

DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks

Andrey Ignatov, Nikolay Kobyshev, Kenneth Vanhoey, Radu Timofte, Luc Van Gool

ETH Zurich

andrey.ignatoff@gmail.com, {nk, vanhoe, timofter, vangoool}@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-produced 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 using an adversarial network. We also present 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 images demonstrate the quality comparable to DSLR-taken photos, while the method itself can be applied to any type of digital cameras.

1 Introduction

During the last several years there has been a significant quality improvement of compact camera sensors, 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, 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.

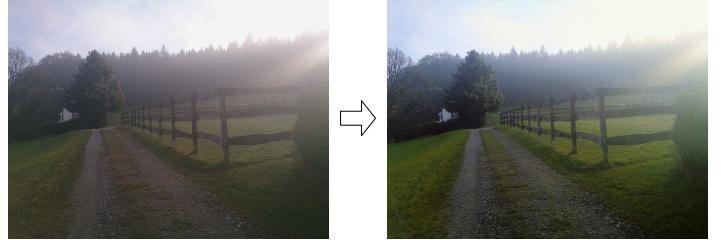


Figure 1: Sony smartphone image enhanced to DSLR-quality by our method. Best zoomed on screen.

1.1 Related work

The problem of image quality enhancement has not been entirely addressed 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 down-scaled version. In [12] 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 [26]. The subsequent works developed deeper and more complex CNN architectures (e.g., [3, 6, 4]). Currently, the best photo-realistic results on this task are achieved using a VGG-based loss function [24] and adversarial networks [21] 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 3–15 convolutional layers [11, 1, 22] or are bi-channel CNNs [2].

Image denoising/sparse inpainting similarly targets removal of noise and artifacts from the pictures. In [16] the authors proposed weighted MSE together with a 3-layer CNN, while in [25] 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 [10], a 17-layer CNN [17] and a recurrent CNN [13] 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 [19].

Table 1: DPED camera characteristics.

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



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

Considerably better performance on this task was obtained using generative adversarial networks [18] or a 16-layer CNN with multinomial cross-entropy loss function [20].

Image adjustment. A few works considered the problem of image color/contrast/exposure adjustment. In [9] the authors proposed an algorithm for automatic exposure correction using hand-designed features and predefined rules. In [5], a more general algorithm was proposed that similarly to [19] used local description of image pixels for reproducing various photographic styles. A different approach was considered in [8], where images with similar content were retrieved from the database and their styles were 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: instead of focusing on images from one particular device, we propose transforming the original photos into photos obtained from DSLR camera that contain the same scene. Thus, the goal is to recover a cross-distribution translation function, where the input distribution is defined by a mobile camera sensor, and the target distribution by a DSLR sensor. Once the function is learned on a set of such pairs or images, it can be further applied to unseen photos at will.

We make the following main contributions:

- A novel approach for the photo enhancement task based on learning a mapping function between photos from mobile de-



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

vices 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 complex loss function for an efficient image quality estimation that consists of color, texture and content terms.
- Several evaluation experiments measuring objective and subjective quality, which demonstrate the advantage of the enhanced photos over the originals, and the parity with their DSLR counterparts.

The remainder of the paper is structured as follows. In Section 2 we describe the novel dataset for photo enhancement. Section 3 presents our architecture and the chosen loss functions. Finally, Section 4 shows and analyzes the experimental results.

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), 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. 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 and 6064 photos from each Canon, BlackBerry

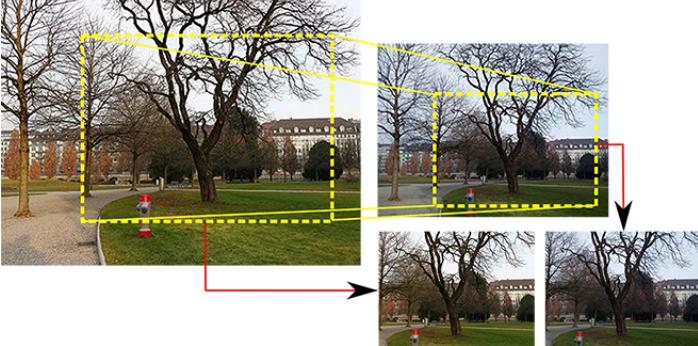


Figure 4: Matching algorithm: an overlapping region is determined by SIFT descriptor matching, followed by a non-linear transform and a crop resulting in two images of size 1080x800 representing the same scene. Here: Canon and BlackBerry images, respectively.

and iPhone 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 [27]. We then crop both images to the intersection part and downscale the largest image crop (usually from DSLR image) to the size of the smallest. Finally, we crop both images to the largest region respecting a pre-defined aspect ratio (i.e., 1080:800) and downsample the result to the preset resolution. We chose that resolution to be 1080x800px as it fits most intersections, even from the lower-resolution iPhone images.

