This ECCV 2018 workshop paper, provided here by the Computer Vision Foundation, is the author-created version. The content of this paper is identical to the content of the officially published ECCV 2018 LNCS version of the paper as available on SpringerLink: https://link.springer.com/conference/eccv

Fast Perceptual Image Enhancement

Etienne de Stoutz[0000−0001−5439−3290], Andrey Ignatov[0000−0003−4205−8748], Nikolay Kobyshev[0000−0001−6456−4946], Radu Timofte[0000−0002−1478−0402], and Luc Van Gool[0000−0002−3445−5711]

ETH Zurich

Abstract. The vast majority of photos taken today are by mobile phones. While their quality is rapidly growing, due to physical limitations and cost constraints the mobile phones cameras struggle to compare in quality with DSLR cameras. This motivates us to computationally enhance these images. We extend upon the results of Ignatov *et al.*, where they are able to translate images from compact mobile cameras into images with comparable quality to high-resolution photos taken by DSLR cameras. However, the neural models employed require large amounts of computational resources and are not lightweight enough to run on mobile devices. We build upon the prior work and explore different network architectures targeting an increase in image quality and speed. With an efficient network architecture which does most of its processing in a lower spatial resolution, we achieve a significantly higher mean opinion score (MOS) than the baseline while speeding up the computation by 6.3× on a consumer-grade CPU. This suggests a promising direction for neural-network-based photo enhancement using the phone hardware of the future.

1 Introduction

The compact camera sensors found in low-end devices such as mobile phones have come a long way in the past few years. Given adequate lighting conditions, they are able to reproduce unprecedented levels of detail and color. Despite their ubiquity, being used for the vast majority of all photographs taken worldwide, they struggle to come close in image quality to DSLR cameras. These professional grade instruments have many advantages including better color reproduction, less noise due to larger sensor sizes, and better automatic tuning of shooting parameters.

Furthermore, many photographs were taken in the past decade using significantly inferior hardware, for example with early digital cameras or early 2010s smartphones. These do not hold up well to our contemporary tastes and are limited in artistic quality by their technical shortcomings.

The previous work by Ignatov *et al.* [8] that this paper is based upon proposes a neural-network powered solution to the aforementioned problems. They use a dataset comprised of image patches from various outdoor scenes simultaneously taken by cell phone cameras and a DSLR. They pose an image translation problem, where they feed the low-quality phone image into a residual convolutional

neural net (CNN) model that generates a target image, which, when the network is trained, is hopefully perceptually close to the high-quality DSLR target image.

In this work, we take a closer look at the problem of translating poor quality photographs from an iPhone 3GS phone into high-quality DSLR photos, since this is the most dramatic increase in quality attempted by Ignatov *et al.* [8]. The computational requirements of this baseline model, however, are quite high (20 s on a high-end CPU and 3.7 GB of RAM for a HD-resolution image). Using a modified generator architecture, we propose a way to decrease this cost while maintaining or improving the resulting image quality.

2 Related Work

A considerable body of work is dedicated to automatic photo enhancement. However, it traditionally only focused on a specific subproblem, such as super-resolution, denoising, deblurring, or colorization. All of these subproblems are tackled simultaneously when we generate plausible high-quality photos from low-end ones. Furthermore, these older works commonly train with artifacts that have been artificially applied to the target image dataset. Recreating and simulating all the flaws in one camera given a picture from another is close to impossible, therefore in order to achieve real-world photo enhancement we use the photos simultaneously captured by a capture rig from Ignatov *et al.* [8]. Despite their limitations, the related works contain many useful ideas, which we briefly review in this section.

Image super-resolution is the task of increasing the resolution of an image, which is usually trained with down-scaled versions of the target image as inputs. Many prior works have been dedicated to doing this using CNNs of progressively larger and more complex nature [4, 14, 18, 20, 22, 23]. Initially, a simple pixel-wise mean squared error (MSE) loss was often used to guarantee high fidelity of the reconstructed images, but this often led to blurry results due to uncertainty in pixel intensity space. Recent works [2] aim at perceptual quality and employ losses based on VGG layers [12], and generative adversarial networks (GANs) [5, 15], which seem to be well suited to generating plausible-looking, realistic high-frequency details.

