DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks

Andrey Ignatov[1], Nikolay Kobyshev[1], Radu Timofte[1], Kenneth Vanhoey[1], Luc Van Gool[1,2]

[1] Computer Vision Laboratory, ETH Zürich, Switzerland   [2] ESAT - PSI, KU Leuven, Belgium

{ihnatova,nk,timofter,vanhoey,vangool}@vision.ee.ethz.ch

Abstract

Despite a rapid rise in the quality of built-in smartphone cameras, their physical limitations — small sensor size, compact lenses and the lack of specific hardware, — impede them to achieve the quality results of DSLR cameras. In this work we present an end-to-end deep learning approach that bridges this gap by translating ordinary photos into DSLR-quality images. We propose learning the translation function using a residual convolutional neural network that improves both color rendition and image sharpness. Since the standard mean squared loss is not well suited for measuring perceptual image quality, we introduce a composite perceptual error function that combines content, color and texture losses. The first two losses are defined analytically, while the texture loss is learned in an adversarial fashion. We also present DPED, a large-scale dataset that consists of real photos captured from three different phones and one high-end reflex camera. Our quantitative and qualitative assessments reveal that the enhanced image quality is comparable to that of DSLR-taken photos, while the methodology is generalized to any type of digital camera.

1. Introduction

During the last several years there has been a significant improvement in compact camera sensors quality, which has brought mobile photography to a substantially new level. Even low-end devices are now able to take reasonably good photos in appropriate lighting conditions, thanks to their advanced software and hardware tools for post-processing. However, when it comes to artistic quality, mobile devices still fall behind their DSLR counterparts. Larger sensors and high-aperture optics yield better photo resolution, color rendition and less noise, whereas their additional sensors help to fine-tune shooting parameters. These physical differences result in strong obstacles, making DSLR camera quality unattainable for compact mobile devices.

While a number of photographer tools for automatic image enhancement exist, they are usually focused on adjusting only global parameters such as contrast or brightness,

Figure 1: iPhone 3GS photo enhanced to DSLR-quality by our method. Best zoomed on screen.

without improving texture quality or taking image semantics into account. Besides that, they are usually based on a pre-defined set of rules that do not always consider the specifics of a particular device. Therefore, the dominant approach to photo post-processing is still based on manual image correction using specialized retouching software.

1.1. Related work

The problem of automatic image quality enhancement has not been addressed in its entirety in the area of computer vision, though a number of sub-tasks and related problems have been already successfully solved using deep learning techniques. Such tasks are usually dealing with image-to-image translation problems, and their common property is that they are targeted at removing artificially added artifacts to the original images. Among the related problems are the following:

Image super-resolution aims at restoring the original image from its downscaled version. In [4] a CNN architecture and MSE loss are used for directly learning low to high resolution mapping. It is the first CNN-based solution to achieve top performance in single image super-resolution, comparable with non-CNN methods [20]. The subsequent works developed deeper and more complex CNN architectures (*e.g.*, [10, 18, 16]). Currently, the best photo-realistic results on this task are achieved using a VGG-based loss function [9] and adversarial networks [12] that turned out to be efficient at recovering plausible high-frequency components.

Image deblurring/dehazing tries to remove artificially added haze or blur from the images. Usually, MSE is used

as a target loss function and the proposed CNN architectures consist of 3 to 15 convolutional layers [14, 2, 6] or are bi-channel CNNs [17].

Image denoising/sparse inpainting similarly targets removal of noise and artifacts from the pictures. In [28] the authors proposed weighted MSE together with a 3-layer CNN, while in [19] it was shown that an 8-layer residual CNN performs better when using a standard mean square error. Among other solutions are a bi-channel CNN [29], a 17-layer CNN [26] and a recurrent CNN [24] that was reapplied several times to the produced results.

Image colorization. Here the goal is to recover colors that were removed from the original image. The baseline approach for this problem is to predict new values for each pixel based on its local description that consists of various hand-crafted features [3]. Considerably better performance on this task was obtained using generative adversarial networks [8] or a 16-layer CNN with a multinomial cross-entropy loss function [27].

Image adjustment. A few works considered the problem of image color/contrast/exposure adjustment. In [25] the authors proposed an algorithm for automatic exposure correction using hand-designed features and predefined rules. In [23], a more general algorithm was proposed that – similarly to [3] – uses local description of image pixels for reproducing various photographic styles. A different approach was considered in [13], where images with similar content are retrieved from a database and their styles are applied to the target picture. All of these adjustments are implicitly included in our end-to-end transformation learning approach by design.

