Efficient Multi-Attention Residual Network for Image Super Resolution

Ufuk Umut ŞENTÜRK
n19133683@cs.hacettepe.edu.tr

Muhammed Emir ÇAKICI
n19238816@cs.hacettepe.edu.tr

Abstract

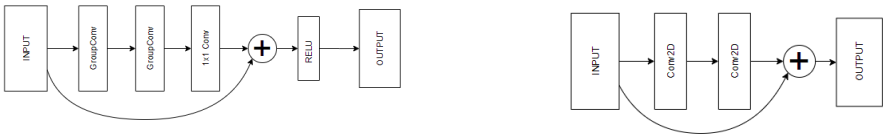
Single image super resolution task is one of the low level vision tasks that probably has multiple solution and known as ill-posed problem. Mainly, many solutions are suitable to reconstruct low frequency, however lack of high frequency features. Our solution is based on end-to-end deep neural network called Efficient Multi-Attention Residual Network(EARN) which has common properties as RNAN[16] such that our baseline solution performs good in terms of evaluation time and metrics. It consists of various attentional mechanisms that combine different aspects of features from high and low level details. Therefore, we propose attention layer called Adaptively Scaled Channel Attention Layer that incorporates with first and second order statistics of the channels by fusing and finding transformation of those descriptor to obtain final descriptor. Our training dataset is DIV2K and testing datasets are B100, Set5, Set14, Urban100 that are known for this specific task by vision community.

1 Introduction

Super resolution task is well-known low level vision ill-posed problem and can be used for many objectives as a side task such as various imagining tasks, surveillance, image compression,restoration of old images etc. Therefore, there is no metric defines fully perceptual quality of the reconstruction. Thus, we try to to extract useful information providing attention at multiple stages to converge human perception. We try to build attention aware features to keep structural information intact and colorize those structural information with strong prior information obtained by residual learning and multi-attentional blocks. Our solution is based on residual attention network for classification task [4] to from hierarchical features and mixed attention depth wise and spatial wise also used in RNAN architecture[16]. However, non-local attention block [7] of RNAN is very much memory consuming and as architecture goes deeper performance drops because of sparse skip connection and low usage of non-local attention block such that at the end and start of architecture. We have also multiple residual attention blocks that differs from RNAN and upsampling module with global and local residual learning. That is why we use efficient residual block that use group convolutions and more skip connections in mask branch Figure 3a, Therefore, our mixed attention is based on local self attention layer[6], instead of non-local attention block, that requires low memory and uses various receptive fields to converge non-local mean to model long-range dependencies, too. Masking block is actually simple U-net architecture. We think that this looks like network in network at multiple stages. Before our efficiently enhanced masking block, we use our adaptively scale channel attention, Figure 3b, to gain strong attentional features being easy to calibrate. Unlike other channel attention layers, we fuse two refined channel descriptors with squeeze-and-excitation block,[5] to obtain better expressivity and flexibility. This operation helps network to become more smooth at early stages and faster convergence compared to RNAN, however, we still believe that we could not fully converge both of this networks. Therefore, this attention block selectively attend channels using this fused and refined depth local descriptors. Besides, we believe that shuffling operation of sub-pixel upsampling [9] convolution with channel attention layers introduces additional stochasticity to the network that

CMP784 Deep Learning, Spring 2020, Hacettepe University, Ankara, TURKEY.

force the network become more robust to the overfitting. Because, we use less number of blocks that might overfit. We see that our SSIM and PSNR scores are promising that drives us to go deeper in this project.



(a) Efficient Residual Block consisting of group convolutions

(b) Basic Residual Block that is used for Trunk Branch and other parts except Mask branch

Figure 1: Different types of residual blocks

2 Related Works

First deep learning study on this area was not that deep at all [8]. It was supervised learning with few layers (3 to 5 layers) and was a big success for that time. It uses pre-up sampling method which means that first it up samples low resolution(LR) image with bicubic interpolation, then interpolation result is fed to super-resolution network. This task can be viewed as image-to-image translation. Thus, we can learn the residual information between LR and HR(high resolution) images. Residual connections are the intuitive and effective way to do this. Also, local and non-local attention modules give better estimation for high frequency details. SRCNN[8] is really plain network that is extended with dense modules [12], gan-based architectures [10][11], residual connections[13][12], attentional modules[14] [16]. Especially, sub-pixel layer that is studied on [9], effectively uses shuffling to up sample feature map instead of transpose convolution operation that suffers corner artifact problems. Such as other inner mechanisms of the network such as de-subpixel[15] layer, group convolutions rather than vanilla convolution have been discovered.