In *image colorization*, the aim is to hallucinate color for each pixel, given only its luminosity. It is trained on images with their color artificially removed. Isola *et al.* [11] achieve state of the art performance using a GAN to solve the more general problem of image-to-image translation.

Image deblurring and dehazing aim to remove optical distortions from photos that have been taken out of focus, while the camera was moving, or of faraway geographical or astronomical features. The neural models employed are CNNs, typically trained on images with artificially added blur or haze, using a MSE loss function [19, 16, 17, 7, 3]. Recently, datasets with both hazy and haze-free images were introduced [1] and solutions such as the one of Ki *et al.* [13] were proposed, which use a GAN, in addition to L1 and perceptual losses. Similar techniques are effective for *image denoising* as well [27, 25, 24, 21].

2.1 General Purpose Image-to-Image Translation and Enhancement

The use of GANs has progressed towards the development of general purpose image-to-image translation. Isola *et al.* [11] propose a conditional GAN architecture for paired data, where the discriminator is conditioned on the input image. Zhu *et al.* [28] relax this requirement, introducing the cycle consistency loss which allows the GAN to train on unpaired data. These two approaches work on many surprising datasets, however, the image quality is too low for our purpose of photo-realistic image enhancement. This is why Ignatov *et al.* introduce paired [8] and unpaired [9] GAN architectures that are specially designed for this purpose.

2.2 Dataset

The DPED dataset [8] consists of photos taken simultaneously by three different cell phone cameras, as well as a Canon 70D DSLR camera. In addition, these photographs are aligned and cut into 100x100 pixel patches, and compared such that patches that differ too much are rejected. In this work, only the iPhone 3GS data is considered. This results in 160k pairs of images.

2.3 Baseline

As a baseline, the residual network with 4 blocks and 64 channels from Ignatov *et al.* [8] is used.

Since using a simple pixel-wise distance metric does not yield the intended perceptual quality results, the output of the network is evaluated using four carefully designed loss functions.

The generated image is compared to the target high-quality DSLR image using the color loss and the content loss. The same four losses and training setup as the baseline are also used by us in this work.

Color Loss. The color loss is computed by applying a Gaussian blur to both source and target images, followed by a MSE function. Let X and Y be the original images, then X_b and Y_b are their blurred versions, using

$$X_b(i,j) = \sum_{k,l} X(i+k, j+l) \cdot G(k,l), \tag{1}$$

where G is the 2D Gaussian blur operator

$$G(k,l) = A \exp\left(-\frac{(k-\mu_x)^2}{2\sigma_x} - \frac{(l-\mu_y)^2}{2\sigma_y} \right). \tag{2}$$

The color loss can then be written as

$$\mathcal{L}_{\text{color}}(X,Y) = \|X_b - Y_b\|_2^2 . \tag{3}$$

We use the same parameters as defined in [8], namely $A = 0.053, \mu_{x,y} = 0$, and $\sigma_{x,y} = 3$.

