RBSRICNN: Raw Burst Super-Resolution through Iterative Convolutional Neural Network

Rao Muhammad Umer
Department of Computer Science
University of Udine
engr.raoumer943@gmail.com

Christian Micheloni
Department of Computer Science
University of Udine
christian.micheloni@uniud.it

Abstract

Modern digital cameras and smartphones mostly rely on image signal processing (ISP) pipelines to produce realistic colored RGB images. However, compared to DSLR cameras, low-quality images are usually obtained in many portable mobile devices with compact camera sensors due to their physical limitations. The low-quality images have multiple degradations *i.e.*, sub-pixel shift due to camera motion, mosaick patterns due to camera color filter array, low-resolution due to smaller camera sensors, and the rest information are corrupted by the noise. Such degradations limit the performance of current Single Image Super-resolution (SISR) methods in recovering high-resolution (HR) image details from a single low-resolution (LR) image. In this work, we propose a Raw Burst Super-Resolution Iterative Convolutional Neural Network (RBSRICNN[1]) that follows the burst photography pipeline as a whole by a forward (physical) model. The proposed Burst SR scheme solves the problem with classical image regularization, convex optimization, and deep learning techniques, compared to existing black-box data-driven methods. The proposed network produces the final output by an iterative refinement of the intermediate SR estimates. We demonstrate the effectiveness of our proposed approach in quantitative and qualitative experiments that generalize robustly to real LR burst inputs with onl synthetic burst data available for training.

1 Introduction

The Burst Super-resolution is the task of fusing several low-resolution (LR) frames to produce a single high-resolution (HR) image. It is a fundamental low-level vision and image processing problem with various practical applications in satellite imaging, deforestation, environment and climate change monitoring, mobile photography, video enhancement, and security and surveillance imaging as well. In the last decade, most of the photos have been taken using built-in smartphone cameras, where the resulting low-quality images are inevitable and undesirable due to their physical limitations. Since the mobile cameras are small and versatile due to their compact camera sensors, there are several key limitations [6] of the mobile phone cameras as compared to a DSLRs *i.e.*small sensor size, limited aperture, noise (*i.e.*photon shot and read noise) and limited dynamic range, limited depth of field due to fixed aperture, limited zoom and color sub-sampling. As a result, the image quality is not comparable with that of DSLR cameras. Therefore, the focus is shifted towards software solutions to mitigate cameras hardware limitations.

Mathematically, the Burst SR problem is described as the following forward image observation model for degradation process:

$$\mathbf{y}_i = \mathbf{MHS}_i(\tilde{\mathbf{x}}) + \eta_i, \quad i = 1, \ldots, B \tag{1}$$

[1]Our code and trained models are publicly available at https://github.com/RaoUmer/RBSRICNN

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

where, $\mathbf{y}_i$ is the *i-th* observed image of the LR burst B images, $\mathbf{M}$ is a *mosaicking operator* that corresponds to the CFA (Color Filter Array) of a camera (usually Bayer), $\mathbf{H}$ is a *down-sampling operator* (*i.e.*bilinear, bicubic, etc.) that resizes an HR image $\tilde{\mathbf{x}}$ by a scaling factor r, $\mathbf{S}_i$ is an *affine transformation* of the coordinate system of the image $\tilde{\mathbf{x}}$ (*i.e.*translation and rotation), and η_i is an additive *heteroskedastic Gaussian noise* related to photon shot and read noise. Recovery of the latent HR image $\tilde{\mathbf{x}}$ from the LR burst belongs to the broad spectrum of inverse problems. The operators $\mathbf{M}$ and $\mathbf{H}$ are typically ill-conditioned, *i.e.*singular. Further, this coupled with the presence of the noise η_i and the affine transformation $\mathbf{S}_i$ perturbing the measurements leads to an highly ill-posed nature of an inverse problem, where a unique solution does not exist. Burst photography pipeline with multi-frame super-resolution (MFSR) is the most common way to deal with such a scenario that generates the HR image from a low-quality burst of raw sensor images.