Our architecture has strong relations with the RNAN [16] architecture which is also based on image classification [4] having main objective that is generating attention aware features and also compromising residual learning for identity mapping to increase depth of feed forward network. FFA-Net [3] proposed pixel-attention mechanism for image dehazing with feature fusion at the end of network. Efficient residual block is proposed by CARN [2] for super resolution task for more lightweight model consisting of cascading blocks that are densely connected to each other. Channel attention is introduced Squeeze and Excitation Network[5] which re-calibrates channels adaptively by modelling inter-dependencies between channel-wise features. It is heavily used in RCAN [26] which is based on EDSR [13] - family network which becomes based for SAN [14], RDN[12], RNAN. SAN[14] actually uses second order statistics by calculating covariance matrix based on feature maps with non-local attention block [7]. Besides, IMDN [1] extends this channel attention by summing local descriptor with its standard deviation to enhance extracted information. Our channel attention layer uses more refined information such that standard deviation and mean descriptors and compromising flexibility by fusing and transforming. Stand alone self attention is developed to replace convolution layer with local attention layer to construct pure self-attention vision model [6]. It is used for high level vision tasks and requires fewer parameters. It uses local positional embeddings to model local relative attention which is defined by relative distance to pixel in interest within spatial extent of local window.

3 The Approach

We formulate problem as below;

$$I_{SR} = F_{EARN}(I_{LR}) \tag{1}$$

where I_{SR} is super-resolution output and I_{SR} is low-resolution input. Our approach is based on supervised learning. We try to achieve good result by combining attention modules and residual blocks. In Figure 4, throughout to network, each residual attention block produces same spatial resolution as low-resolution input. We use residual blocks because of the nature of image super

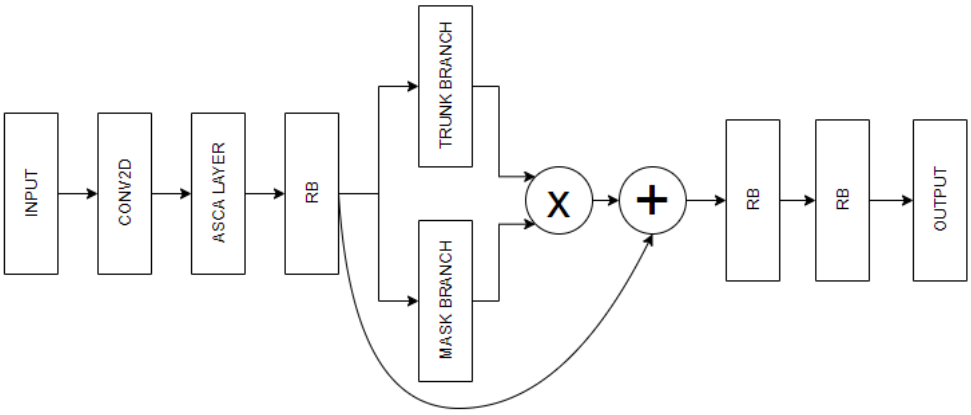
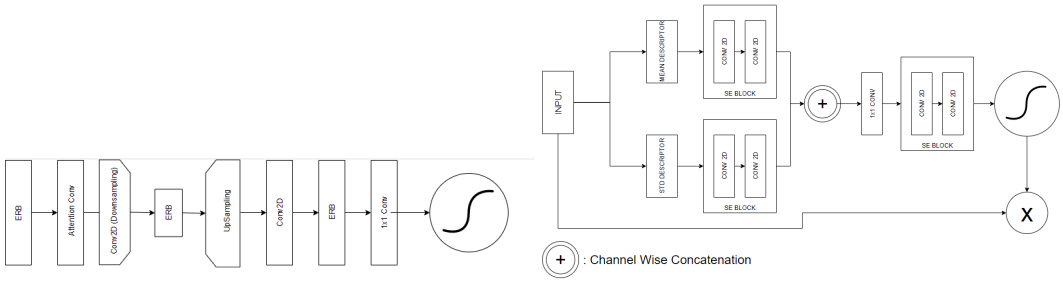


Figure 2: Efficient Residual Attention Block - (E-RAB).



