a) Convergence curves of models trained with different amounts of data.



(b) Convergence curves of models with different complexities.

Figure 2: Illustration on impact of amounts of data and model complexities on validation performance.

However, due to the small volume of data pairs, models suffer from severe overfitting problem. Hence, mechanisms for training large models without overfitting are required to deal with this challenge. We submitted our models and prove that our method are able to suppress model overfitting in SR. Our methods successfully reconstruct HR images from severely degraded real LR images without unpleasant artifacts related to overfitting. Our approach won the second place in the challenge.

3. Methodology

In this section we show the overfitting problem in SR and present our proposed methods. The rest of this section is organized as follows: Sec. 3.1 describes how SR networks overfit on training dataset from NTIRE 2019 Real SR Challenge. Then, we formulate the overfitting issue and data augmentation. Later, Sec. 3.3 and 3.4 introduce the data augmentation method with MixUp and the data synthesis method with learned degradation, respectively. Finally, in Sec. 3.5 we illustrate the network architecture.

3.1. Overfitting in Super Resolution

In this challenge, a new dataset of real LR and HR paired images (RealSR), with the spatial resolution no smaller than 1000×1000, is publicly available. This dataset contains only 60 images for training (See Sec. 4.2 for details). Due to the limited diversity and amount of training data, large models exhibit undesirable overfitting behaviors even when using straightforward data augmentation techniques (e.g. random crop, rotation, flipping). For instance, a well-trained model poorly generalize to the test set and tends to generate unpleasant artifacts on test images.

To start off with right intuitions, Figure 3 illustrates the impact of data volume and model complexity evaluated on the validation set. The validation set consists of 20 images covering contents that do not exist in the training set. In the first setting, we construct a sufficiently large network (with 26M parameters) and train the network with different sizes of data, starting with the first $2,000$ sub-images (from about 10 images) and increasing gradually to all $12,837$ sub-images (cover 60 images). In Figure 2a, we can observe that while all models quickly overfit to training set, increasing amounts of training data will lead to better performance in the training phase. In another setting, we use the whole training set to train models with different sizes, ranging from 2M to 15M. Figure 2b shows that larger models do not necessarily achieve higher PSNR values at the early stage and suffer from severe overfitting if training continues. In contrast, the overfitting problem on small models becomes less severe. This example conveys the central message: overfitting in SR is partially due to the mismatch between data volume and model complexity. To enjoy the merits of large model, we present two methods to remedy such a discrepancy by supplying additional training pairs.

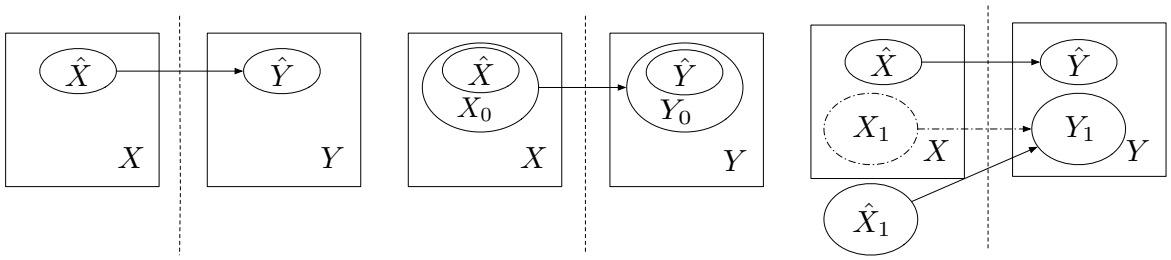
3.2. Problem Formulation

To facilitate the discussion, we first formulate the overfitting problem and data augmentation. Let X, Y be the LR images and their HR counterparts on the true data space, where true data refer to image pairs with the desired degradation function, which can be either pre-defined kernels or unknown real degradations. For each $y \in Y$, we have $x = g(y)$, where g is the *degradation* function mapping Y onto X. In SISR task, given an observation set $(\hat{X}, \hat{Y}) \subset (X, Y)$ as the training set, our goal is to find an inverse mapping function f_θ by optimizing a well-defined loss function $\mathcal{L}$

$$\hat{\theta} = \arg\min_{\theta} \sum_{(x,y)\in(\hat{X},\hat{Y})} \mathcal{L}(f_\theta(x), y). \tag{1}$$

The major risk of this framework is that f_θ may be *biased*, leading to poor generalization ability on unobserved data points. This problem is severe especially when observations are insufficient to cover the true data manifold.