Numerous works have been addressed towards the task of SISR [16, 25, 36, 37, 35, 24, 38, 31, 29, 27, 40, 23, 13, 9] and real-world SISR [20, 33, 26, 11, 28, 30, 32]. Due to the ill-posed nature of the SISR problem, the existing methods have limited performance to recover high frequency details through single image learned priors. On the other hand, the MFSR aims to recover the latent HR image using multiple LR frames by exploiting the additional signal information due to sub-pixel shifts, compared to the SISR methods. The recent works [12, 18, 19, 34, 7, 3] have demonstrated the potential of MFSR methods that aim to fuse multiple LR frames to reconstruct a HR output. However, the deep learning based Raw Burst SR methods are black-box data-driven approaches with larger model size due to not directly model the image formation process, while our proposed scheme (RBSRICNN) has the merit of interpretability and small model size with good reconstruction results by following the image observation model as shown in Fig. 1. Our proposed scheme exploits a powerful image regularization and large-scale optimization techniques by training a deep CNNs in an iterative manner to produce the final SR output with an iterative refinement of the intermediate SR estimates.

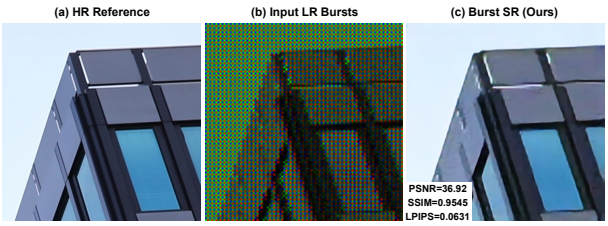

Figure 1: An image from the Zurich RAW to RGB Dataset [15] (testset), where we present (a) the ground truth HR reference image of size $384 \times 384 \times 3$, (b) the input LR bursts of size $(W \times H \times C \times B)$ $48 \times 48 \times 4 \times 14$, and (c) the Burst SR output of size $384 \times 384 \times 3$ of our network (RBSRICNN). All images are converted from raw sensor space to sRGB for visualization purpose.

2 Proposed Methodology

2.1 Problem Formulation

By referencing to the image observation model (1), the recovery of $\mathbf{x}$ from $\mathbf{y}_i$ mostly relies on the variational approach by combining the observations and prior knowledge, and the solution is obtained by minimizing the following objective function:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2 B} \sum_{i=1}^{B} \|\mathbf{y}_i - \mathbf{MHS}_i(\mathbf{x})\|_2^2 + \lambda \mathcal{R}(\mathbf{x}), \tag{2}$$

where the first term corresponds to the data fidelity term that measures the closeness of the solution to the observations, while the second term (*i.e.*$\mathcal{R}(\mathbf{x})$) corresponds to the regularization term that encodes any priors knowledge about the GT image, and λ is the trade-off parameter that governs the compromise between the data fidelity and the regularizer term. The Eq. (2) can be also written in the following form:

$$\mathbf{J}(\mathbf{x}) = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2 B} \|\mathbf{y} - \mathbf{Ax}\|_2^2 + \lambda \mathcal{R}(\mathbf{x}), \tag{3}$$

where the $\mathbf{A}$=$\mathbf{MHS}$ corresponds to the camera response. In the next section 2.2, we employ a proper optimization strategy to find the solution that minimizes the objective function (3) to get the required latent HR image.

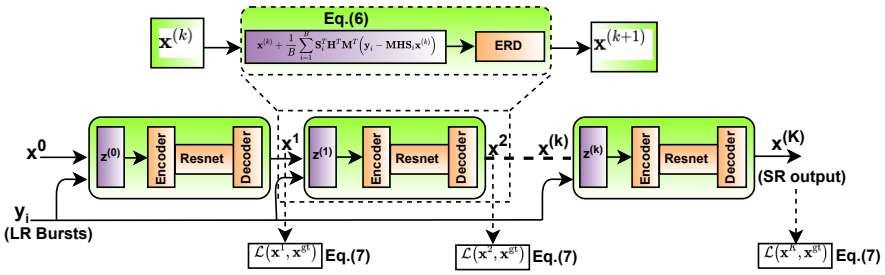

Figure 2: Visualizes the structure of the our proposed iterative Raw Burst SR scheme. Given an LR burst images ($\mathbf{y}_i$), each network's stage produces a new estimate $\mathbf{x}^{(k+1)}$ from the previous step estimate $\mathbf{x}^{(k)}$. A single optimizer is used for all network stages with shared structures and parameters.

