Unsupervised Degradation Representation Learning for Blind Super-Resolution with Transformer-based Encoder

Devashish Prasad
Department of Computer Science
Purdue University
prasadd@purdue.edu

Abstract

Super-resolution (SR) has been studied and implemented since the 1990s because of its wide range of applications. It is used in Medical Imaging, CCTV surveillance, Gaming, and Astronomical Imaging, to name a few. Training practical SR models are typically challenging because these models tend to get biased toward training data distribution (type of degradation of low resolution (LR) images). Traditional SR models don't generalize well to real-world unknown test time images.

Recently, researchers have been paying more attention to making the SR models more robust such that they become invariant to the degradation process of the LR image input. It is known as the blind image SR task that aims to super-resolved LR images that result from an unknown degradation process and generate high-resolution (HR) images. In this project, we present a new method that performs better than the recent state-of-the-art Blind SR method. We present a detailed comparative analysis of our method with other methods. These methods were originally trained (by their authors) in different training and testing environments. And so, these pre-trained models cannot be compared directly. To compare these models fairly, in this project, we reproduce and carry out our detailed experiments of training and evaluating these models under common training and testing settings. Code for the project is available at https://github.com/DevashishPrasad/blind-super-resolution

1. Introduction

Image Super-Resolution (SR) refers to the task of enhancing the resolution of an image from low resolution (LR) to high resolution (HR). Image super-resolution has been a topic of interest studied by researchers even before the deep learning era since the early 2000s [2, 18, 30, 31, 39]. After looking at the success of deep learning and CNNs in other computer vision tasks, many people have tried and improved CNN-based techniques a lot for image super-resolution tasks [5, 6, 15, 16]. These studies assume that the LR image is a bicubic down-sampled (down-scaled using a bicubic interpolation algorithm) version of HR. They train CNNs using these assumptions and use a dataset of HR (target y) images paired with their bicubic down-sampled LR (input x) images. The training and testing procedure of these simple CNN-based super-resolution models is shown in Fig 1. The performance of these models is estimated by comparing the generated image similarity with the ground truth HR image. And to compare two image similarities, the two widely used metrics are peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [34]. Acceptable values for wireless transmission PSNR quality loss are considered about 25 dB and more, and many SR models report an average of 26 to 29 dB on standard evaluation datasets of around 100 images. Despite these exciting improvements, these methods tend to fail in many real-world scenarios because of the bicubic down-sampling assumption (or a specific algorithm-based down-sampling assumption). The performance of SR models trained in this way is limited to the kinds of inputs they are trained on, and it drops dramatically when tried on other kinds of inputs.

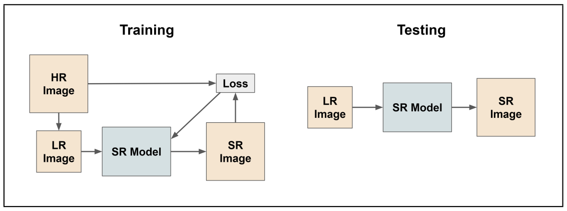

Figure 1. The figure shows traditional (i.e. non-blind) super-resolution models training and testing process. Note that loss (generally L1 loss) is calculated between HR Image ground truth and SR image prediction. And then, the gradients calculated using this loss are back-propagated through the SR Model network to train the weights

The inconsistency between the simplistic image downsampling (image degradation) assumption of existing SR methods and the complex degradations of real-world images has led researchers to build degradation-aware SR models [38]. Over the years, many techniques have been proposed [23], each having its advantages and disadvantages. In this project, we choose to solve the blind SR problem using a specific type of method in which we construct and train a degradation estimation network along with our super-resolution model. The training and testing procedure of this type of blind super-resolution model is shown in Fig 2. The degradation estimation network predicts a unique representation that describes the degradation process. This representation is then concatenated with convolutional blocks of the super-resolution model. To train such a degradation estimation network, we use Equation 1, which can model (represent) any degradation process [32].