(a) Efficient Mask Branch with Sigmoid Gating Function

(b) Adaptively Scaled Channel Attention Layer - (ASCA Layer). SE block refers to squeeze-and-excitation block which consisting of 2 1x1 2D convolutions where first one is used for squeeze operation and second one is used for excitation operation expanding channel dimension, mean descriptor and standard deviation descriptors are calculated for each channel.

Figure 3: Attention Mechansims

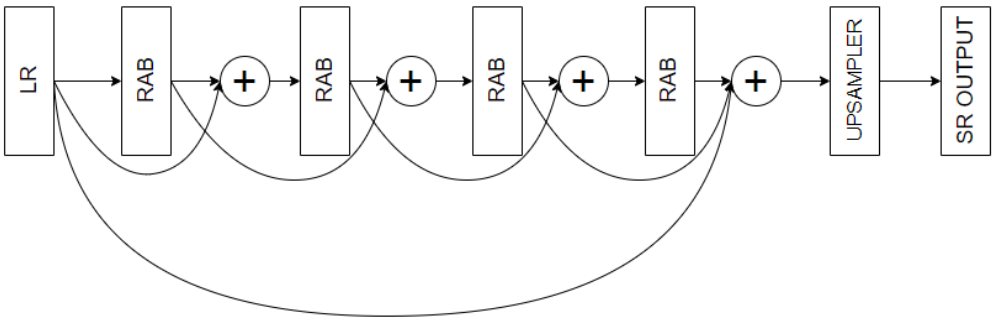


Figure 4: Final Architecture with local and global skip connections.

resolution task is based on image to image translation and learning residuals between low and high resolution image is easier in this case. As can be seen in Figure 2 our architecture's heart is the Efficient Residual Attention Block (RAB). We define our model's RAB module in three parts which is the head, body and the tail of the model. At the head of the module we have a convolutional layer which works as a low-level feature extractor. After feature extractor, in the body of our module first starts with novel Adaptively Scaled Channel Attention Layer. Two different channel-wise descriptor are obtained respectively mean descriptor and standard deviation descriptor. After that, information goes to the Squeeze and Excitation Blocks to get refined descriptor by incorporating global embedding with squeeze operation and re-calibrating with excitation operation. The information that goes out from the SE Blocks subjected to channel wise concatenation process as fusion operation. After that, we use 1x1 convolution layer to create bottleneck and interaction between mean and std descriptors. We use one more SE Block to create mask and use sigmoid function to finish the process.

$$m(z) = \sigma(W_F ReLU(W_P z)) \tag{2}$$

where z is fused descriptor from mean and std of channels, m is mask, W_P and W_F are convolution parameters, σ is sigmoid gating function that controls which and how much information should pass. We apply one Residual block to regulate the masking and try to prevent the information loss causing the masking. After regularization we create two branches, one for Trunk Branch and the other one is Mask Branch. Trunk Branch has two basic residual blocks which anything is fancy about it. In [16] the authors find that simplified RB not only contributes to image super-resolution, but also helps to construct very deep network for other image restoration tasks.