2.2 Optimization Scheme

In the Majorization-Minimization (MM) [14, 10, 21] framework, an iterative algorithm for solving the minimization problem $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \mathbf{J}(\mathbf{x})$ takes the form $\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}} \mathbf{Q}(\mathbf{x}; \mathbf{x}^{(k)})$, where $\mathbf{Q}(\mathbf{x}; \mathbf{x}^{(k)})$ is the majorizer of the function $\mathbf{J}(\mathbf{x})$ at a fixed point $\mathbf{x}^{(k)}$ by satisfying the following two conditions: $\mathbf{Q}(\mathbf{x}; \mathbf{x}^{(k)}) > \mathbf{J}(\mathbf{x})$, $\forall \mathbf{x} \neq \mathbf{x}^{(k)}$ and $\mathbf{Q}(\mathbf{x}^{(k)}; \mathbf{x}^{(k)}) = \mathbf{J}(\mathbf{x}^{(k)})$. We need to upper-bound the $\mathbf{J}(\mathbf{x})$ by a suitable majorizer $\mathbf{Q}(\mathbf{x}; \mathbf{x}^{(k)})$ in order to get the required solution. Instead of minimizing the actual objective function (3) due to its complexity, we minimize the majorizer function $\mathbf{Q}(.)$ to produce the next solution estimate $\mathbf{x}^{(k+1)}$. By satisfying the above two properties of the majorizer function, iteratively minimizing $\mathbf{Q}(.; \mathbf{x}^{(k)})$ also decreases the actual objective function $\mathbf{J}(.)$ [14]. We write a quadratic majorizer for the complete objective function (3) as the following form:

$$\tilde{d}(\mathbf{x}; \mathbf{x}^{(k)}) = \frac{1}{2\sigma^2 B} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + g(\mathbf{x}, \mathbf{x}^{(k)}), \tag{4}$$

To start an estimate $\mathbf{x}^{(k)}$, we have:

$$g(\mathbf{x}, \mathbf{x}^{(k)}) = \frac{1}{2\sigma^2 B} \left(\mathbf{x} - \mathbf{x}^{(k)}\right)^T \left[\alpha \mathbf{I} - \mathbf{A}^T \mathbf{A}\right] \left(\mathbf{x} - \mathbf{x}^{(k)}\right), \tag{5}$$

where $\tilde{d}(\cdot, \cdot)$ is a distance function between $\mathbf{x}$ and $\mathbf{x}^{(k)}$. To get a valid majorizer, we need to satisfy the above two conditions such as $g(\mathbf{x}, \mathbf{y}) > 0, \forall \mathbf{x} \neq \mathbf{y}$ and $g(\mathbf{x}, \mathbf{x}) = 0$. This suggests that $\alpha \mathbf{I} - \mathbf{A}^T \mathbf{A}$ must be a positive definite matrix, which only holds if $\alpha > \|\mathbf{A}^T \mathbf{A}\|_2 \Rightarrow \alpha \geq B$. Finally, we proceed with the following formulation to iteratively minimize the quadratic majorizer function $\mathbf{Q}(.)$ as:

$$\hat{\mathbf{x}}^{(k)} = \arg\min_{\mathbf{x}} \mathbf{Q}(\mathbf{x}; \mathbf{x}^{(k)}) = \tilde{d}\left(\mathbf{x}; \mathbf{x}^{(k)}\right) + \lambda \mathcal{R}(\mathbf{x}) = \frac{\alpha}{2\sigma^2 B} \|\mathbf{x} - \mathbf{z}^k\|_2^2 + \lambda \mathcal{R}(\mathbf{x}) + const. = \mathrm{Prox}_{(\lambda/\alpha\sigma^2)\mathcal{R}(.)}(\mathbf{z}^k), \tag{6}$$

where $\mathbf{z}^k = \mathbf{x}^k + \mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}^k) \Rightarrow \mathbf{z}^k = \mathbf{x}^{(k)} + \frac{1}{B} \sum_{i=1}^{B} \mathbf{S}_i^T \mathbf{H}^T \mathbf{M}^T \left(\mathbf{y}_i - \mathbf{M}\mathbf{H}\mathbf{S}_i \mathbf{x}^{(k)}\right)$ (see Fig. 2), and the *const.* does not depend on $\mathbf{x}$ and thus it is irrelevant to the optimization task. The $\mathrm{Prox}_{(.)}$ is the proximal operator and it is computed as in [29]. Since the Eq. (6) is treated as the objective function of a denoising problem where $\mathbf{z}$ is the noisy observation, we employ a deep denoising neural network to get the required solution estimate $\hat{\mathbf{x}}^{(k)}$ by unrolling the network into K finite stages. Moreover, in the Eq. (6), we decouple the degradation operator $\mathbf{A}$ from the latent image $\mathbf{x}$ and now we need to tackle it with a less complex problem *i.e.*denoising. For the fast convergence and less computational cost, we adopt a strategy similar to [18, 29, 19], where the trainable extrapolation weights $\mathbf{w}^{(\mathbf{k})}$ are learnt directly from the training data instead of the fixed ones [22].