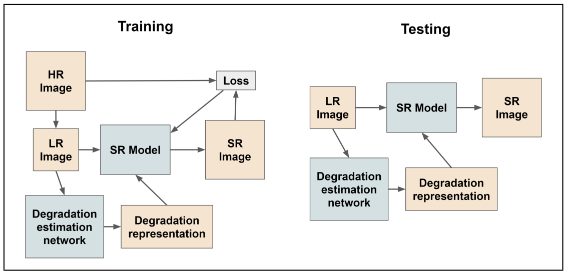
Training
HR Image
Loss
LR Image
SR Model
SR Image
Degradation estimation network
Degradation representation
Testing
LR Image
SR Model
SR Image
Degradation estimation network
Degradation representation

Figure 2. The figure shows the blind super-resolution model training and testing process. We will add a degradation estimation network which will take the LR image as input and produce a representation (mostly a vector or tensor) of the degradation of the LR image. We will train this network too. During the testing time, we will use the output of this trained degradation estimation network and make our SR model degradation aware.

Considering the inverse of the SR task, for a scale factor of s, the classical degradation model of SR assumes the LR image y is a blurred, decimated, and noisy version of an HR image x. Mathematically, this can be expressed by

$$y = (x \otimes k) \downarrow s + n \tag{1}$$

where $\otimes$ represents the two-dimensional convolution of x with blur kernel k, $\downarrow$ s denotes the standard s-fold downsampler (bicubic in our case), and n is usually assumed to be additive white Gaussian noise (AWGN) specified by the standard deviation (or noise level) σ. Using Equation 1, Zhang et al. [38] have been successful at super-resolving LR images for which the degradation (k,s, and n) is already known. However, the blind image super-resolution task aims to recover the HR image from any LR image whose degradation is unknown to the SR model. In other words, a blind SR method first needs to estimate the degradation parameters (k, s, and n) and then super-resolve the image using these parameters. Note that, defining the degradation process like this explicitly, we assume that Equation 1 can represent any degradation. Some studies argue that Equation cannot represent all degradations, and they use other ways to approach the problem and define the degradation process. However, we use Equation 1. because even if it cannot represent all degradations, it can certainly, represent many complex degradations, and solving this problem will be a good direction forward.

In this project, we present our method based on the above framework i.e. using train a degradation estimation network along with our super-resolution model. We take inspiration from a recent work [32] called Unsupervised Degradation Representation Learning for Blind Super-Resolution (UDRL) which uses a contrastive learning-based approach to learn the degradation estimation network. By looking at the success of vision transformers in widespread computer vision tasks, we use a ViT (Vision Transformer) [7] based backbone for our degradation estimation network. We are the first ones to try such an approach to the best of our knowledge. In the next section, we go through the literature review of the blind super-resolution techniques and briefly discuss how transformer-based architectures are applied to super-resolution tasks. We then present our proposed method and also do an ablation study. We compare our method with the original UDRL [32], along with a prominent non-blind baseline EDSR [22].

2. Literature Review

In this section, we give an overview of the current blind SR research and try to justify our choice of considering UDRL as our framework. Specifically, we propose a new taxonomy to categorize existing methods into different classes according to their ways of solving the Blind SR problem.

All Image SR papers can be categorized into two groups, Non-blind SR and Blind SR. And all of the Blind SR papers can be categorized into two subgroups Domain Modeling and Degradation Modeling.

Domain Modeling SR approaches like [4, 9, 26, 33, 35] try to learn an SR model with unpaired images i.e. the model is trained on a dataset in which LR images are not derived from any HR images (i.e. LR mages occur naturally). In such a setting, ground truth HR images are unavailable, and only the input LR images are present. One assumption of this task is that all LR images have the same or similar degradation (they belong to the same degradation domain). There are various ways in which prior techniques have trained such models. But all of these models are trained to implicitly estimate the degradation using the LR images and then learn to super-resolve the LR images using different HR images (or different HR-LR pairs of images) as reference. While test time, it is guaranteed that the model will be super-resolving LR images of this same domain of training.