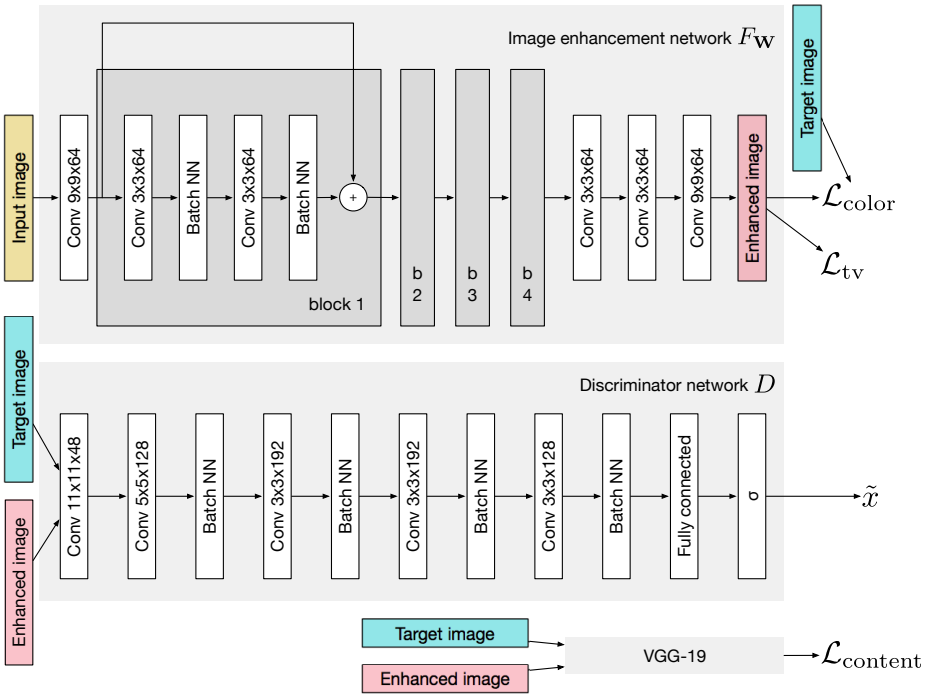

Fig. 1. The overall architecture of the DPED baseline [8]

Content Loss. The content loss is computed by comparing the two images after they have been processed by a certain number of layers of VGG-19. This is superior to a pixel-wise loss such as per-pixel MSE, because it closely resembles human perception [8, 26], abstracting away such negligible details as a small shift in pixels, for example. It is also important because it helps preserve the semantics of the image. It is defined as

$$\mathcal{L}_{\text{content}} = \frac{1}{C_j H_j W_j} \left\| \psi_j \left(F_{\text{w}} \left(I_s \right) \right) - \psi_j \left(I_t \right) \right\| \tag{4}$$

where $\psi_j(\cdot)$ is the feature map of the VGG-19 network after its j-th convolutional layer, C_j, H_j, and W_j are the number, height, and width of this map, and $F_{\mathbf{W}} \left(I_s \right)$ denotes the enhanced image.

Texture Loss. One important loss which technically makes this network a GAN is the texture loss [8]. Here, the output images are not directly compared to the targets, instead, a discriminator network is tasked with telling apart real DSLR images from fake, generated ones. During training, its weights are optimized for maximum discriminator accuracy, while the generator's weights are optimized in the opposite direction, to try to minimize the discriminator's accuracy, therefore producing convincing fake images.

Before feeding the image in, it is first converted to grayscale, as this loss is specifically targeted on texture processing. It can be written as

$$\mathcal{L}_{\text{texture}} = -\sum_i \log D(F_{\mathbf{W}}(I_s), I_t), \tag{5}$$

where $F_{\mathbf{W}}$ and D denote the generator and discriminator networks, respectively.

Total Variation Loss. A total variation loss is also included, so as to encourage the output image to be spatially smooth, and to reduce noise.

$$\mathcal{L}_{\text{tv}} = \frac{1}{CHW}\|\nabla_x F_{\mathbf{W}}(I_s) + \nabla_y F_{\mathbf{W}}(I_s)\| \tag{6}$$

Again, C, H, and W are the number of channels, height, and width of the generated image $F_{\mathbf{W}}(I_s)$. It is given a low weight overall.

Total Loss. The total loss is comprised from a weighted sum of all above mentioned losses.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{content}} + 0.4 \cdot \mathcal{L}_{\text{texture}} + 0.1 \cdot \mathcal{L}_{\text{color}} + 400 \cdot \mathcal{L}_{\text{tv}}, \tag{7}$$

Ignatov *et al.* [8] use the *relu_5_4* layer of the VGG-19 network, and mention that the above coefficients where chosen in experiments run on the DPED dataset.

3 Experiments and Results

3.1 Experiments

Adjusting Residual CNN Parameters. In order to gain an understanding of the performance properties of the DPED model [8], the baseline's residual CNN was modified in the number of filters (or channels) each layer would have, the size of each filter's kernel, and the number of residual blocks there would be in total. While reducing the number of blocks was effective and increasing the performance, and decreasing the number of features even more so, this came at a large cost in image quality. Kernel sizes of 5×5 were also attempted instead of 3×3, but did not provide the quality improvements necessary to justify their computational costs.

In Fig. 2 and Table 1, a frontier can be seen, beyond which this simple architecture tuning cannot reach. More sophisticated improvements must therefore be explored.