2.3 Network Architecture and Training

The proposed Raw Burst SR scheme is shown in Fig. 2. We unroll the proposed RBSRICNN into K stages, where each stage computes the refined estimate of the solution. $\mathbf{y}_i$ is an input Raw LR burst, $\mathbf{x}^0$ is an initial estimate, and $\mathbf{x}^K$ is the final estimated SR image. We adapt the similar Encoder-Resnet-Decoder architecture as done in [29]. Due to the iterative nature of our Burst SR approach, the network parameters are updated by using Truncated Backpropagation Through Time (TBPTT) algorithm as done in [18, 29] to train our network, where the sequence is unrolled into a small number of k-steps out of total K and then the back-propagation is performed on the small

Table 1: **Comparison with others.** We compare our method with the common evaluation metrics (PSNR/SSIM/LPIPS). The quantitative SR results ($\times 4$ upscale) are shown over the synthetic and real Burst SR test sets. The arrows indicate if high $\uparrow$ or low $\downarrow$ values are desired.

Burst SR Method	#Params [M]	#Conv2d	Synthetic data			Real data			Fine-tuned on Real data
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	
DeepJoint [12]+RRDB [33]	17.26	371	33.25	0.881	0.195	42.13	0.957	0.088	✓
DeepBurstSR [3]	5.25	48	34.48	**0.905**	**0.118**	**45.17**	**0.978**	**0.037**	✓
HighRes-net [7]	1.11	25	34.30	0.891	0.170	43.99	0.972	0.051	✓
RBSRICNN (ours)	**0.38**	**12**	**37.62**	0.895	0.166	41.40	0.952	0.101	✗

k-steps. During the training, we use the following function to minimize the ℓ_1-Loss between the estimated latent SR image ($\mathbf{x}^{(k)}$) and ground-truth (GT) ($\mathbf{x}^{(gt)}$) after k-steps as:

$$\mathcal{L} = \arg\min_{\Theta} \mathcal{L}(\mathbf{\Theta}) = \frac{1}{2} \sum_{i=1}^{N} \|\mathbf{x}_i^k - \mathbf{x}_i^{gt}\|_1 \tag{7}$$

where N is the mini-batch size and $\mathbf{\Theta}$ are the trainable parameters of our network.

3 Experiments

3.1 Training data and evaluation metrics

We used $46,839$ and 1204 sRGB images from the Zurich RAW to RGB dataset [15] for the training and the validation, respectively. We generate the synthetic RAW LR bursts with their corresponding HR using the data generation code in [3], where the sRGB image is first converted to the *Raw (linear) sensor* space using an inverse camera pipeline [4], then the LR burst is generated by applying random translations and rotations, followed by bilinear downsampling, further mosaicked and corrupted by random noise. For the evaluation of our method on the real dataset, we used the BurstSR testset containing 639 real-world LR bursts, provided in the Burst SR challenge [2]. In the real LR bursts, each burst sequence contains 14 RAW images captured using a handheld smartphone camera using identical camera settings (*e.g.*, exposure, ISO) resulting in a small random offset between the images within the burst. We evaluated the trained model under PSNR, SSIM, and LPIPS [39] metrics. The quantitative Burst SR results are evaluated on the raw *linear sensor* space.

3.2 Training description

We estimate the warp matrix (*i.e.*$\mathbf{S}_i$, refers to the section-2) to align the bursts to the base/reference frame using the Enhanced Correlation Coefficient (ECC) [8] method as do in [19] for all experiments. For those bursts whose are not aligned by the ECC method, we keep them in the training and testing data by making the assumption of the identity matrix. For the training phase, we set the input Raw burst LR patch sizes as $48 \times 48 \times 4$ with their corresponding Raw HR patch sizes as $384 \times 384 \times 3$ by the scaling factor $\times 4$. We use the LR burst size of 14. We train the network for 368k iterations with a batch size of 2 using Adam optimizer [17] with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ without weight decay to minimize the loss (7). The learning rate is set to 10^{-3} for all iterations. We unroll the proposed network into K stages, where we set K as 10. We implemented our method with Pytorch 1.7.1. The experiments are performed under Windows 10 with i7-8700 CPU with 32GB RAM and on NVIDIA GeForce RTX-3090 GPU with 24GB memory. The average running times (image per second on GPU) are 0.3350 and 0.8838 over the synthetic and real testsets, respectively.