The most widely-used technique to reduce such a risk is data augmentation. Specifically, in the perspective of data augmentation, an addition set (X', Y'), which is beyond the training set $(\hat{X}, \hat{Y})$ but believed inside the true data manifold (X, Y), are introduced for training. In SISR,



(a) Original SISR. (b) MixUp augmentation. (c) Data synthesis.

Figure 3: Illustration on how data augmentation and data synthesis work. (a) The observation set $(\hat{X}, \hat{Y})$ is a subset of the true data set (X, Y). (b) MixUp technique supplies additional training pairs and the augmentation set (X_0, Y_0) covers the observation set. (c) Data synthesis method estimates inaccessible LR images X_1 from extra high-quality HR images Y_1. The estimation $\hat{X}_1$, accompanied with Y_1, constitutes a synthetic dataset and help to reduce the risk of overfitting.

(X', Y') can be obtained by rotating each data pair in $(\hat{X}, \hat{Y})$. We hypothesize that for each $(x, y) \in (X', Y')$, we have $g(y) = x$, indicating that data pairs in observation set and those in augmentation set follow the same degradation mapping.

3.3. Data Augmentation with MixUp

We consider a simple yet effective data augmentation method, *MixUp* [42]. In MixUp, each time we randomly sample two samples (x_i, y_i) and (x_j, y_j) in the set $(\hat{X}, \hat{Y})$. Then we form a new sample by a linear interpolation of these two samples:

$$x' = \lambda x_i + (1 - \lambda) x_j \tag{2}$$
$$y' = \lambda y_i + (1 - \lambda) y_j, \tag{3}$$

where $\lambda \in [0, 1]$ is a random number drawn from a beta distribution $\mathbf{Beta}(\alpha, \alpha)$.

In super resolution, we can assume that the degradation function g is a linear mapping, which can be formulated as $x = g(y) = Dy + n$, where D is the downsampling matrix and n is the noise. If D and n are determinded, we have

$$x' = \lambda x_i + (1 - \lambda) x_j \tag{4}$$
$$= \lambda(Dy_i + n_i)) + (1 - \lambda)(Dy_j + n_j) \tag{5}$$
$$= D(\lambda y_i + (1 - \lambda) y_j) + (\lambda n_i + (1 - \lambda) n_j) \tag{6}$$
$$= Dy' + n', \tag{7}$$

where $n' = \lambda n_i + (1 - \lambda) n_j$. n' is the noise and drawn from the same distribution of n. This property also holds when n is signal-dependent. This indicates that although the MixUp-augmented data pairs have unnatural visual effects, they follow the same degradation model with the true data and can be used to learn the inverse mapping f.

Moreover, MixUp provides a linear neighbourhood of real data, making the learned inverse mapping more robust. With MixUp, we can easily obtain multiple times of data pairs to train the network. As illustrated in Figure 3b, the observation set $(\hat{X}, \hat{Y})$ is a subset of MixUp-augmented dataset (X_0, Y_0) and the latter on has greater cardinality.

Experiments in Sec. 4.3 show that this simple augmentation method can simultaneously suppress overfitting and improve performance.

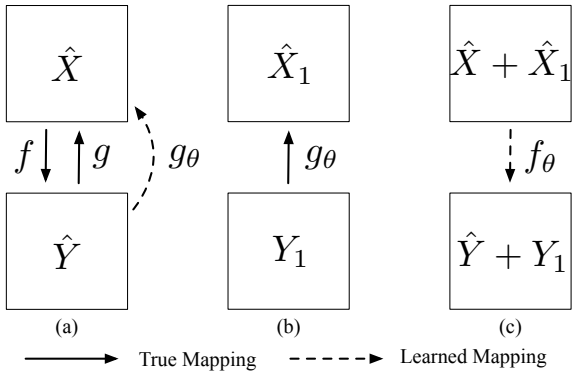


Figure 4: Overview of pipeline for data synthesis. The approach aims to learn two mapping functions $g_\theta : \hat{Y} \to \hat{X}$ and $f_\theta : \hat{X} + \hat{X}_1 \to \hat{Y} + Y_1$. (a) Learn g_θ that $g_\theta(y) \approx g(y)$ and (b) synthesize LR images from Y_1. (c) The vanilla training process on both observed and synthetic data.