1.2. Contributions

The key challenge we face is dealing with all the aforementioned enhancements at once. Even advanced tools cannot notably improve image sharpness, texture details or small color variations that were lost by the camera sensor, thus we can not generate target enhanced photos from the existing ones. Corrupting DSLR photos and training an algorithm on the corrupted images does not work either: the solution would not generalize to real-world and very complex artifacts unless they are modeled and applied as corruptions, which is infeasible. To tackle this problem, we present a different approach: we propose to learn the transformation that modifies photos taken by a given camera to DSLR-quality ones. Thus, the goal is to learn a cross-distribution translation function, where the input distribution is defined by a given mobile camera sensor, and the target distribution by a DSLR sensor. To supervise the learning process, we create and leverage a dataset of images capturing the same scene with different cameras. Once the function is learned, it can be further applied to unseen photos at will.

Table 1: DPED camera characteristics.

Camera	Sensor	Image size	Photo quality
iPhone 3GS	3 MP	2048×1536	Poor
BlackBerry Passport	13 MP	4160×3120	Mediocre
Sony Xperia Z	13 MP	2592×1944	Average
Canon 70D DSLR	20 MP	3648×2432	Excellent

Figure 2: The rig with the four DPED cameras from Table 1.

Our main contributions are

- A novel approach for the photo enhancement task based on learning a mapping function between photos from mobile devices and a DSLR camera. The target model is trained in an end-to-end fashion without using any additional supervision or handcrafted features.
- A new large-scale dataset of over 6K photos taken synchronously by a DSLR camera and 3 low-end cameras of smartphones in a wide variety of conditions.
- A multi-term loss function composed of color, texture and content terms, allowing an efficient image quality estimation.
- Experiments measuring objective and subjective quality demonstrating the advantage of the enhanced photos over the originals and, at the same time, their comparable quality with the DSLR counterparts.

The remainder of the paper is structured as follows. In Section 2 we describe the new DPED dataset. Section 3 presents our architecture and the chosen loss functions. Section 4 shows and analyzes the experimental results. Finally, Section 5 concludes the paper.

2. DSLR Photo Enhancement Dataset (DPED)

In order to tackle the problem of image translation from poor quality images captured by smartphone cameras to superior quality images achieved by a professional DSLR camera, we introduce a large-scale real-world dataset, namely the "DSLR Photo Enhancement Dataset" (DPED)[1], that can be used for the general photo quality enhancement task. DPED consists of photos taken in the wild synchronously by three smartphones and one DSLR camera.

---

[1] dped-photos.vision.ee.ethz.ch

iPhone BlackBerry Sony Canon

Figure 3: Example quadruplets of images taken synchronously by the DPED four cameras.

The devices used to collect the data are described in Table 1 and example quadruplets can be seen in Figure 3.

To ensure that all cameras were capturing photos simultaneously, the devices were mounted on a tripod and activated remotely by a wireless control system (see Figure 2). In total, over 22K photos were collected during 3 weeks, including 4549 photos from Sony smartphone, 5727 from iPhone and 6015 photos from each Canon and BlackBerry cameras. The photos were taken during the daytime in a wide variety of places and in various illumination and weather conditions. The photos were captured in automatic mode, and we used default settings for all cameras throughout the whole collection procedure.

Matching algorithm. The synchronously captured images are not perfectly aligned since the cameras have different viewing angles and positions as can be seen in Figure 3. To address this, we performed additional non-linear transformations resulting in a fixed-resolution image that our network takes as an input. The algorithm goes as follows (see Fig. 4). First, for each (phone-DSLR) image pair, we compute and match SIFT keypoints [15] across the images. These are used to estimate a homography using RANSAC [21]. We then crop both images to the intersection part and downscale the DSLR image crop to the size of the phone crop.