Parametric ReLU. Parametric ReLU [6] is an activation function defined as

$$\text{PReLU}\,(y_i) = \begin{cases} y_i, & \text{if } y_i > 0 \\ a_i y_i, & \text{if } y_i \leq 0 \end{cases} \tag{8}$$

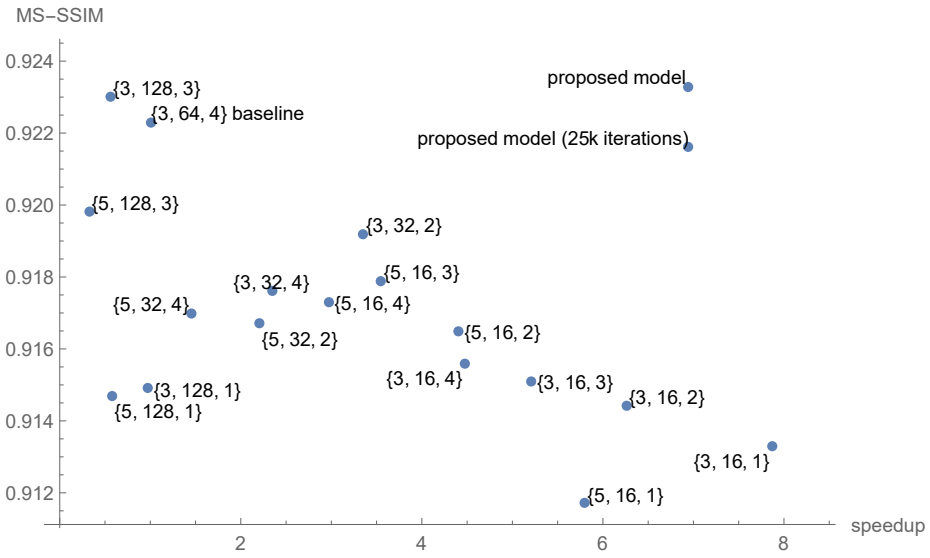

Fig. 2. Speedup (relative to the baseline) vs. MS-SSIM results on DPED test images, from adjusting residual CNN parameters. Key: {kernel size, channels, blocks}. Proposed method for reference. All models trained for 25k iterations, except for the proposed model, at 40k.

where y_i is the i-th element of the feature vector, and a_i is the i-th element of the PReLU learned parameter vector. This permits the network to learn a slope for the ReLU activation function instead of leaving it at a constant 0 for negative inputs. In theory, this would cause the network to learn faster, prevent ReLUs from going dormant, and overall provide more power for the network at a small performance cost.

In practice though (see an example in Table 2), this cost was more than what was hoped, and it did not perceptibly increase the image quality.

Strided and Transposed Convolutions. In order to more drastically reduce the computation time requirements, a change in the original architecture was implemented, where the spatial resolution of the feature maps is halved, and subsequently halved again, using strided convolutional layers. At the same time, each of these strided layers doubles the number of feature maps, as suggested by Johnson *et al.* [12].

This down-sampling operation is followed by two residual blocks at this new, $4\times$ reduced resolution, which is then followed by transposed (fractionally strided) convolution layers, which scale the feature map back up to its original resolution, using a trainable up-sampling convolution.

Table 1. Average PSNR/SSIM results on DPED test images, using the original residual CNN architecture with adjusted parameters. 25k iterations, batch size 50.

kernel size	channels	blocks	time (s)	PSNR	MS-SSIM
3	64	4	25.155	**22.6225**	0.9223
3	16	1	**6.885**	22.1479	0.9133
3	16	2	8.629	22.0441	0.9144
3	16	3	10.376	22.1148	0.9151
3	16	4	12.106	22.1362	0.9156
3	32	2	16.137	22.3807	0.9192
3	32	4	23.106	22.3300	0.9176
3	128	1	59.775	22.4285	0.9149
3	128	3	95.532	22.2768	**0.9230**
5	16	1	9.297	21.8157	0.9117
5	16	2	12.332	21.6677	0.9165
5	16	3	15.211	22.0704	0.9179
5	16	4	18.243	21.9391	0.9173
5	32	2	24.538	21.9434	0.9167
5	32	4	37.137	21.5100	0.9170
5	128	1	93.066	22.0770	0.9147
5	128	3	164.068	21.5695	0.9198

At each resolution, the previous feature maps of the same resolution are added to the new maps, through skip connections, in order to facilitate this network to learn simple, non-destructive transformations like the identity function.