3.4. Data Synthesis with Learned Degradation

Beyond MixUp, we also investigate another strategy to provide more training examples – data synthesis via learning degradation process. As depicted in Figure 4, given an observation set $(\hat{X}, \hat{Y})$ comprising images with finite content diversity, there might be a risk of biased sampling from the true data distribution. Formally, let $\hat{P}$ and P be the observed and true data distribution, respectively. For some training pairs $(x, y) \in (X, Y)$ with biased sampling, $\hat{P}(x, y)$ could diverge far from $P(x, y)$. In the extreme, suppose that there is an imbalanced training set with purely text images, then it is unlikely for models trained with such a dataset to generalize well on other contents (e.g., human face, natural scenery, animal, etc.). In practice, a small set $(\hat{X}, \hat{Y})$ is usually both imbalanced and noisy, which in-

crease the risk of overfitting.

To bridge the gap between $\hat{P}$ and P, we propose a data synthesis technique to provide training pairs with higher diversity. As illustrated in Figure 4, given a high-quality diverse HR dataset (e.g. DIV2K [2], Flickr2K [38], etc.) as $\hat{Y}$, the corresponding LR image set X_1 is not accessible since the true degradation $g : Y \rightarrow X$ is unknown. Due to nuisance factors, including blur (e.g. motion or defocus), compression artifacts, color and sensor noise, etc., it is usually impractical to effectively model the true image degradation in real-world scenarios. Rather than managing to model a complicated image degradation process, we propose to use a neural network model denoted as g_θ to learn the degradation g on finite observation set $(\hat{Y}, \hat{X})$.

With well-optimized g_θ, we can obtain estimated LR images $\hat{X}_1$, where for each $\hat{x} \in \hat{X}_1$ we have $\hat{x} = g_\theta(y)$ for $y \in Y_1$. As g_θ is an approximation of g, we expect that for each $y \in Y_1$, the LR counterpart $x \in X_1$ and $\hat{x} \in \hat{X}_1$ should not diverge too far. We will refer to set $(\hat{X}_1, Y_1)$ as the synthetic dataset. With extra data pairs, we turns Eqn. 1 into

$$\hat{\theta} = \arg\min_{\theta} \sum_{(x,y)\in(\hat{X}+\hat{X}_1,\hat{Y}+Y_1)} \mathcal{L}(f_\theta(x), y). \quad (8)$$

During training the SR network f_θ, we treat the synthetic data as additional training data and mix them with the original real data. Both networks f_θ and g_θ have the same architecture (see Sec. 3.5). The main difference is that g_θ takes the HR image as input and generate its LR counterpart, while f_θ is modeling an inverse mapping. The overall pipeline is shown in Figure 4.

This approach is mainly inspired by *Back-Translation* [33, 30] in Neural Machine Translation. In the context of super resolution, [5] proposes to use a GAN to stimulate image degradation and shares a similar motivation. The fundamental differences between this paper and [5] are two-fold: 1) we do not add any generative adversarial component into our PSNR-oriented models; 2) we train both networks with paired image data.

3.5. Network Architecture

As illustrated in Figure 5, the proposed network has a U-Net structure and consists of 4 cascading blocks, each of which has 4 Residual Channel Attention Blocks (RCABs). The spatial resolution of features is decreased 2 times using convolution layers with stride 2, and then it is increased twice via pixel shuffle layers. The basic building block is RCAB proposed in RCAN [43], and the main difference between our model and RCAN is the global network topology. Specifically, motivated by CARN [3], we use both local and global cascading modules to fully utilize hierarchical feature information derived from multiple blocks. The outputs of RCAB are cascaded into higher layers, followed by a single 1×1 convolution layer, all of which serve as cascading blocks. Similarly, global cascading modules have the same topology, where the unit blocks are replaced by cascading blocks. To reduce computational cost, the main branch network works at $1/4H \times 1/4W$ resolution.