Training CNN on the aligned high-resolution images is infeasible, thus patches of size 100×100px were extracted from these photos. Our preliminary experiments revealed that larger patch sizes do not lead to better performance, while requiring considerably more computational resources. We extracted patches using a non-overlapping sliding window. The window was moving in parallel along
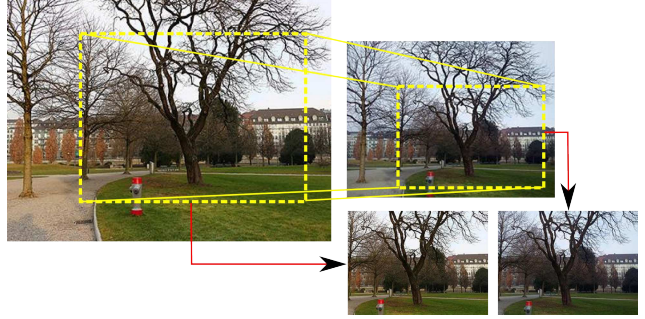

Figure 4: Matching algorithm: an overlapping region is determined by SIFT descriptor matching, followed by a non-linear transform and a crop resulting in two images of the same resolution representing the same scene. Here: Canon and BlackBerry images, respectively.

both images from each phone-DSLR image pair, and its position on the phone image was additionally adjusted by shifts and rotations based on the cross-correlation metrics. To avoid significant displacements, only patches with cross-correlation greater than 0.9 were included in the dataset. Around 100 original images were reserved for testing, the rest of the photos were used for training and validation. This procedure resulted in 139K, 160K and 162K training and 2.4-4.3K test patches for BlackBerry-Canon, iPhone-Canon and Sony-Canon pairs, respectively. It should be emphasized that both training and test patches are precisely matched, the potential shifts do not exceed 5 pixels. In the following we assume that these patches of size $3{\times}100{\times}100$ constitute the input data to our CNNs.

3. Method

Given a low-quality photo I_s (source image), the goal of the considered enhancement task is to reproduce the image I_t (target image) taken by a DSLR camera. A deep residual CNN $F_\mathbf{W}$ parameterized by weights $\mathbf{W}$ is used to learn the underlying translation function. Given the training set $\{I_s^j, I_t^j\}_{j=1}^N$ consisting of N image pairs, it is trained to minimize:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \frac{1}{N} \sum_{j=1}^{N} \mathcal{L}\big(F_\mathbf{W}(I_s^j), I_t^j\big), \qquad (1)$$

where $\mathcal{L}$ denotes a multi-term loss function we detail in section 3.1. We then define the system architecture of our solution in Section 3.2.

3.1. Loss function

The main difficulty of the image enhancement task is that input and target photos cannot be matched densely (*i.e.*, pixel-to-pixel): different optics and sensors cause specific local non-linear distortions and aberrations, leading

Figure 5: Fragments from the original and blurred images taken by the phone (two left-most) and DSLR (two right-most) camera. Blurring removes high-frequencies and makes color comparison easier.

to a non-constant shift of pixels between each image pair even after precise alignment. Hence, the standard per-pixel losses, besides being doubtful as a perceptual quality metric, are not applicable in our case. We build our loss function under the assumption that the overall perceptual image quality can be decomposed into three independent parts: i) color quality, ii) texture quality and iii) content quality. We now define loss functions for each component, and ensure invariance to local shifts by design.

3.1.1 Color loss

To measure the color difference between the enhanced and target images, we propose applying a Gaussian blur (see Figure 5) and computing Euclidean distance between the obtained representations. In the context of CNNs, this is equivalent to using one additional convolutional layer with a fixed Gaussian kernel followed by the mean squared error (MSE) function. Color loss can be written as:

$$\mathcal{L}_{\mathrm{color}}(X, Y) = \|X_b - Y_b\|_2^2, \qquad (2)$$

where X_b and Y_b are the blurred images of X and Y, resp.:

$$X_b(i, j) = \sum_{k,l} X(i + k, j + l) \cdot G(k, l), \qquad (3)$$

and the 2D Gaussian blur operator is given by

$$G(k, l) = A \exp\left(-\frac{(k - \mu_x)^2}{2\sigma_x} - \frac{(l - \mu_y)^2}{2\sigma_y}\right) \qquad (4)$$

where we defined $A = 0.053$, $\mu_{x,y} = 0$, and $\sigma_{x,y} = 3$.