Degradation Modeling SR approaches do not aim to learn the degradation implicitly. Instead, these methods aim to build a degradation estimator model that can estimate the degradation (based on an explicit assumption like Equation 1) of any given LR image during test time. We categorize these papers into four subgroups 1) LR to Bicubic to SR, 2) Modified Loss Function, 3) Test Time Training, and 4) Degradation Estimation.

First, LR to Bicubic to SR techniques like [13, 29] first try to build an image-to-image translation model that can translate any LR image into a Bicubic down-sampled look-alike image. And then, they use the Non-blind SR techniques to super-resolve this bicubic look-alike image. So no matter what degradation LR images might have, they will be always converted to a bicubic look-alike image, and the traditional Non-blind SR model will not struggle because it was trained on bicubic LR images.

Second, Modified Loss Function techniques like [12, 14, 28] blame the traditional L1 loss function of SR models as the culprit of poor performance in a blind setting. And so, these techniques train Non-blind SR models with very different and novel loss functions or architectures (mainly to get rid of L1 loss) such that the model becomes good at estimating blind degradations.

Third, Test Time Training techniques like [8, 21] train the SR models on various small patches extracted out of the test time image and then super-resolve the whole LR image after training.

Finally, Degradation Estimation based techniques [**?**, 17, 20, 25, 32, 36] are the ones we take inspiration from in this project. These techniques aim first at training a separate model that can estimate the degradation (or a representation of the degradation) and then train a degradation-aware SR model that uses the output of the degradation estimation model while super-resolving images.

Degradation Estimation based techniques? Test Time Training-based models have an overhead of training at the test time that limits their applications in the real world. LR to Bicubic to SR and Modified loss functions techniques don't attain higher metric scores and struggle with a diverse set of degradations. And this is the reason researchers have been paying the most attention to the degradation estimation-based techniques, out of which we have chosen the UDRL [32].

In this project, we modify the degradation estimation network of UDRL and replace it with a small vanilla ViT. We use transformers by looking at [19, 24] as they demonstrate applying a transformers-based backbone to the super-resolution problem.

3. Approach

We choose EDSR [22] as our baseline which is a Non-blind model having a single SR network. And as discussed earlier UDRL [**?**] trains two networks, the first network estimates the degradation of the LR image, and the second network uses this degradation estimation to super resolve the LR. We will call the first network as encoder and the second network the decoder. UDRL uses Equation 1 as the base assumption to model degradation.

For this project, we modify the architectures of both of these models due to computational limitations. One more reason to modify the architecture was to give all of these models almost the same number of parameters. And we create our model with the same number of parameters as well. As the number of parameters used by all three models is similar, we can be very sure that the accuracy of the models is just because of the framework and methodology and not just because of scaling up the number of parameters. Next, we describe details about all three models and how we modified them.

3.1. EDSR

The original Non-blind EDSR model architecture is made by stacking 32 Residual blocks one after the other. Each block has the same shape and size with 3x3 filters and 256 features or channels. In the end, EDSR has a pixel shuffle upsampling layer after all Residual blocks. It rearranges elements in a tensor of shape (C $\times r^2$, H, W) to a tensor of shape (C, H $\times$ r, W $\times r$), where C is the number of channels, r is an upscale factor, H is height and W is width.

The original training strategy for this model involved training on patches of size 48x48 extracted from LR images of the respective HR images. The authors used L1 loss and two augmentation strategies random horizontal flips and random 90 rotations.

We keep the training strategy the same original authors did. However, we reduce the number of blocks to 10 and the number of features to 128.