4. Experiments

4.1. Technical Details

For all experiments, we implement our models with the PyTorch [28] framework and train them using NVIDIA Titan Xp GPUs. The mini-batch size is set to 16 and the spatial size of cropped patch is 128×128. For initialization, the weights are randomly drawn from zero-mean Gaussian distributions as described in [19]. For optimization, we use Adam [23] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\delta = 10^{-8}$. The learning rate is initialized as 2×10^{-4} and then decayed by half every 10^5 iterations. We train all models for a total of 5×10^5 iterations. We use ℓ_1 loss instead of ℓ_2 as suggested in [26]. We empirically set $\alpha = 1.2$ for MixUp. The SR results are evaluated on PSNR and SSIM [41] on RGB space. For all convergence curves plotted in this paper, we calculate the average PSNR value on the central 1000×1000 patch of each image in validation set.

4.2. Dataset

We mainly train our models on the new Real-SR dataset, denoted as RealSR dataset below. The default splits of RealSR dataset consist of 60 training images, 20 validation images and 20 test images. Evaluation of the trained models is performed on 20 validation images since test images are not publicly available. As described in Sec. 3.4, we also include a prevalent DIV2K dataset [2] as additional training data, since these images cover diverse contents, including objects, environments, animals, natural scenery, etc. Following [26], we use 800 training images as training set.

To prepare training data, we first crop the HR images into a set of 480×480 sub-images with a stride 240 for DIV2K dataset. Similarly, we crop HR images into sub-images of size 200×200 and stride 100 for RealSR dataset. In this manner we have totally $12,837$ and $32,208$ sub-images from RealSR and DIV2K dataset, respectively. To fully utilize the dataset, training images are augmented with random horizontal/vertical flips and rotations. During training, a patch of size 128×128 is randomly cropped from a sub-image.

4.3. Experiments on MixUp

In this section we study the effect of MixUp on different types of dataset. Different from Sec. 4.4, we only use $12,837$ sub-images from RealSR dataset as training set. As described in Sec. 3.3, MixUp serves as a regularization on

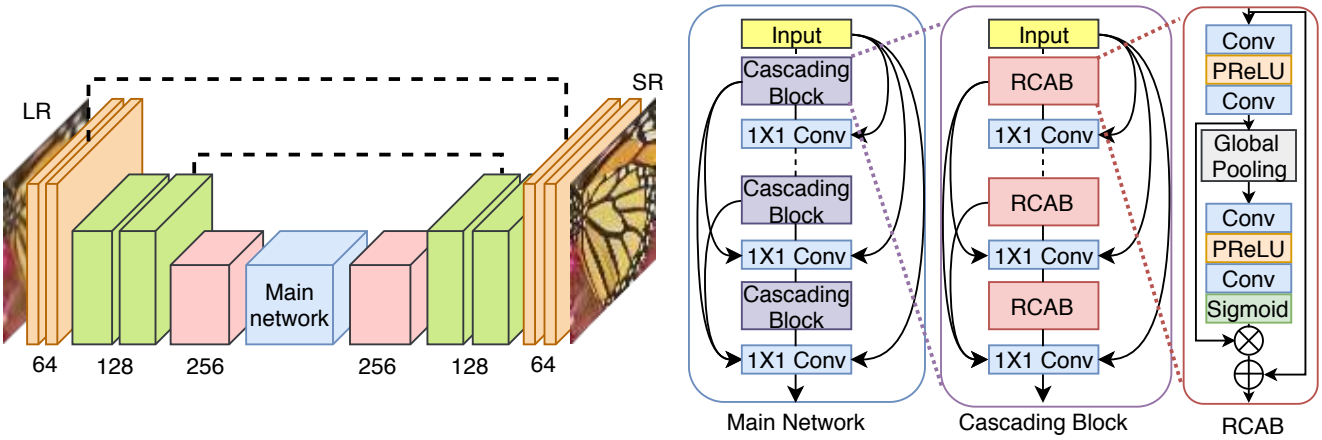


Figure 5: Overall structure of our network.