The idea behind this loss is to evaluate the difference in brightness, contrast and major colors between the images while eliminating texture and content comparison. Hence, we fixed a constant σ by visual inspection as the smallest value that ensures that texture and content are dropped. The crucial property of this loss is its invariance to small distortions. Figure 6 demonstrates the MSE and Color losses for image pairs (X, Y), where Y equals X shifted in a random direction by n pixels. As one can see, color loss is nearly insensitive to small distortions ($\leqslant 2$ pixels). For higher shifts
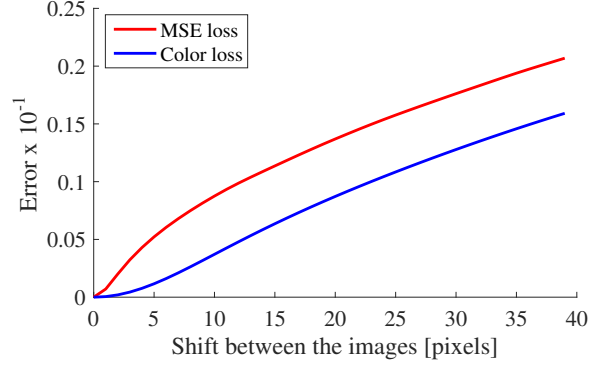

Figure 6: Comparison between MSE and color loss as a function of the magnitude of shift between images. Results were averaged over 50K images.

(3-5px), it is still about 5-10 times smaller compared to the MSE, whereas for larger displacements it demonstrates similar magnitude and behavior. As a result, color loss forces the enhanced image to have the same color distribution as the target one, while being tolerant to small mismatches.

3.1.2 Texture loss

Instead of using a pre-defined loss function, we build upon generative adversarial networks (GANs) [5] to directly learn a suitable metric for measuring texture quality. The discriminator CNN is applied to grayscale images so that it is targeted specifically on texture processing. It observes both fake (improved) and real (target) images, and its goal is to predict whether the input image is real or not. It is trained to minimize the cross-entropy loss function, and the texture loss is defined as a standard generator objective:

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

where $F_{\mathbf{W}}$ and D denote the generator and discriminator networks, respectively. The discriminator is pre-trained on the {phone, DSLR} image pairs, and then trained jointly with the proposed network as is conventional for GANs. It should be noted that this loss is shift-invariant by definition since no alignment is required in this case.

3.1.3 Content loss

Inspired by [9, 12], we define our content loss based on the activation maps produced by the ReLU layers of the pre-trained VGG-19 network. Instead of measuring per-pixel difference between the images, this loss encourages them to have similar feature representation that comprises various aspects of their content and perceptual quality. In our case it

is used to preserve image semantics since other losses don't consider it. Let $\psi_j()$ be the feature map obtained after the j-th convolutional layer of the VGG-19 CNN, then our content loss is defined as Euclidean distance between feature representations of the enhanced and target images:

$$\mathcal{L}_{\text{content}} = \frac{1}{C_j H_j W_j} \|\psi_j\big(F_{\mathbf{W}}(I_s)\big) - \psi_j\big(I_t\big)\|, \quad (6)$$

where C_j, H_j and W_j denotes the number, height and width of the feature maps, and $F_{\mathbf{W}}(I_s)$ the enhanced image.

3.1.4 Total variation loss

In addition to previous losses, we add total variation (TV) loss [1] to enforce spatial smoothness of the produced images:

$$\mathcal{L}_{\text{tv}} = \frac{1}{CHW} \|\nabla_x F_{\mathbf{W}}(I_s) + \nabla_y F_{\mathbf{W}}(I_s)\|, \quad (7)$$

where C, H and W are the dimensions of the generated image $F_{\mathbf{W}}(I_s)$. As it is relatively lowly weighted (see Eqn. 8), it does not harm high-frequency components while it is quite effective at removing salt-and-pepper noise.

3.1.5 Total loss

Our final loss is defined as a weighted sum of previous losses with the following coefficients:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{content}} + 0.4 \cdot \mathcal{L}_{\text{texture}} + 0.1 \cdot \mathcal{L}_{\text{color}} + 400 \cdot \mathcal{L}_{\text{tv}}, \quad (8)$$

where the content loss is based on the features produced by the *relu*_5_4 layer of the VGG-19 network. The coefficients were chosen based on preliminary experiments on the DPED training data.

3.2. Generator and Discriminator CNNs

Figure 7 illustrates the overall architecture of the proposed CNNs. Our image transformation network is fully-convolutional, and starts with a 9×9 layer followed by four residual blocks. Each residual block consists of two 3×3 layers alternated with batch-normalization layers. We use two additional layers with kernels of size 3×3 and one with 9×9 kernels after the residual blocks. All layers in the transformation network have 64 channels and are followed by a *ReLU* activation function, except for the last one, where a scaled *tanh* is applied to the outputs.