3.2. UDRL

The blind UDRL model has two networks. Its encoder network (degradation estimation network) learns to predict the representation of the degradation using a contrastive learning-based technique called MoCo [10]. The encoder has a simple architecture with just 6 convolutional layers with kernel size 3x3. The encoder outputs a 1D vector of size 256 which we consider as the representation of the degradation. The output vector of this first vector is concatenated with certain CNN blocks of the second network (SR network/decoder) while super-resolving the LR image. These special CNN blocks are called Degradation Aware (DA) blocks. Each DA block performs a non-trivial operation using two full-connected (FC) layers and a reshape layer to concatenate the representation vector. The decoder network consists of 5 residual groups, with each group comprising 5 DA blocks. Similar to EDSR, the SR network has a pixel shuffle upsampling layer at the end after all residual groups and DA blocks.

The original training strategy involved training UDRL in different settings. It was first trained on noise-free degradations with isotropic Gaussian kernels with different ranges of kernel width. Then, the network was trained on more general degradations with anisotropic Gaussian kernels and noises. Again with different ranges of the covariance matrix. The training strategy involves training the degradation encoder for 100 epochs under a contrastive learning strategy. And then, the whole network (encoder + SR model) is trained for 500 epochs jointly end to end. The overall loss function is defined as L = L1 loss + Contrastive loss.

Again, like EDSR we keep the training strategy the same as the original authors. However, we reduce the number of residual groups from 5 to 3, with each group comprising 3 DA blocks.

3.3. UDRLTE

We call our model UDRLTE (Unsupervised Degradation Representation Learning for Blind Super-Resolution with Transformer-based Encoder). As the name suggests we use a Transformer-based Encoder. We choose a small ViT which has about the same number of parameters as the encoder of UDRL. Our ViT also outputs a 1D vector of size 256 as the representation of the degradation and we use the CLS token output embedding to get this representation. Our ViT takes an image patch of size 48x48 as input and first, it divides it into 16 patches each of size 12x12. All the like positional embeddings, self-attention heads, and other architecture details as exactly the same as from the original ViT model. But it is significantly smaller as our ViT has only 6 self-attention heads and 3 transformer blocks. This is to keep the number of parameters about the same as UDRL encoder. Figure 3. summarizes the overall architecture.

We keep the decoder the same as UDRL's modified decoder with 3 residual blocks and 3 DA blocks. We also keep the training strategy the same.

3.4. Number of parameters

It is important to keep the number of parameters of all three models about the same for a fair comparison. In Table 1. we present the number of parameters each of our architecture has. Note that the total number of parameters of UDRL and UDRLTE include both, the encoder and decoder.

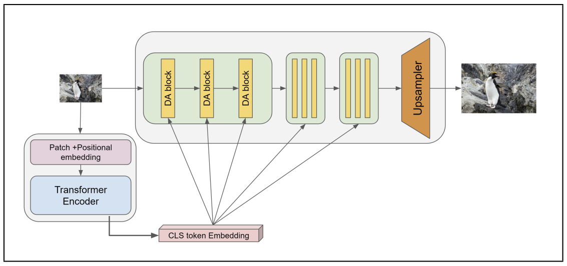

Figure 3. The figure shows the overall schematic diagram of the UDRLTE architecture. There are a lot of hidden details that are omitted from the diagram for the sake of simplicity. The architecture consists of a transformer-based encoder and a degradation-aware super-resolution model (decoder).

Table 1. Comparision of the total number of parameters of EDSR, UDRL, and UDRLTE)

Model	Total number of parameters
EDSR	3844227
UDRL	3883395
UDRLTE	3913717

4. Experiments

4.1. Dataset

The authors of EDSR and UDRL used 800 training images in DIV2K [1] as the training set, and included five benchmark datasets (Set5 [3], Set14 [37], B100 [27], and Urban100 [11], DIV2K Val [1]) for evaluation. All these datasets are used heavily by Non-blind SR techniques. The authors of the DIV2K dataset have released LR-HR pairs (LR generated using bicubic) for various scales, and typically every paper uses x2 and x4 upsampling scales. In a Blind SR setting, we usually ignore the bicubic LRs of DIV2K and generate their own versions of LR images while training. They use some probability distribution to sample Gaussian kernels and noise parameters and then use these parameters to produce LR images (using Equation 1) from their respective HR images. Blind SR models are trained on such a randomly generated dataset in which the SR model (upsampler) does not know what these degradation parameters are. We use this similar training strategy to train our models.