$$att_{pre}(x) = F_{PA}(F_{CA}(x)) \tag{3}$$

$$F_{RAB}(x) = F_{trunk}(att_{pre}(x)) F_{mask}(att_{pre}(x)) + x \tag{4}$$

We omit efficient residual blocks in Equation 3 and Equation 4 for simplicity which describes residual attention block basically. The main point of mask branch is how to get information on larger scope and larger receptive field size by stacked and feature selection at multiple stages in hierarchical order. Because of the multiplication between trunk and mask branch, derivative of this operation prevents trunk branch to be updated with wrong gradients because masking becomes update filter during backpropagation. We don't use max pooling in our mask branch as similar in [16] because of the information and detail loss aggressively that max pooling causes. Mask branch has basic U shape architecture as in 3a. It starts with local attention convolution as Equation 5 and downsampling convolution layer. Relative attention is formulated as

$$y_{ij} = \sum_{a,b \in N_k(i,j)} softmax_{ab}(q_{ij}^T k_{ab} + q_{ij}^T r_{a-i,b-j}) v_{ab} \tag{5}$$

where $q_i j$ is queries, $k_i j$ are keys and $v_i j$ are values such that all of them is obtained from 1x1 convolutions which are learned transformations. $N_k(i, j)$ represents neighboring pixels which is extent centered at position i, j. $r_i j$ are relative position embeddings which scales attention relative to center. It is similar to the many other self-attention and non-local attention mechanism in many ways. However, position embedding gives another expression power on the feature maps and also it provides grouping partitions on pixel features depth-wise. Repetition of this attention layer enlarges receptive field and provides multi-representation of features. After downsampling, we apply Efficient Residual Block which we will be explaining later. Then, we apply upsampling with pixel shuffling instead of transpose convolution. At the end of the mask branch, we apply 1x1 convolution to power up representation power and reclaim feature maps that are lost at pixel shuffle layer that uses them for shuffling and sigmoid function to create mask deciding which information will pass from the mask and which won't. We multiply the information that goes out of these two branches as Equation 4 as masking operation. Then, we have two Residual block to pass information and create output. If we talk about the Efficient Residual Block in Figure 1a it has some differences from the usual residual blocks as can be seen in Figure 1b. As in [2], we use two 3x3 group convolution and one 1x1 convolution to regain same feature map number in residual block. This causes to reduce the computation depending on group size. As can be seen in Figure 4 we have four Residual Attention Block (RAB) and one Upsampler, which is sub-pixel layer in this case however we might change it to meta-upscale module[20] having flexibility on scale factor, which all have local skip connections at each layer and one global skip connection.

We use multi-loss function as described below;

$$Loss(p) = \alpha * L_1(p) + \beta * L^{SSIM}(p) + \theta * L^{Perceptual}(p) \tag{6}$$

where L^{SSIM} is SSIM loss described in 7, L^{SSIM} is perceptual loss [18] using pretrained VGG19 embeddings.

$$L^{SSIM}(p) = \frac{1}{N} \sum_{p \in P} 1 - SSIM(p) \tag{7}$$

We did not use perceptual loss alone, because it tends to trick network in the embedding space ignoring other important low-level details. Structural Similarity Index(SSIM) is used as loss function directly because it measures structural similarity such as shapes and gradients implicitly. Total variation regularization[28] and texture loss [17] are taken into account; Even texture loss creates realistic textures, determining patch size equal to texture is empirical and we did not make it really work to worth. Total variation is ineffective in our evaluation. Because, our results suffer from interpolated/smoothed regions rather than noise.

4 Experimental Results

All models are trained with DIV2K[25] dataset which has 800 training pictures and 100 validation/test pictures. Set5 [21], Set14 [22], BSD100[24] and Urban100[23] are used for evaluation. We use SSIM and PSNR as evaluation metrics as in general conventions. We try to rebuild RNAN with hyper-parameters as described in its paper. α, β and θ are set to 1.5, 0.6 and 0.4 respectively. We trained our models about 500 epochs which still is not the match for original RNAN in terms of training, original RNAN was trained 3 days with NVDIA Titan Xp GPU as authors stated [1].