3.3 Comparison with the Burst SR methods

We compared our method with existing Burst SR methods including DeepJoint [12]+RRDB [33], DeepBurstSR [3], and HighRes-net [7]. Table 1 compares the quantitative SR results of our method with others over the synthetic and real Burst SR testsets. We have achieved good performance on the synthetic LR burst testset, while, the parameters and depth of our proposed network is much less than the other methods. On the real-world LR burst testset, the DeepBurstSR outperforms the others methods in terms of PSNR/SSIM/LPIPS, while we have a comparable performance, even though our method is not fine-tuned on the real training data with comparable visual results as shown in Fig. 3. The fine-tuning [3] increases the performance, but it also further requires significant additional

Figure 3: $\times 4$ SR visual comparison of the proposed RBSRICNN method with the existing Burst SR methods on the real-world BurstSR [3] testset. All images are converted from the raw sensor space to sRGB for visualization purpose.

labelled training data that is difficult to collect in practice, and it is also more computational expensive in terms of training time and hardware resources. We also participated in the NTIRE2021 Burst SR Challenges [2] and our method (**MLP_BSR**) is ranked among other participants as well.

3.4 Ablation Study

For our ablation study, we compared the impact of different numbers of input burst frames and iterative steps for the proposed Burst SR method in Table 2. It shows the generalization ability of our method to the larger input LR bursts and iterative steps, and effective utilization of the information from the multiple LR frames in order to improve burst SR performance.

Table 2: **Ablation study.** Impact of different number of input burst frames (B) and number of iterative steps (K). The quantitative results are reported on the synthetic burst testset.

Burst Size (B)	**iterative steps** ($K = 5$)			**iterative steps** ($K = 10$)		
	PSNR↑	**SSIM**↑	**LPIPS**↓	**PSNR**↑	**SSIM**↑	**LPIPS**↓
2	34.19	0.8790	0.2498	34.12	0.8777	0.2480
4	34.69	0.8852	0.2359	34.66	0.8842	0.2317
8	35.09	0.8887	0.2277	34.99	0.8876	0.2217
14	35.12	0.8896	0.2255	35.30	0.8903	0.2165
16	**35.21**	**0.8907**	**0.2232**	35.30	**0.8909**	0.2168
32	35.23	0.8902	0.2236	**35.41**	**0.8909**	**0.2159**

3.5 Discussion and Limitation

Our proposed method has a close connection to other proximal algorithms such as ISTA [5] and FISTA [1] that require the exact form of the employed regularizer such as Total Variation / Hessian Schatten-norm [21]. However, in our case, the regularizer is learned implicitly from the training data (*i.e.*non-convex form) and therefore we do not have any assumptions regarding the explicit form of the regularizer. Our proposed algorithm acts as an inexact form of proximal gradient descent steps. Since we are estimating the warping matrix by using the ECC [8] method to align the observations to the reference frame, sometimes its estimation is imprecise that will introduce undesirable artifacts to the final SR result.

4 Conclusion

We proposed a deep iterative Burst CNNs for a multi-frame super-resolution task by following the burst photography pipeline. The proposed burst SR scheme follows the burst physical model and solves the overall problem through iterative refinement of the intermediate solution estimates to get the final SR output. The proposed network exploits powerful image regularization, large-scale optimization, and deep learning techniques for multi-frame image restoration. Our model requires much less parameters and 2d convolution operation in comparison to other competing methods. Our method achieves good burst SR results in terms of the synthetic data as well as comparable visual quality of the real-world bursts with respect to the existing approaches.

References

[1] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, pages 183–202, 2009.

[2] Goutam Bhat, Martin Danelljan, Radu Timofte, et al. NTIRE 2021 challenge on burst super-resolution: Methods and results. In *CVPRW*, 2021.

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deep burst super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9209–9218, 2021.

[4] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11036–11045, 2019.

[5] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, pages 1413–1457, 2004.

[6] Mauricio Delbracio, Damien Kelly, Michael S Brown, and Peyman Milanfar. Mobile computational photography: A tour. *arXiv preprint arXiv:2102.09000*, 2021.