For the common and fair evaluation, we use a test set published by [17] which was generated using five benchmark datasets using different degradation parameters (of Equation 1) for different images randomly. We find this dataset the most appropriate because it was created using a complex degradation kernel of type an-isotropic. Such a difficult dataset benchmark will test all the models to their limit. We only use a 4x upscaling factor as it is more challenging than 2x, and the model needs to really perform well to upscale an image four times. Lastly, we use consistent

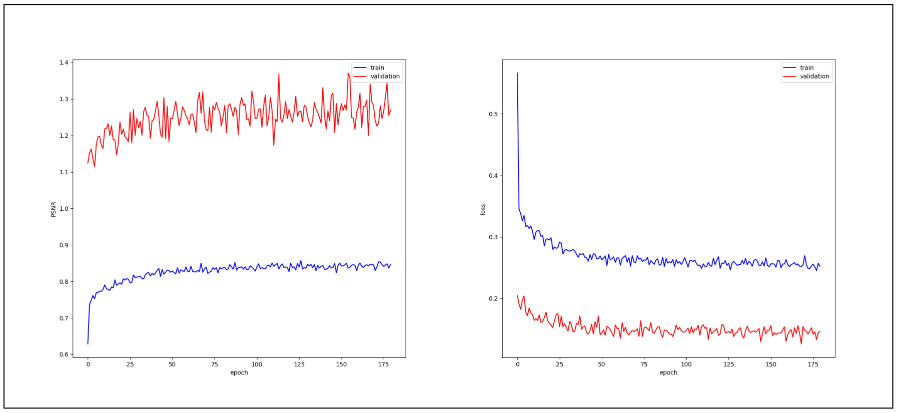

Figure 4. Left: EDSR - PSNR vs Epochs, Right: EDSR - Loss vs Epochs.

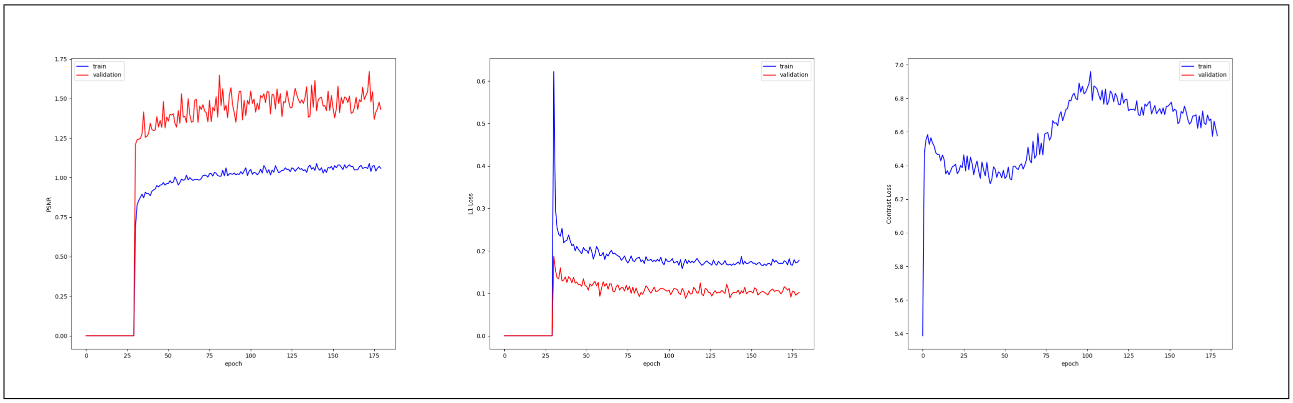

Figure 5. Left: UDRL Encoder + Decoder - PSNR vs Epochs, Middle: UDRL Encoder + Decoder - Loss vs Epochs, Right: UDRL Encoder - Loss vs Epochs

metrics of PSNR and SSIM over whole images (instead of considering just a particular channel) across all models, unlike previous methods.