This new architecture introduced slight checkerboard artifacts related to the upscaling process, but overall, it allowed for a much faster model without the loss in quality associated with the more straightforward approaches previously described. In Table 2 are summarized the quantitative results for several configurations.

3.2 Results

Table 2. Average PSNR/SSIM results on DPED test images, using the proposed strided architecture with varying parameters. The best configuration we propose, line 3, was chosen as a compromise between quality and speed.

kernel size	channels	PReLU	time (s)	PSNR	MS-SSIM
3	16-64	no	**7.729**	22.3049	0.9176
3	32-128	no	14.641	22.3909	0.9235
3-4	*16-64*	*no*	*7.987*	*22.5248*	*0.9233*
3-4	32-128	no	15.413	**22.6636**	**0.9248**
4	16-64	no	8.84	22.2421	0.9232
4	32-128	no	17.826	22.2812	0.9206
4	32-128	yes	20.584	22.3166	0.9209

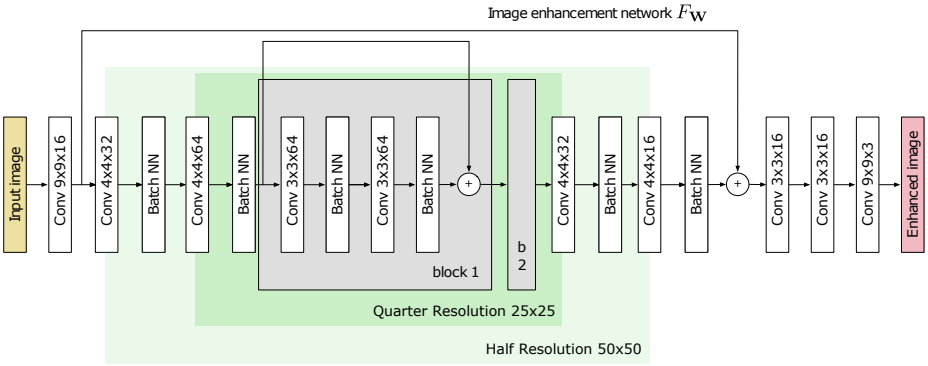

Fig. 3. The generator architecture of the proposed method. Discriminator and losses are the same as in the baseline.