[7] Michel Deudon, Alfredo Kalaitzis, Israel Goytom, Md Rifat Arefin, Zhichao Lin, Kris Sankaran, Vincent Michalski, Samira E Kahou, Julien Cornebise, and Yoshua Bengio. Highres-net: Recursive fusion for multi-frame super-resolution of satellite imagery. *arXiv preprint arXiv:2002.06460*, 2020.

[8] Georgios D Evangelidis and Emmanouil Z Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1858–1865, 2008.

[9] Yuchen Fan, Jiahui Yu, Yiqun Mei, Yulun Zhang, Yun Fu, Ding Liu, and Thomas S Huang. Neural sparse representation for image restoration. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[10] Mário AT Figueiredo, José M Bioucas-Dias, and Robert D Nowak. Majorization–Minimization algorithms for wavelet-based image restoration. *IEEE Transactions on Image processing*, pages 2980–2991, 2007.

[11] Manuel Fritsche, Shuhang Gu, and Radu Timofte. Frequency separation for real-world super-resolution. *ICCV workshops*, 2019.

[12] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG)*, pages 1–12, 2016.

[13] Yuanbiao Gou, Boyun Li, Zitao Liu, Songfan Yang, and Xi Peng. Clearer: Multi-scale neural architecture search for image restoration. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[14] David R Hunter and Kenneth Lange. A tutorial on MM algorithms. *The American Statistician*, pages 30–37, 2004.

[15] Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 536–537, 2020.

[16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. *CVPR*, pages 1646–1654, 2016.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, pages 1–15, 2015.

[18] Filippos Kokkinos and Stamatios Lefkimmiatis. Iterative joint image demosaicking and denoising using a residual denoising network. *IEEE Transactions on Image Processing*, pages 4177–4188, 2019.

[19] Filippos Kokkinos and Stamatis Lefkimmiatis. Iterative residual cnns for burst photography applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5929–5938, 2019.

[20] Christian Ledig et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 4681–4690, 2017.

[21] Stamatios Lefkimmiatis, Aurélien Bourquard, and Michael Unser. Hessian-based norm regularization for image restoration with biomedical applications. *IEEE Transactions on Image Processing*, pages 983–995, 2011.

[22] Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. *Advances in neural information processing systems (NIPS)*, pages 379–387, 2015.

[23] Wenbo Li, Kun Zhou, Lu Qi, Nianjuan Jiang, Jiangbo Lu, and Jiaya Jia. Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[24] Yaoman Li, Jinglei Yang, Zheng Liu, Xiaomin Yang, Gwanggil Jeon, and Wei Wu. Feedback network for image super-resolution. *CVPR*, 2019.

[25] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. *CVPRW*, pages 1132–1140, 2017.

[26] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Unsupervised learning for real-world super-resolution. *ICCV workshops*, 2019.

[27] Zhengxiong Luo, Yan Huang, Shang Li, Liang Wang, and Tieniu Tan. Unfolding the alternating optimization for blind super resolution. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[28] Rao Muhammad Umer, Gian Luca Foresti, and Christian Micheloni. Deep generative adversarial residual convolutional networks for real-world super-resolution. In *CVPRW*, pages 438–439, 2020.

[29] Rao Muhammad Umer, Gian Luca Foresti, and Christian Micheloni. Deep iterative residual convolutional network for single image super-resolution. In *Proceedings of the International Conference of Pattern Recognition (ICPR)*, January 2021.

[30] Rao Muhammad Umer and Christian Micheloni. Deep cyclic generative adversarial residual convolutional networks for real image super-resolution. In *ECCVW*, August 2020.

[31] Rao Muhammad Umer, Gian Luca Foresti, and Christian Micheloni. Deep super-resolution network for single image super-resolution with realistic degradations. In *ICDSC*, pages 21:1–21:7, September 2019.

[32] Rao Muhammad Umer, Asad Munir, and Christian Micheloni. A deep residual star generative adversarial network for multi-domain image super-resolution. In *6th International Conference on Smart and Sustainable Technologies (SpliTech)*, 2021.

[33] 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 *ECCV*, 2018.

[34] Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. Handheld multi-frame super-resolution. *ACM Transactions on Graphics (TOG)*, pages 1–18, 2019.

[35] Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *CVPRW*, pages 701–710, 2018.

[36] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. *CVPR*, pages 2808–2817, 2017.

[37] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. *CVPR*, pages 3262–3271, 2018.

[38] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Deep plug-and-play super-resolution for arbitrary blur kernels. In *CVPR*, pages 1671–1681, 2019.

[39] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

[40] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. Cross-scale internal graph neural network for image super-resolution. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.