4.2. Training details

For fair evaluation, we use the exact same training strategy for all three models. We train all the models for 180 epochs. We use almost the same hyper-parameters across all three models and the same l1 loss function. For UDRL and UDRLTE we train encoders for 30 epochs and while the encoder is trained via the contrastive learning strategy the decoder is not touched at all. After 30 epochs we start training both the encoder and decoder in an end to en fashion. Our training scheme is highly inspired by the original UDRL training scheme. We use the same degradation scheme to generate HR-LR pair images across three models.

We also present the learning curves of all three models. Figure 4 shows training curves for EDSR while Figure 5 shows training curves for UDRL, and Figure 6 shows training curves for UDRLTE. The Red color represents the validation set performance or loss curve, and the Blue color represents the training set performance or loss curve.

We can see that as the models are smaller, their performance gets plateaued after a certain number of epochs. It shows that models have reached their limit and cannot improve further. While training UDRL and UDRLTE we see an abrupt rise in the encoder loss near epoch 30, because of starting to use the decoder as well after that point. Encoder + Decoder incurs more loss and it takes time to then gradually start decreasing the loss again.

4.3. Evaluation

Table 2. shows the evaluation results of the three models on the five blind super-resolution benchmark datasets opensourced by [17]. EDSR and UDRL evaluation is straight forward i.e. just take the LR image and make forward prop through the model pr Encoder + Decoder. No matter what dimensions the LR image has, we can always do a forward prop in EDSR and UDRL exploiting their fully convolutional architecture. However, in UDRLTE the transformer-based encoder is not fully convolutional and does take only

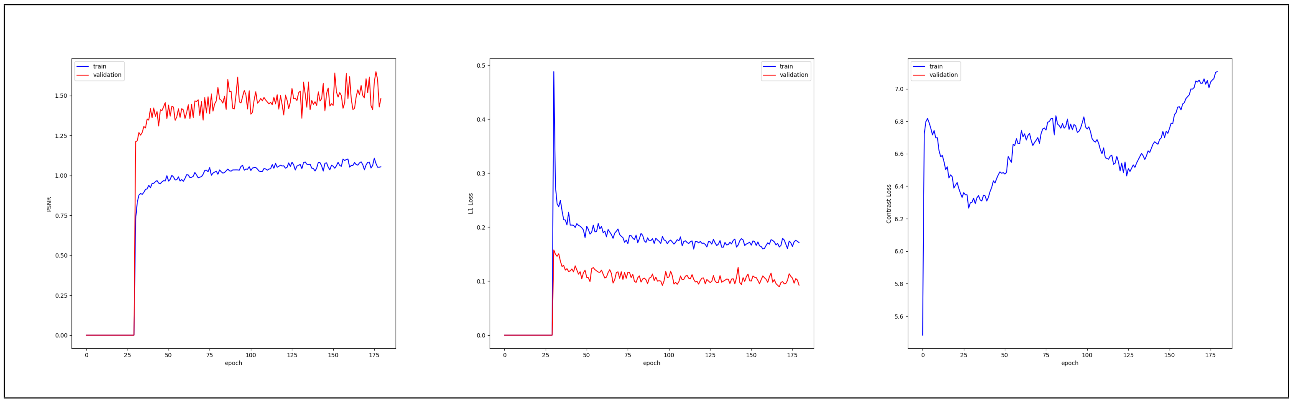

Figure 6. Left: UDRLTE Encoder + Decoder - PSNR vs Epochs, Middle: UDRLTE Encoder + Decoder - Loss vs Epochs, Right: UDRLTE Encoder - Loss vs Epochs