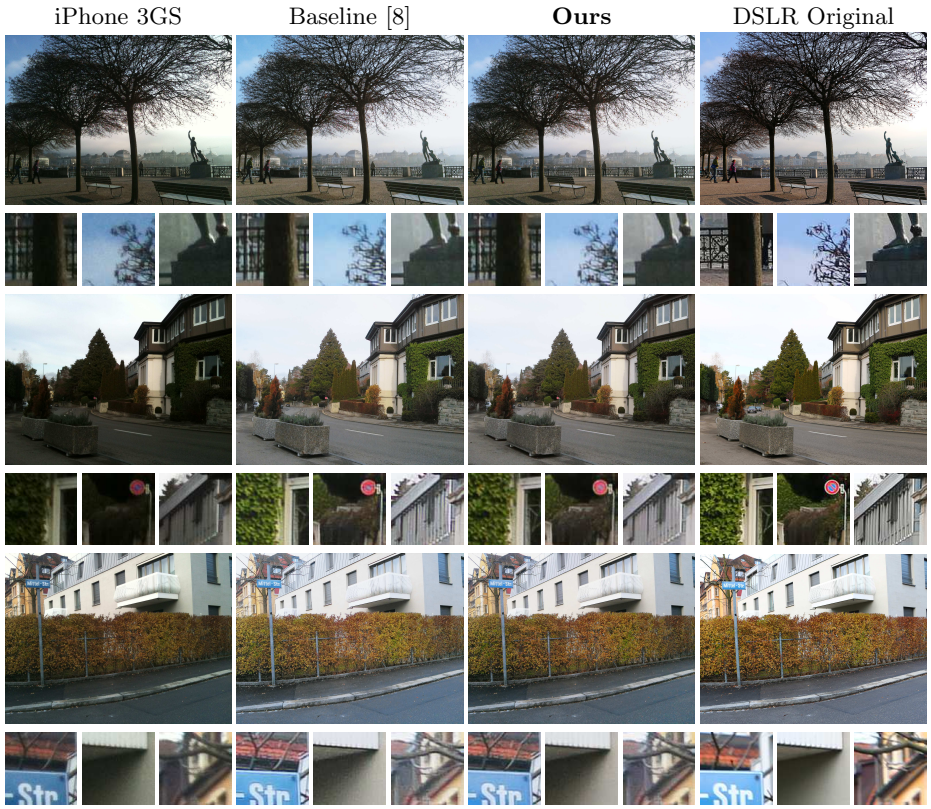

Fig. 4. Visual assessment. From left to right: The input test image from the iPhone 3GS, the output from the baseline model, the output from our model, and the (cropped) ground truth photograph from the DSLR camera.

The best result we achieved was with this new strided approach. The generator architecture is shown in Fig. 3. We chose a kernel size of 3×3, except in the strided convolutional layers, where we opted for 4×4 instead, in order to mitigate the checkerboard artifacts. The number of feature maps starts at 16 and increases up to 64 in the middle of the network. We trained the network for 40k iterations using an Adam optimizer and a batch size of 50.

Our network [1] takes only 3.2 s of CPU time to enhance a 1280×720 px image compared to the baseline's 20.5 s. This represents a 6.3-fold speedup. Additionally, the amount of RAM required is reduced from 3.7 GB to 2.3 GB.

As part of a PIRM 2018 challenge on perceptual image enhancement on smartphones [10], a user study was conducted where 2000 people were asked to rate the visual results (photos) of the solutions submitted by challenge participants. The users were able to rate each photo with scores of 1, 2, 3 and 4, corresponding to low and high-quality visual results. The average of all user ratings was then computed and considered as a MOS score of each solution.

Table 3. PIRM 2018 challenge final ranking of teams and baselines [10]

Team	PSNR	MS-SSIM	**MOS**	CPU (ms)	GPU (ms)	RAM (GB)
Mt.Phoenix	21.99	0.9125	2.6804	682	64	1.4
EdS (Ours)	21.65	0.9048	2.6523	3241	253	2.3
BOE-SBG	21.99	0.9079	2.6283	1620	111	1.6
MENet	22.22	0.9086	2.6108	1461	138	1.8
Rainbow	21.85	0.9067	2.5583	828	111	1.6
KAIST-VICLAB	21.56	0.8948	2.5123	2153	181	2.3
SNPR	22.03	0.9042	2.4650	1448	81	1.6
DPED (Baseline)	21.38	0.9034	2.4411	20462	1517	3.7
Geometry	21.79	0.9068	2.4324	833	83	1.6
IV SR+	21.60	0.8957	2.4309	1375	125	1.6
SRCNN (Baseline)	21.31	0.8929	2.2950	3274	204	2.6
TEAM ALEX	21.87	0.9036	2.1196	781	70	1.6

With a MOS of 2.6523, our submission (see Table 3) scored significantly higher than the DPED baseline (2.4411) and was second only to the winning submission, which scored 2.6804. The submission was tested against a different test set, which partially explains its lower PSNR and MS-SSIM scores. It should be noted that the submission shares the same architecture as this paper's main result, but was trained for only 33k iterations.

Differences between the DPED baseline and our result are somewhat subtle. Our model produces noticeably fewer colored artifacts around hard edges (e.g. Fig. 4, first row, first zoom box), more accurate colors (e.g. the sky in first row, second box), as well as reduced noise in smooth shadows (last row, second box), and in dense foliage (middle row, first box), it produces more realistic textures

[1] Codes and models publicly released at: http://www.vision.ee.ethz.ch/ timofter/

than the baseline. Contrast, especially in vertical features (middle row, third box), is often less pronounced. However, this comes with the advantage of fewer grid-like artifacts. For more visual results of our method we refer the reader to the Appendix.

While these subjective evaluation methods are clearly in favor of our method, the PSNR and MS-SSIM scores comparing the generated images to the target DSLR photos are less conclusive. PSNR and MS-SSIM seem to be only weakly correlated with MOS [10]. Better perceptual quality metrics including ones requiring no reference images might be a promising component of future works.