METHOD	SCALE	Set5		Set14		B100		URBAN100		DIV2K	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
RNAN(paper)	x2	38.17	0.9611	33.87	0.9207	32.32	0.9014	32.73	0.9340	-	-
RNAN(paper)	x4	32.49	0.8982	28.83	0.7878	27.72	0.7421	26.61	0.8023	-	-
RNAN	x2	37.83	0.948	33.43	0.911	32.03	0.899	31.22	0.913	36.46	0.944
RNAN	x3	33.97	0.904	30.04	0.889	28.94	0.786	27.68	0.828	33.74	0.931
RNAN	x4	32.04	0.854	28.34	0.781	27.38	0.744	25.63	0.739	32.54	0.931
EARN	x2	37.96	0.948	33.54	0.892	32.01	0.889	31.76	0.918	34.11	0.943
EARN	x3	31.88	0.876	30.99	0.868	29.77	0.802	28.83	0.792	32.93	0.912
EARN	x4	31.82	0.851	28.35	0.751	27.63	0.721	25.45	0.741	34.31	0.941

Table 1: Paper represents original RNAN results, and other are trained by us. Last column represents validation scores

We apply data augmentation with horizontal/vertical flip and 90 degree rotations. Adam is used for optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, epsilon=1e-8 and learning rate is set to 1e-4 and halved every 200 epochs.Group number is (number of input channel)/4 for group convolution, however performance slightly dropped when using depth-wise convolution. Number of input channels of blocks and layers is 64. Batch size is 32 and predicted super resolution, input patch size is 48 such that we use 16 input patches for one image as in RNAN, bicubic degradation is used for reconstruction model. Four residual attention blocks are used for this evaluation.

Our network outperforms some scales and dataset RNAN as seen in Table 1 even with less epoch and less resource.. Our efficiency comes from accuracy with smaller architecture in terms of layers and parameters and also evaluation time. average feed forward of EARN is 0.76 seconds per sample on our GPU, RNAN with 10 blocks takes 6.5 seconds and RNAN with 2 blocks takes 2.6 seconds as stated in its paper[16]. However, some important details are missing our reconstructed images which is still disappointing, indicated by lower SSIM score than original RNAN.

Therefore, in Figure 6 we can see that network performs really well and reconstructs easily with x2 scale task which gives strong prior to the network. Therefore our results are better than RNAN which is trained by us, RNAN produces slightly more smoothed version. However, both are so close to ground truth perceptually. However, for the case of x4 scale in Figure 5, first sample is not fully constructed by EARN and RNAN, there is bias on one diagonal against other. Feature extractor filters

[1] `https://github.com/yulunzhang/RNAN/issues/8`

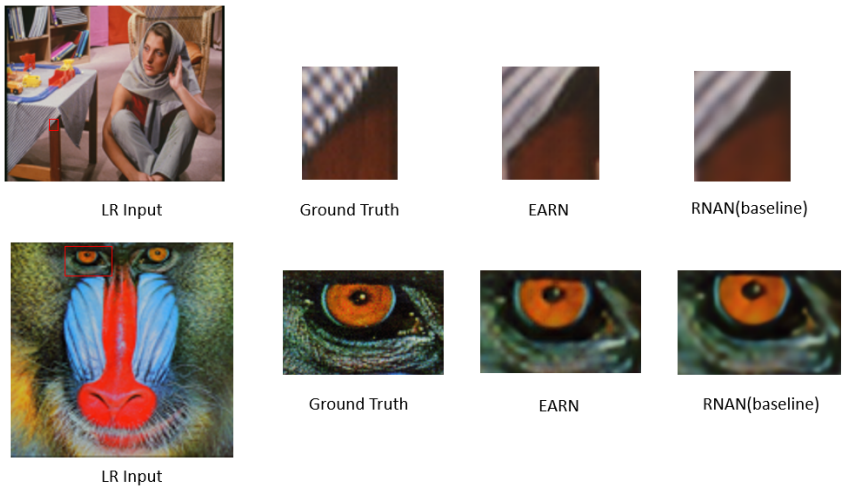


Figure 5: Comparison on Scale 4 results, RNAN baseline is trained by us.

Figure 6: Comparison on Scale 2 results, RNAN baseline is trained by us.