Table 2. PSNR and SSIM of models trained from scratch on 5 standard benchmark datasets. (upscaling factor x4)

Method	BSD 100	Div2K Val 100	Manga 109	Set 14	Set 5	Urban 100
EDSR	24.383/0.63989	26.314/0.74177	23.290/0.74886	22.164/0.60684	26.248/0.75809	21.486/0.62373
UDRL	24.675/0.65564	26.719/0.75946	24.108/0.78270	22.645/0.62794	26.766/0.77929	21.872/0.65189
UDRLTE 1 patch	24.676/0.65765	26.795/0.76241	24.207/0.78812	22.621/0.62942	26.879/0.78363	21.918/0.65404
UDRLTE 10 patch	24.676/0.65762	26.795/0.76236	24.208/0.78809	22.619/0.62935	26.879/0.78364	21.918/0.65399
UDRLTE 0 patch	24.675/0.65756	26.794/0.76230	24.208/0.78804	22.619/0.62928	26.877/0.78351	21.918/0.65399

fixed-size images are input (48x48 to be precise). To solve this issue, we randomly crop a 48x48 patch out of the input LR image and then pass it through the encoder and get a degradation representation. The decoder inference part in UDRLTE is exactly the same as that of UDRL. We also experimented with another scheme in which we randomly crop out 10 48x48 images instead of 1 and then take an average of their representation vectors to get the final representation. Surprisingly such a scheme performs slightly worse than the previous scheme.

In the table, we can clearly see that UDRLTE beats UDRL and EDSR. EDSR being a baseline clearly fails but it is because of the transformer-based backbone that UDRLTE was able to beat the original UDRL.

4.4. Ablation study

Is it possible that the representation of the transformer backbone is being ignored by the UDRLTE? To find the answer to this question we conducted one more experiment in which we set the representation vector as all zeros. If UDRLTE was to ignore this representation then we should have not seen any change in the performance. However, we can see that doing so hurts the performance a little bit. This clearly means that the transformer-based encoder is able to deliver meaningful degradation representation which is helpful for super-resolving the LR image. We show the zero vector representation scheme as UDRLTE 0 patch in the Table 2.

5. Conclusion

In this report, we introduced the blind image super-resolution task and presented a degradation modeling-based strategy to train the CNN-based deep learning models. We trained and evaluated three models namely EDSR, UDRL, and UDRLTE, under the exact same environment across all models. We draw several useful conclusions from our experiments. We showed that UDRLTE having a transformer-based encoder attains the best performance (PSNR and SSIM) as compared to the other two approaches. Our experiments support this claim by giving evidence about UDRLTE being the most effective for Blind SR tasks. And we also did an ablation study where we again see evidence that the UDRLTE encoder is giving us meaningful representations. For our future work, we plan to train and test the models with more training epochs, different model sizes, and more complex noisy degradations.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1122–1131, 2017. 4

[2] J. Allebach and Ping Wah Wong. Edge-directed interpolation. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 707–710 vol.3, 1996. 1

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

In *Proceedings of the British Machine Vision Conference*, pages 135.1–135.10. BMVA Press, 2012. 4

[4] Adrian Bulat, Jing Yang, and Georgios Tzimiropoulos. To learn image super-resolution, use a gan to learn how to do image degradation first, 2018. 2

[5] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 184–199, Cham, 2014. Springer International Publishing. 1

[6] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network, 2016. 1

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. 2

[8] Mohammad Emad, Maurice Peemen, and Henk Corporaal. Dualsr: Zero-shot dual learning for real-world super-resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1630–1639, January 2021. 3

[9] Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. Closed-loop matters: Dual regression networks for single image super-resolution. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5406–5415, 2020. 2