The discriminator CNN consists of five convolutional layers each followed by a *LeakyReLU* nonlinearity and batch normalization. The first, second and fifth convolutional layers are strided with a step size of 4, 2 and 2, respectively. A sigmoidal activation function is applied to the outputs of the last fully-connected layer containing 1024 neurons and produces a probability that the input image was taken by the target DSLR camera.
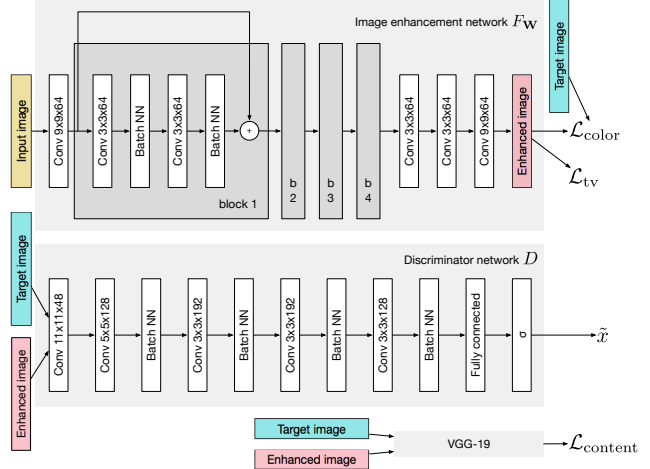

Figure 7: The overall architecture of the proposed system.

3.3. Training details

The network was trained on a *NVidia Titan X* GPU for 20K iterations using a batch size of 50. The parameters of the network were optimized using *Adam* [11] modification of stochastic gradient descent with a learning rate of 5e-4. The whole pipeline and experimental setup was identical for all cameras.

4. Experiments

Our general goal to "improve image quality" is subjective and hard to evaluate quantitatively. We suggest a set of tools and methods from the literature that are most relevant to our problem. We use them, as well as our proposed method, on a set of test images taken by mobile devices and compare how close the results are to the DSRL shots.

In section 4.1, we present the methods we compare to. Then we present both objective and subjective evaluations: the former w.r.t. the ground truth reference (*i.e.*, the DSLR images) in section 4.2, the latter with no-reference subjective quality scores in section 4.3. Finally, section 4.4 analyzes the limitations of the proposed solution.

4.1. Benchmark methods

In addition to our proposed photo enhancement solution, we compare with the following tools and methods.

Apple Photo Enhancer (APE) is a commercial product known to generate among the best visual results, while the algorithm is unpublished. We trigger the method using the automatic *Enhance* function from the *Photos* app. It performs image improvement without taking any parameters.

Dong *et al*. [4] is a fundamental baseline super-resolution method, thus addredding a task related to end-to-end image-to-image mapping. Hence we chose it to apply

Figure 8: From left to right, top to bottom: original iPhone photo and the same image after applying, respectively: APE, Dong *et al*. [4], Johnson *et al*. [9], our generator network, and the corresponding DSLR image.

Table 2: Average PSNR/SSIM results on DPED test images.

Phone	APE		Dong et al. [4]		Johnson et al. [9]		**Ours**	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
iPhone	17.28	0.8631	19.27	0.8992	**20.32**	0.9161	20.08	**0.9201**
BlackBerry	18.91	0.8922	18.89	0.9134	**20.11**	0.9298	20.07	**0.9328**
Sony	19.45	0.9168	21.21	0.9382	21.33	0.9434	**21.81**	**0.9437**

on our task and compare with. The method relies on a standard 3-layer CNN and MSE loss function and maps from low resolution / corrupted image to the restored image.

Johnson *et al*. [9] is one of the latest state of the art in photo-realistic super-resolution and style transferring tasks. The method is based on a deep residual network (with four residual blocks, each consisting of two convolutional layers) that is trained to minimize a VGG-based loss function.

Manual enhancement. We asked a graphical artist to enhance color, sharpness and general look-and-feel of 9 images using professional software (Adobe Photoshop CS6). A time limit of one workday was given, so as to simulate a realistic scenario.

Figure 8 illustrates the ensemble of enhancement methods we consider for comparison in our experiments. Dong *et al*. [4] and Johnson *et al*. [9] are trained using the same train image pairs as for our solution for each of the smartphones from the DPED dataset.