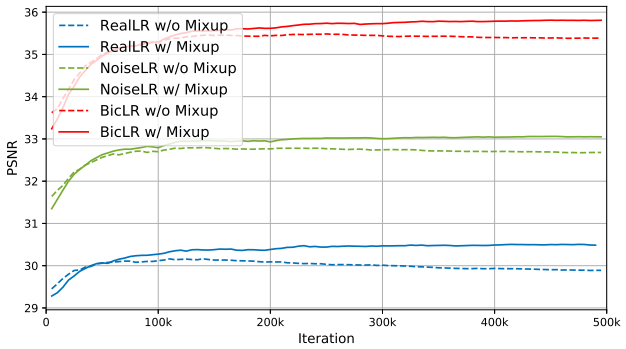


Figure 6: Convergence curves of models trained w/ and w/o MixUp. "RealLR", "NoisyLR" and "BicLR" indicate images sampled from $\hat{X}_{real}$, $\hat{X}_{noise}$ and $\hat{X}_{bic}$.

Figure 7: Convergence curves of SR networks trained on observed data combined with different amounts of synthetic data.

data manifold. To verify the effectiveness of this regularization on various types of degradation, we study three settings by generating LR from HR images as follows:

- Real LR images from RealSR training set
- Bicubic downsample HR images with a factor $4\times$ and then upsample to the original resolution.
- Bicubic downsample HR images with a factor $4\times$ and then upsample to the original resolution, with realistic noise [17] added to LR images.

Similarly, the corresponding validation set is constructed in the same manner for each setting. We denote the LR images as $\hat{X}_{real}$, $\hat{X}_{bic}$ and $\hat{X}_{noise}$, which have the same ground truth $\hat{Y}$. On three datasets we train models with and without MixUp to investigate effects of MixUp.

It can be observed from Figure 6 that after the first learning rate decay (100K), models trained without MixUp quickly deteriorate their validation performance due to overfitting, while those with MixUp keep the same validation accuracy until termination. In super-resolution task, MixUp significantly reduces overfitting and guarantees robust training.

4.4. Experiments on Data Synthesis

In the scope of this section, we mainly use $12,837$ sub-image pairs from RealSR dataset as the observation set and $32,208$ HR sub-images from DIV2K dataset for data synthesis. We first train the degradation model g_θ with $12,837$ training sub-image pairs and the training settings are same as those for f_θ. The model converges at 20K iterations. We aim to provide a systematic analysis of SR networks trained on different synthetic dataset $(\hat{X}_1, Y_1)$ to build a clearer picture about the progressive effects of incremental amounts of synthetic data to the generalization ability.

To validate the assumption that the observation set $(\hat{X}, \hat{Y})$ is biased sampled, we evaluate how the validation error varies while increasing volumes of synthetic data (i.e., higher diversity). Specifically, models are built using a base observation set combined with the augmentation set $(\hat{X}_1, Y_1)$ that starts with 0 sub-image and grows incrementally to all $32,208$ sub-images. Note that the experimental settings degenerate to a baseline scenario without any regu-

larization when $(\hat{X}_1, Y_1)$ contains no sub-image.

According to the results shown in Figure 7, the benefits of adding synthetic data are delaying and reducing overfitting on training set. As expected, adding more and more synthetic data to the training set encourages better generalization. The best combination comprises $45,045$ sub-images ($12,837$ from $(\hat{X}, \hat{Y})$ and $32,208$ from $(\hat{X}_1, Y_1)$), which achieves a PSNR of 30.46dB, 0.25dB better than the baseline model.