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2019. 3

[11] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5197–5206, 2015. 4

[12] Zheng Hui, Jie Li, Xiumei Wang, and Xinbo Gao. Learning the non-differentiable optimization for blind super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2093–2102, June 2021. 3

[13] Shady Abu Hussein, Tom Tirer, and Raja Giryes. Correction filter for single image super-resolution: Robustifying off-the-shelf deep super-resolvers, 2019. 3

[14] Younghyun Jo, Seoung Wug Oh, Peter Vajda, and Seon Joo Kim. Tackling the ill-posedness of super-resolution through adaptive target generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16236–16245, June 2021. 3

[15] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654, 2016. 1

[16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1637–1645, 2016. 1

[17] Soo Ye Kim, Hyeonjun Sim, and Munchurl Kim. Koalanet: Blind super-resolution using kernel-oriented adaptive local adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10611–10620, June 2021. 3, 4, 5

[18] Xin Li and M.T. Orchard. New edge-directed interpolation. *IEEE Transactions on Image Processing*, 10(10):1521–1527, 2001. 1

[19] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer, 2021. 3

[20] Jingyun Liang, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Mutual affine network for spatially variant kernel estimation in blind image super-resolution, 2021. 3

[21] Jingyun Liang, Kai Zhang, Shuhang Gu, Luc Van Gool, and Radu Timofte. Flow-based kernel prior with application to blind super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10601–10610, June 2021. 3

[22] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution, 2017. 2, 3

[23] Anran Liu, Yihao Liu, Jinjin Gu, Yu Qiao, and Chao Dong. Blind image super-resolution: A survey and beyond, 2021. 2

[24] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. Transformer for single image super-resolution, 2021. 3

[25] zhengxiong luo, Yan Huang, Shang Li, Liang Wang, and Tieniu Tan. Unfolding the alternating optimization for blind super resolution. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5632–5643. Curran Associates, Inc., 2020. 3

[26] Shunta Maeda. Unpaired image super-resolution using pseudo-supervision. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 288–297, 2020. 2

[27] 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. volume 2, pages 416–423 vol.2, 02 2001. 4

[28] Qian Ning, Weisheng Dong, Xin Li, Jinjian Wu, and GUANGMING Shi. Uncertainty-driven loss for single image super-resolution. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16398–16409. Curran Associates, Inc., 2021. 3

[29] Mohammad Saeed Rad, Thomas Yu, Claudiu Musat, Hazim Kemal Ekenel, Behzad Bozorgtabar, and Jean-Philippe Thiran. Benefiting from bicubically down-sampled images for learning real-world image super-resolution, 2020. 3

[30] Jian Sun, Zongben Xu, and Harry Shum. Image super-resolution using gradient profile prior. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 1

[31] Yu-Wing Tai, Shuaicheng Liu, Michael S. Brown, and Stephen Lin. Super resolution using edge prior and single image detail synthesis. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2400–2407, 2010. 1

[32] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Jungang Yang, Wei An, and Yulan Guo. Unsupervised degradation representation learning for blind super-resolution, 2021. 2, 3

[33] Wei Wang, Haochen Zhang, Zehuan Yuan, and Changhu Wang. Unsupervised real-world super-resolution: A domain adaptation perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4318–4327, October 2021. 2

[34] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 1

[35] Yunxuan Wei, Shuhang Gu, Yawei Li, Radu Timofte, Longcun Jin, and Hengjie Song. Unsupervised real-world image super resolution via domain-distance aware training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13385–13394, June 2021. 2

[36] Liangbin Xie, Xintao Wang, Chao Dong, Zhongang Qi, and Ying Shan. Finding discriminative filters for specific degradations in blind super-resolution. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 51–61. Curran Associates, Inc., 2021. 3

[37] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces*, 2010. 4

[38] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[39] Lei Zhang and Xiaolin Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE Transactions on Image Processing*, 15(8):2226–2238, 2006. 1