4 Conclusion

Thanks to strided convolutions, a promising architecture was found in the quest for efficient photo enhancement on mobile hardware. Our model produces clear, detailed images exceeding the quality of the baseline, while only requiring 16 % as much computation time.

Even though, as evidenced by the PIRM 2018 challenge results [10], further speed improvements will definitely be seen in future works, it is reassuring to conclude that convolutional neural network-based image enhancement can already produce high quality results with performance acceptable for mobile devices.

Acknowledgments

This work was partly supported by ETH Zurich General Fund and a hardware (GPU) grant from NVIDIA.

References

1. Ancuti, C., Ancuti, C.O., Timofte, R.: Ntire 2018 challenge on image dehazing: Methods and results. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2018)
2. Blau, Y., Mechrez, R., Timofte, R., Michaeli, T., Zelnik-Manor, L.: 2018 pirm challenge on perceptual image super-resolution. In: European Conference on Computer Vision Workshops (2018)
3. Cai, B., Xu, X., Jia, K., Qing, C., Tao, D.: Dehazenet: An end-to-end system for single image haze removal. IEEE Transactions on Image Processing **25**(11), 5187–5198 (2016)
4. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: European conference on computer vision. pp. 184–199. Springer (2014)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
6. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)

7. Hradiš, M., Kotera, J., Zemcík, P., Šroubek, F.: Convolutional neural networks for direct text deblurring. In: Proceedings of BMVC. vol. 10, p. 2 (2015)
8. Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., Van Gool, L.: Dslr-quality photos on mobile devices with deep convolutional networks. In: the IEEE Int. Conf. on Computer Vision (ICCV) (2017)
9. Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., Van Gool, L.: Wespe: weakly supervised photo enhancer for digital cameras. arXiv preprint arXiv:1709.01118 (2017)
10. Ignatov, A., Timofte, R., et al.: Pirm challenge on perceptual image enhancement on smartphones: Report. In: European Conference on Computer Vision Workshops (2018)
11. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. arXiv preprint (2017)
12. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision. pp. 694–711. Springer (2016)
13. Ki, S., Sim, H., Choi, J.S., Kim, S., Kim, M.: Fully end-to-end learning based conditional boundary equilibrium gan with receptive field sizes enlarged for single ultra-high resolution image dehazing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 817–824 (2018)
14. Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1646–1654 (2016)
15. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A.P., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. vol. 2, p. 4 (2017)
16. Li, B., Peng, X., Wang, Z., Xu, J., Feng, D.: Aod-net: All-in-one dehazing network. In: Proceedings of the IEEE International Conference on Computer Vision. vol. 1, p. 7 (2017)
17. Ling, Z., Fan, G., Wang, Y., Lu, X.: Learning deep transmission network for single image dehazing. In: Image Processing (ICIP), 2016 IEEE International Conference on. pp. 2296–2300. IEEE (2016)
18. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: Advances in neural information processing systems. pp. 2802–2810 (2016)
19. Ren, W., Liu, S., Zhang, H., Pan, J., Cao, X., Yang, M.H.: Single image dehazing via multi-scale convolutional neural networks. In: European conference on computer vision. pp. 154–169. Springer (2016)
20. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1874–1883 (2016)
21. Svoboda, P., Hradis, M., Barina, D., Zemcik, P.: Compression artifacts removal using convolutional neural networks. arXiv preprint arXiv:1605.00366 (2016)
22. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L., Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M., 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. pp. 1110–1121. IEEE (2017)
23. Timofte, R., Gu, S., Wu, J., Van Gool, L.: Ntire 2018 challenge on single image super-resolution: Methods and results. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2018)

24. Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Joint rain detection and removal via iterative region dependent multi-task learning. CoRR, abs/1609.07769 **2**, 3 (2016)
25. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26**(7), 3142–3155 (2017)
26. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. arXiv preprint (2018)
27. Zhang, X., Wu, R.: Fast depth image denoising and enhancement using a deep convolutional network. In: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on. pp. 2499–2503. IEEE (2016)
28. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint (2017)

Appendix. Results of the Proposed Method

Fig. 5. Visual results for our method.

Fig. 6. Visual results for our method.

Fig. 7. Visual results for our method.

Fig. 8. Visual results for our method.