4.5. Comparison with the State-of-the-arts

To further investigate overfitting on limited data, we include both light-weight networks (e.g., FSRCNN [13], CARN [3]) and larger networks (e.g., RCAN [43], RRDB [40]) in our comparison. We reimplement these state-of-the-art methods on RealSR dataset. Note that most of the existing methods operate at low resolution and upsample feature maps at the very end of the networks. Therefore, we simply modify the models by downsampling LR images with a stride 4 in the first convolution layer, which is consistent with our U-Net architecture. Throughout experiments, we find existing large models can easily overfit to the training set, and therefore we study early stopped versions of those models to provide a stronger comparison. In contrast, early stopping is not necessary for light-weight networks and our method. We stress that early stopping strategy does not solve the overfitting problem (see also Sec. 3.1), as both training error and validation error are high. With early stopping, a large model will underfit and fail to make full use of model capacity. Specifically, an early stopped large model tends to restore blurry images while a overfitted version generates sharp images with unpleasant artifacts. Following [26], self-ensemble strategy is also applied to further improve generalization performance and the self-ensemble version is denoted with "*".

Table 1 lists the quantitative results (PSNR / SSIM) on RealSR validation set. These results provide two insights: (1) both MixUp and data synthesis can significantly suppress overfitting on limited training data. (2) MixUp and data synthesis are not mutually exclusive, as one can additionally apply MixUp technique on the additional synthetic data to further improve the final performance.

In Figure 9, we show visual comparisons on state-of-the-art networks and our model. For image "cam2_08", we observe that most of the compared methods cannot recover the lines of text and would suffer from blurring artifacts. In contrast, our model can alleviate the blurring artifacts better and recover more details. Similar observations are shown in images "cam2_07" and "cam1_06".

5. Discussion

In this section we further discuss the effectiveness of data synthesis. With a sufficiently large dataset comprising high-quality HR images, one question remains unanswered is how the quality of generated LR images affects generalization ability. Our investigation involves applying various degradation types to HR images from DIV2K training set, while RealSR dataset remains unchanged. LR images are produced with three different degradation processes:

- Add White Gaussian noise with $\sigma = 25$ to HR images.
- Bicubic downsample HR images with a factor $4\times$ and then upsample to the original resolution.
- Construct a network to learn degradation.

The corresponding data pairs constitute a synthetic dataset, where we will refer to these augmentation set as $\hat{X}_{1\_noise}$, $\hat{X}_{1\_bic}$ and $\hat{X}_{1\_net}$. Convergence curves of models trained on different types of augmentation set are shown in Figure 8. We see that the use of synthetic data essentially reduce overfitting problem, compared with the baseline. In addition, LR images from $\hat{X}_{1\_noise}$, $\hat{X}_{1\_bic}$ and $\hat{X}_{1\_net}$ are completely different from each other. The best generalization is reached by the model trained with $\hat{X}_{1\_net}$, indicating that the learned mapping function g_θ among the investigated degradation types would be the most "similar" to the unknown true degradation g. One can also investigate the

Table 1: Model comparisons on validation set. The best and second best results are **highlighted** and underlined, respectively. "+ES" denotes early stopping and "*" denotes self-ensemble strategy.

Method	PSNR	SSIM
FSRCNN[13]	28.3394	0.8254
CARN[3] + ES	29.1620	0.8580
RRDB[40] + ES	29.4581	0.8643
RCAN[43] + ES	29.6299	0.8675
U-Net(Ours) + Synthesis	29.8503	0.8731
U-Net(Ours) + MixUp	29.9055	0.8729
U-Net(Ours) + Synthesis + MixUp	<u>30.0278</u>	<u>0.8753</u>
U-Net(Ours)* + Synthesis + MixUp	**30.1624**	**0.8777**

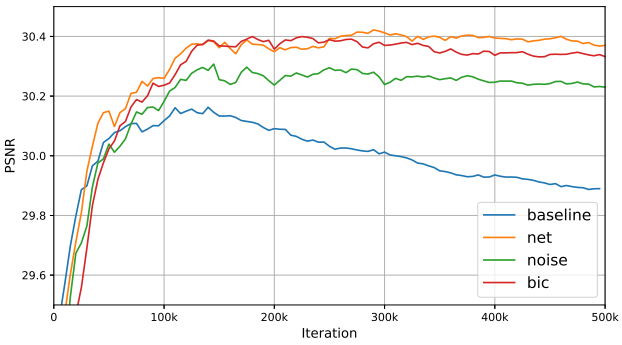


Figure 8: Convergence curves of SR networks trained on observed data combined with different types of synthetic data.