4.2. Quantitative evaluation

We first quantitatively compare APE, Dong *et al*. [4], Johnson *et al*. [9] and our method on the task of mapping photos from three low-end cameras to the high-quality DSLR (Canon) images and report the results in Table 2. As such, we do not evaluate global image quality but, rather, we measure resemblance to a reference (the ground truth DSLR image). We use classical distance metrics, namely PSNR and SSIM scores: the former measures signal distortion w.r.t. the reference, the latter measures structural similarity which is known to be a strong cue for perceived quality [22]. First, one can note that our method is the best in terms of SSIM, at the same time producing images that are cleaner and sharper, thus perceptually performs the best. On PSNR terms, our method competes with the state of the art: it slightly improves or worsens depending on the dataset, *i.e.*, on the actual phone used. Alignment issues could be responsible for these minor variations, and thus we consider Johnson *et al*.'s method [4] and ours equivalent here, while outperforming other methods. In Fig. 8 we show visual re-
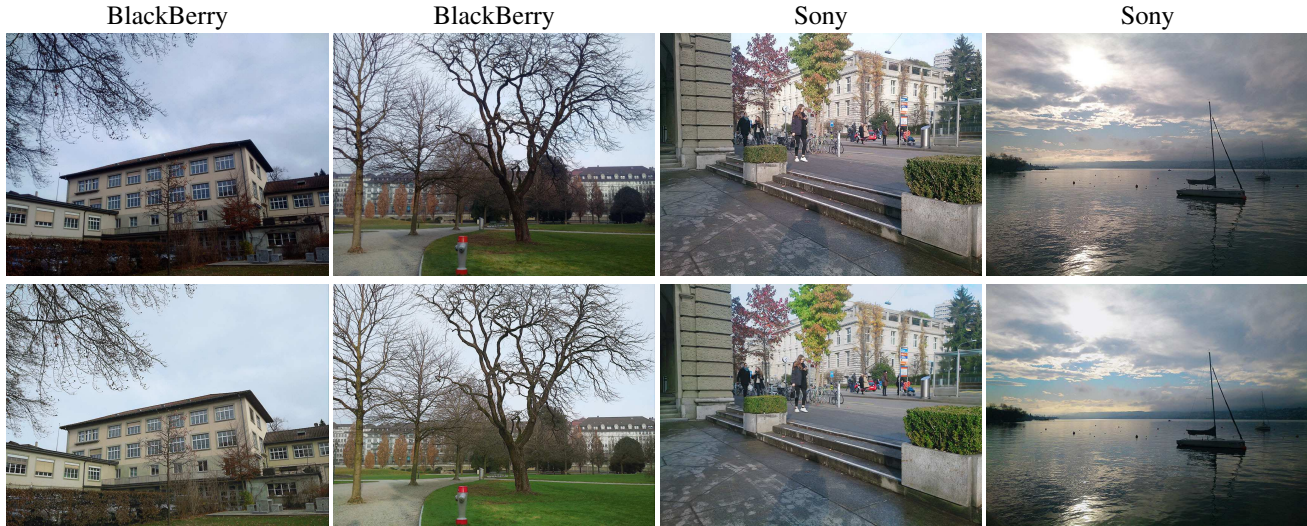
| BlackBerry | BlackBerry | Sony | Sony |

Figure 9: Four examples of original (top) vs. enhanced (bottom) images captured by BlackBerry and Sony cameras.

sults comparing to the source photo (iPhone) and the target DSLR photo (Canon). More results are in the supplementary material.

4.3. User study

Our goal is to produce DSLR-quality images for the end user of smartphone cameras. To measure overall quality we designed a no-reference user study where subjects are repeatedly asked to choose the better looking picture out of a displayed pair. Users were instructed to ignore precise picture composition errors (*e.g.*, field of view, perspective variation, *etc*.). There was no time limit given to the participants, images were shown in full resolution and the users were allowed to zoom in and out at will. In this setting, we did the following pairwise comparisons (every group of experiments contains 3 classes of pictures, the users were shown all possible pairwise combinations of these classes):

(i) Comparison between:

- original low-end phone photos,
- DSLR photos,
- photos enhanced by our proposed method.

At every question, the user is shown two pictures from different categories (original, DSLR or enhanced). 9 scenes were used for each phone (*e.g.*, see Fig. 11). In total, there are 27 questions for every phone, thus 81 in total.

(ii) Additionally, we compared (iPhone images only):

- photos enhanced by the proposed method,
- photos enhanced manually (by a professional),
- photos enhanced by APE.

We again considered 9 images that resulted in 27 binary selection questions. Thus, in total the study consists of 108

binary questions. All pairs are shuffled randomly for every subject, as is the sequence of displayed images. 42 subjects unaware of the goal of this work participated. They are mostly young scientists with a computer science background.