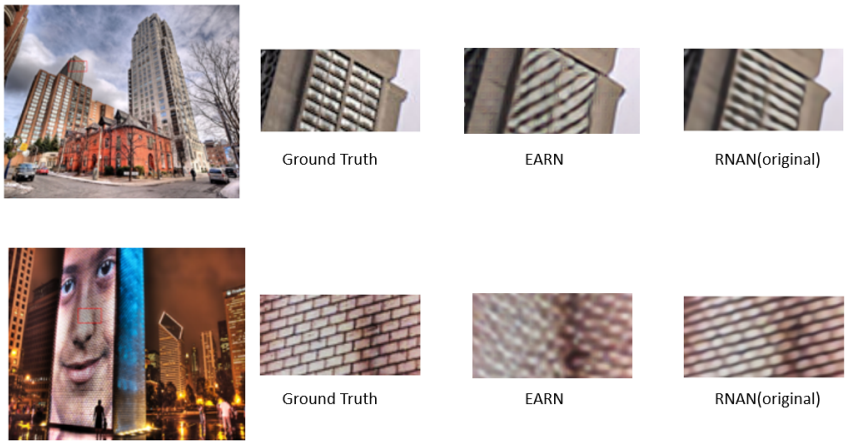


Figure 7: Comparison on Scale 4 results with original RNAN.

should contribute same amount of weights for those diagonals. Second sample in the Figure 5 is not still fully constructed, however our result is litte more detailed than RNAN.

You can see failure cases while comparing with original RNAN samples taken from its respective study [16] in Figure 7. As seen in the examples, we still can not figure out to solve lowest details compared to original RNAN.

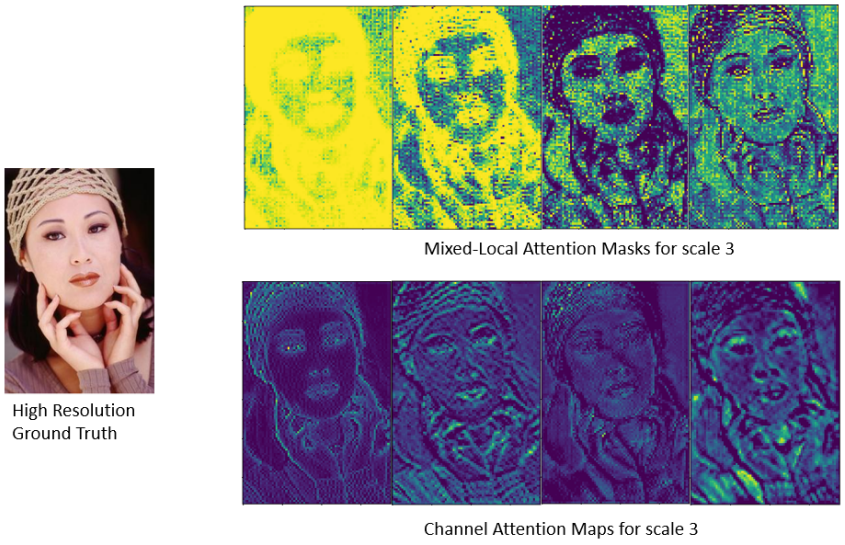


Figure 8: Different Attention Maps

In Figure 8, Mixel-Local attention map gives general attention around eyes mouth and nose, and it gives more attention into more details such as eyebrow. Therefore, It does not attend to more uniform regions. Actually, last type attention masks are the ones we want to maximize across the layers to achieve better results.

Ablation studies were done for three cases that are controlled experiments. First one is to remove ASCA layer, second one is to remove self local attention layer and last one is to remove global and local skip connections. We would also want to evaluate different group numbers for group convolution, however it would require wide parameter search.

METHOD	**Set5**		**Set14**		**B100**		**URBAN100**		**DIV2K**	
	PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
EARN	31.82	0.851	28.35	0.751	27.63	0.721	25.45	0.741	34.31	0.941
EARN w/o ASCA	31.69	0.839	28.17	0.738	27.41	0.703	25.28	0.731	34.07	0.925
EARN w/o SC	31.58	0.831	28.03	0.727	27.21	0.699	25.21	0.726	33.98	0.911
EARN w/o LA	31.72	0.848	28.33	0.748	27.59	0.718	25.32	0.739	34.26	0.936

Table 2: We obtain results by removing ASCA layer, Local Attention(LA) layer, skip connections(SC) from our original network for scale 4.