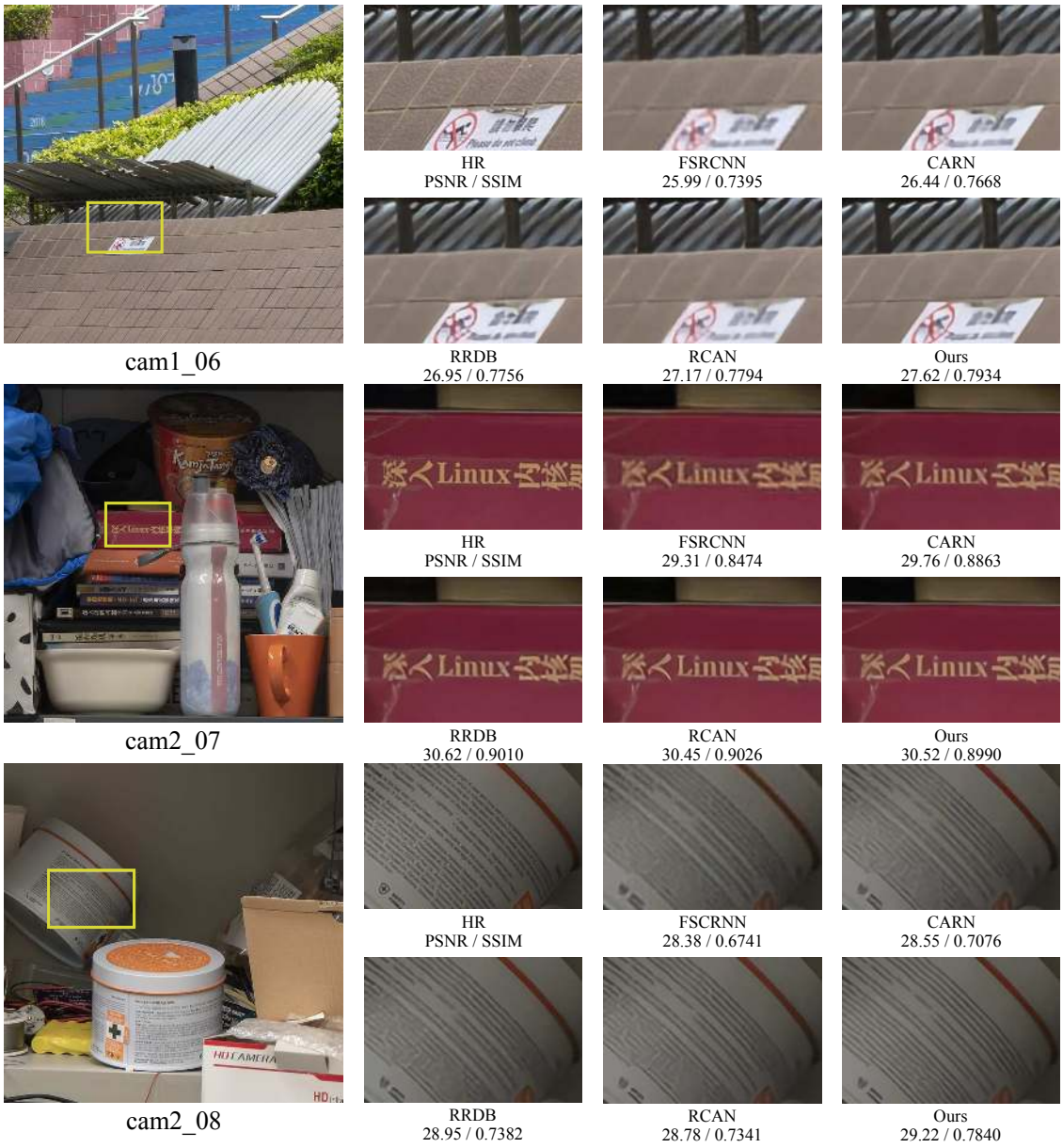


Figure 9: Visual comparison of FSRCNN [13], CARN [3], RRDB [40], RCAN [43] and our method on validation dataset.

sensitivity of SR networks to different kinds of degradation models, which will be left to our future work.