Figure 10 shows results: for every experiment the first 3 bars show the results of the pairwise comparison averaged over the 9 images shown, while the last bar shows the fraction of cases when the selected method was chosen over all experiments.

The subfigures 10a-c show the results of enhancing photos from 3 different mobile devices. It can be seen that in all cases both pictures taken with a DSLR as well as pictures enhanced by the proposed CNN are picked much more often than the original ones taken with the mobile devices. When subjects are asked to select the better picture among the DSLR-picture and our enhanced picture, the choice is almost random (see the third bar in subfigures 10a-c). This means that the quality difference is inexistent or indistinguishable, and users resort to chance.

Subfigure 10d shows user choices among our method, human artist work, and APE. Although human enhancement turns out to be slightly preferred to the automatic APE, the images enhanced by our method are picked more often, outperforming even manual retouching.

We can conclude that our results are of on pair quality compared to DSLR images, while starting from low quality phone cameras. The human subjects are unable to distinguish between them – the preferences are equally distributed.

4.4. Limitations

Since the proposed enhancement process is fully-automated, some flaws are inevitable. Two typical artifacts
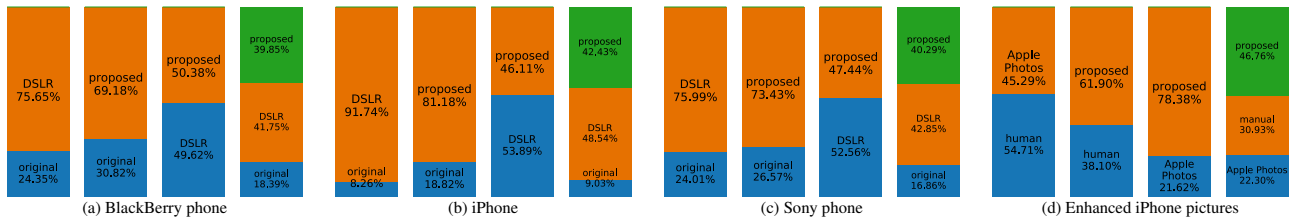
Figure 10: User study: results of pairwise comparisons. In every subfigure, the first three bars show the result of the pairwise experiments, while the last bar shows the distribution of the aggregated scores.

Figure 11: The 9 scenes shown to the participants of the user study. Here: BlackBerry images enhanced using our technique.
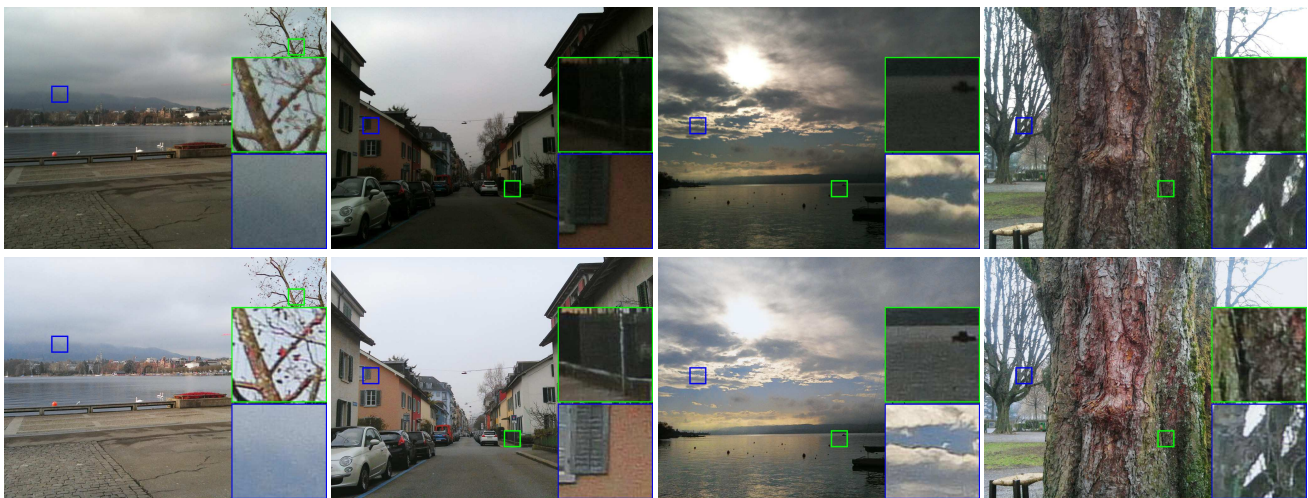