As seen in Table 2, removing local attention layer hurts performance not that much. Actually, this block is used to replace vanilla convolution layers. Thus, changing all attention layer to this self-local attention layer may make difference. However, ASCA layer and skip connections are key components of the network that are highly anticipated. Thus, lack of those components gives poor performance compared to final architecture.

5 Conclusion

We try to super resolve single images with deep learning and get visually pleasant results. Our results are promising, however we fail to outperform our original baseline mostly because of lack of resource. We see that EARN without local attention block perform similar to original EARN. As we mention before, this hybrid structure might be the reason of lower results. That means we have other options such that using residual blocks without local attention block and populating this block inside of network having non-local attention block at the start of the network which shares skip connections with those blocks as in SAN [14]. That way, we may get more details in our results. The most important part is to have good detailed structural image outputs which is measured by SSIM in this case. We believe that more efficient novel loss functions or more well-engineered multi-loss functions lead better detailed images. Besides, these kind of tasks such as image de-noising, de-hazing might share same architecture or, broadly, same framework. For instance, feeding noisy image into network that outputs noise-free super resolution output.

References

[1] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. 2019. Lightweight Image Super-Resolution with Information Multi-distillation Network. In Proceedings of the 27th ACM International Conference on Multimedia (MM '19).

[2] N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in ECCV, 2018.

[3] Qin, X., Wang, Z., Bai, Y., Xie, X., Jia, H. (2019). FFA-Net: Feature Fusion Attention Network for Single Image Dehazing. ArXiv, abs/1911.07559.

[4] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang. Residual attention network for image classification. In Computer Vision and Pattern Recognition, 2017.

[5] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[6] Ramachandran, Prajit & Parmar, Niki & Vaswani, Ashish & Bello, Irwan & Levskaya, Anselm & Shlens, Jonathon. (2019). Stand-Alone Self-Attention in Vision Models.

[7] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, "Non-local neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[8] Chao Dong, Chen Change Loy, Kaiming He, Xiaoou Tang. Image Super-Resolution Using Deep Convolutional Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Preprint, 2015

[9] Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network Wenzhe Shi, Jose Caballero; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1874-1883.

[10] C. Ledig et al., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 105-114.

[11] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Loy, C.C., Qiao, Y., & Tang, X. (2018). ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. ArXiv, abs/1809.00219.

[12] Y. Zhang, Y. Tian, Y. Kong, B. Zhong and Y. Fu, "Residual Dense Network for Image Super-Resolution," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 2472-2481.

[13] B. Lim, S. Son, H. Kim, S. Nah and K. M. Lee, "Enhanced Deep Residual Networks for Single Image Super-Resolution," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, 2017, pp. 1132-1140.

[14] T. Dai, J. Cai, Y. Zhang, S. Xia and L. Zhang, "Second-Order Attention Network for Single Image Super-Resolution," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 11057-11066.

[15] T. Vu, C. Van Nguyen, T. X. Pham, T. M. Luu, and C. D. Yoo, "Fast and efficient image quality enhancement via desubpixel convolutional neural networks," in ECCV Workshop, 2018.

[16] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu. Residual non-local attention networks for image restoration. arXiv preprint arXiv:1903.10082, 2019

[17] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in NIPS, 2015.

[18] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for realtime style transfer and super-resolution," in ECCV, 2016

[19] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in NIPS, 2016.

[20] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun, "Meta-sr: A magnification-arbitrary network for super-resolution," in CVPR, 2019.

[21] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in BMVC, 2012.

[22] R. Zeyde, M. Elad, and M. Protter, "On single image scaleup using sparse-representations," in International Conference on Curves and Surfaces, 2010.

[23] J.-B. Huang, A. Singh, and N. Ahuja, "Single image superresolution from transformed self-exemplars," in CVPR, 2015.

[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in ICCV, 2001.

[25] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in CVPRW, 2017.

[26] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in ECCV, 2018.

[27] H. Zhao, O. Gallo, I. Frosio and J. Kautz, "Loss Functions for Image Restoration With Neural Networks," in IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 47-57, March 2017, doi: 10.1109/TCI.2016.2644865.

[28] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," Physica D: Nonlinear Phenomena, vol. 60, 1992.