6. Conclusion

In this paper, we propose two simple yet effective methods to reduce overfitting problem in SR networks. Our method won the second place in NTIRE2019 Real SR Challenge. Particularly, we introduce MixUp technique to encourage networks trained with limited data to generalize well. In addition, data synthesis with learned degradation are employed to train models using extra high-quality images with higher content diversity. This strategy proves to be successful in reducing biases of data. By combining both techniques, large models can be trained without overfitting and achieve satisfactory generalization performance. Since the proposed approach is network-independent, it is expected to be easily applied to other network architectures and image restoration tasks. Future work will explore the effectiveness of our approach in more settings.

Acknowledgements. This work is partially supported by National Key Research and Development Program of China (2016YFC1400704), Shenzhen Research Program (JCYJ20170818164704758, JCYJ20150925163005055, CXB201104220032A), and Joint Lab of CAS-HK.

References

[1] Ntire workshop and challenges @ cvpr 2019. https://competitions.codalab.org/competitions/21439#learn_the_details. Accessed: 2019-04-12. 1, 2

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 5

[3] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 252–268, 2018. 1, 5, 7, 8

[4] Antreas Antoniou, Amos J. Storkey, and Harrison A Edwards. Data augmentation generative adversarial networks. *CoRR*, abs/1711.04340, 2018. 2

[5] Adrian Bulat, Jing Yang, and Georgios Tzimiropoulos. To learn image super-resolution, use a gan to learn how to do image degradation first. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 185–200, 2018. 1, 5

[6] 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. *arXiv preprint arXiv:1904.00523*, 2019. 1

[7] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution, 2019. 1

[8] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 1, 2

[9] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1430–1438, 2016. 1

[10] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1, 2

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014. 1, 2

[12] 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, 2016. 1, 2

[13] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*, pages 391–407. Springer, 2016. 1, 2, 7, 8

[14] Jin Gao, Haibin Ling, Weiming Hu, and Junliang Xing. Transfer learning based visual tracking with gaussian processes regression. In *European conference on computer vision*, pages 188–203. Springer, 2014. 1

[15] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 1, 2

[16] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. *arXiv preprint arXiv:1904.03377*, 2019. 1

[17] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. *arXiv preprint arXiv:1807.04686*, 2018. 6

[18] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. Deep backprojection networks for super-resolution. In *Conference on Computer Vision and Pattern Recognition*, 2018. 1

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 5

[20] Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018. 1, 2

[21] 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

[22] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016. 1

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 5

[24] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate superresolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 5, 2017. 1

[25] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, volume 2, page 4, 2017. 2

[26] 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 computer vision and pattern recognition (CVPR) workshops*, volume 1, page 4, 2017. 2, 5, 7

[27] Seongkyu Mun, Sangwook Park, David K Han, and Hanseok Ko. Generative adversarial network based acoustic scene training set augmentation and selection using svm hyperplane. *Proc. DCASE*, pages 93–97, 2017. 2

[28] 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. 5

[29] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017. 2

[30] Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. Investigating backtranslation in neural machine translation. *arXiv preprint arXiv:1804.06189*, 2018. 5

[31] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In *Advances in neural information processing systems*, pages 3236–3246, 2017. 2

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[33] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 86–96, 2016. 5

[34] 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. 1

[35] Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. Transformation invariance in pattern recognitiontangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pages 239–274. Springer, 1998. 2

[36] Leon Sixt, Benjamin Wild, and Tim Landgraf. Rendergan: Generating realistic labeled data. *Frontiers in Robotics and AI*, 5:66, 2018. 2

[37] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4539–4547, 2017. 1

[38] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1110–1121. IEEE, 2017. 5

[39] Tianyang Wang, Jun Huan, and Bo Li. Data dropout: Optimizing training data for convolutional neural networks. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 39–46. IEEE, 2018. 1

[40] 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. 1, 2, 7, 8

[41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[42] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 1, 2, 4

[43] 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, 5, 7, 8

[44] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[45] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017. 2

[46] Xinyue Zhu, Yifan Liu, Zengchang Qin, and Jiahong Li. Data augmentation in emotion classification using generative adversarial networks. *arXiv preprint arXiv:1711.00648*, 2017. 2