Figure 12: Typical artifacts generated by our method (2nd row) compared with original iPhone images (1st row)

that can appear on the processed images are color deviations (see ground/mountains in first image of Fig. 12) and too high contrast levels (second image). Although they often cause rather plausible visual effects, in some situations this can lead to content changes that may look artificial, *i.e.* greenish asphalt in the second image of Fig. 12. Another notable problem is noise amplification – due to the nature of GANs, they can effectively restore high frequency-components. However, high-frequency noise is emphasized too. Fig. 12 (2nd and 3rd images) shows that a high noise in the original image is amplified in the enhanced image. Note that this noise issue occurs mostly on the lowest-quality photos (*i.e.*, from the iPhone), not on the better phone cameras.

Finally, the need of a strong supervision in the form of matched source/target training image pairs makes the process tedious to repeat for other cameras. To overcome this, we propose a weakly-supervised approach in [7] that does not require the mentioned correspondence.

5. Conclusions

We proposed a photo enhancement solution to effectively transform cameras from common smartphones into high quality DSLR cameras. Our end-to-end deep learning approach uses a composite perceptual error function that combines content, color and texture losses. To train and evaluate our method we introduced DPED – a large-scale dataset that consists of real photos captured from three different phones and one high-end reflex camera, and suggested an efficient way of calibrating the images so that they are suitable for image-to-image learning. Our quantitative and qualitative assessments reveal that the enhanced images demonstrate a quality comparable to DSLR-taken photos, and the method itself can be applied to cameras of various quality levels.

References

[1] H. A. Aly and E. Dubois. Image up-sampling using total-variation regularization with a new observation model. *IEEE Transactions on Image Processing*, 14(10):1647–1659, Oct 2005. 5

[2] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, Nov 2016. 2

[3] Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[4] C. Dong, C. C. Loy, K. He, and X. Tang. *Learning a Deep Convolutional Network for Image Super-Resolution*, pages 184–199. Springer International Publishing, Cham, 2014. 1, 5, 6

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 4

[6] M. Hradiš, J. Kotera, P. Zemčík, and F. Šroubek. Convolutional neural networks for direct text deblurring. In *Proceedings of BMVC 2015*. The British Machine Vision Association and Society for Pattern Recognition, 2015. 2

[7] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey, and L. Van Gool. Wespe: Weakly supervised photo enhancer for digital cameras. 2017. 8

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016. 2

[9] J. Johnson, A. Alahi, and L. Fei-Fei. *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*, pages 694–711. Springer International Publishing, Cham, 2016. 1, 4, 6

[10] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1646–1654, June 2016. 1

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[12] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 4

[13] J.-Y. Lee, K. Sunkavalli, Z. Lin, X. Shen, and I. So Kweon. Automatic content-aware color and tone stylization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[14] Z. Ling, G. Fan, Y. Wang, and X. Lu. Learning deep transmission network for single image dehazing. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2296–2300, Sept 2016. 2

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 3

[16] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2802–2810. Curran Associates, Inc., 2016. 1

[17] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. *Single Image Dehazing via Multi-scale Convolutional Neural Networks*, pages 154–169. Springer International Publishing, Cham, 2016. 2

[18] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1

[19] P. Svoboda, M. Hradis, D. Barina, and P. Zemčík. Compression artifacts removal using convolutional neural networks. *CoRR*, abs/1605.00366, 2016. 2

[20] R. Timofte, V. De Smet, and L. Van Gool. *A+: Adjusted Anchored Neighborhood Regression for Fast Super-Resolution*, pages 111–126. Springer International Publishing, Cham, 2015. 1

[21] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. 3

[22] Z. 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, April 2004. 6

[23] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu. Automatic photo adjustment using deep neural networks. *ACM Trans. Graph.*, 35(2):11:1–11:15, Feb. 2016. 2

[24] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Joint rain detection and removal via iterative region dependent multi-task learning. *CoRR*, abs/1609.07769, 2016. 2

[25] L. Yuan and J. Sun. *Automatic Exposure Correction of Consumer Photographs*, pages 771–785. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. 2

[26] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 2017. 2

[27] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. *ECCV*, 2016. 2

[28] X. Zhang and R. Wu. Fast depth image denoising and enhancement using a deep convolutional network. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2499–2503, March 2016. 2

[29] E. Zhou, H. Fan, Z. Cao, Y. Jiang, and Q. Yin. Learning face hallucination in the wild. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 3871–3877. AAAI Press, 2015. 2