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_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:

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

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



Figure 5: Fragments from the original and blurred images taken by the phone (two left-most) and DSLR (two right-most) camera. After blurring the hi-frequency differences between the images are gone that makes color comparison easier.

3.1 Loss function

The major 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 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:

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

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

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

where $G(k, l)$ is the Gaussian blur

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

with $A = 0.053$, $\mu_{x,y} = 0$, and $\sigma_{x,y} = 3$ in our experiments.

The idea behind this loss is to evaluate brightness, contrast and major colors between the images by eliminating texture and content from the images. Hence we fixed a constant σ by visual inspection as the smallest value that ensures that texture and contrasts are eliminated. 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 was obtained by shifting X in a random direction by n pixels. As one can see, color loss is almost not sensitive to small distortions (1-2 pixels). For higher shifts (3-5px), it is still about 5-10 times smaller compared

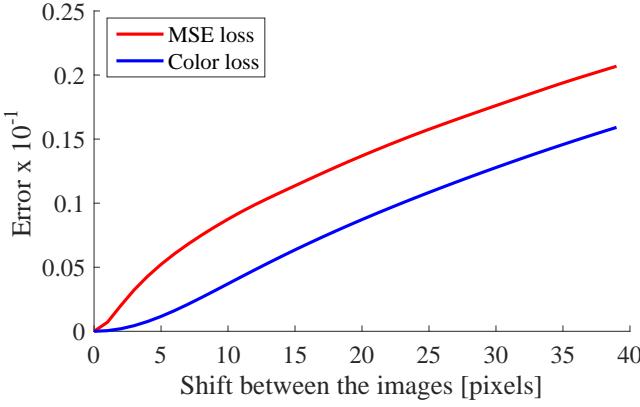


Figure 6: Comparison between MSE and color loss as a function of the magnitude of shift between images. Results were averaged over 50K images.

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) [28] to directly learn a suitable metrics for measuring texture quality. In this work we deviate from the standard GAN architecture: we use a bi-channel CNN-discriminator that observes both fake (improved) and real (target) images. The reasoning behind this is that it is easier to learn tiny differences when observing samples from both distributions, rather than trying to predict the result without any ground truth. In particular, the considered CNN achieves the result of 99% after 300 iterations when trained to differentiate between phone and DSLR images, while its single-channel counterpart shows only 86% after 5000 steps.

The discriminator CNN is applied to grayscale images so that it is targeted specifically on texture processing. The fake and real photos in each input pair are randomly permuted, and the goal of the discriminator is to predict whether the first image in the pair is real or not. The network is trained to minimize the cross-entropy loss function, and the texture loss is defined as a standard generator objective:

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

where $F_{\mathbf{W}}$ and D denote the generator and discriminator networks, respectively. It should be noted that this loss is shift-invariant by definition: it is learned on images that are not perfectly aligned.

3.1.3 Content loss

Inspired by [24, 21], 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 preserves image semantics

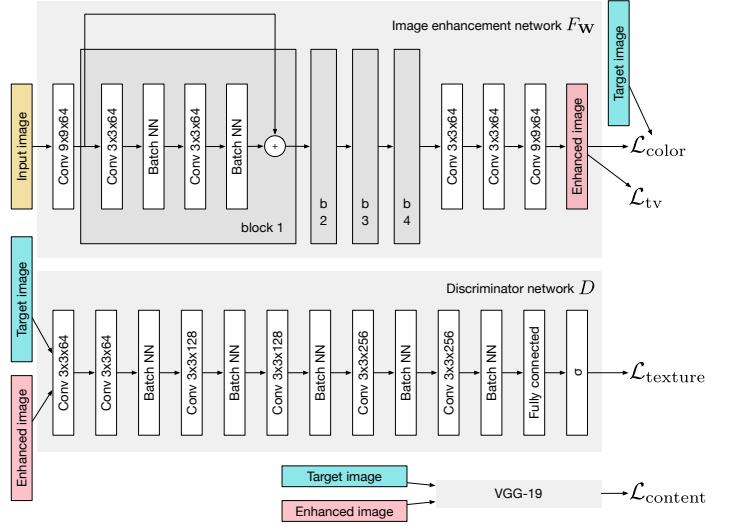


Figure 7: The overall architecture of the proposed system.

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:

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

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

3.1.4 Total variation loss.

In addition to previous losses, we add total variation loss [7] to enforce spatial smoothness of the produced images. The loss is defined as:

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

where C , H and W are the dimensions of the generated image $F_{\mathbf{W}}(I_s)$.

3.1.5 Total loss

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

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{color}} + \mathcal{L}_{\text{texture}} + 10^{-1} \cdot \mathcal{L}_{\text{content}} + 10^2 \cdot \mathcal{L}_{\text{tv}}, \quad (8)$$

where the content loss is based on the features produced by *relu_5_2* layer of VGG-19 network. The coefficients were chosen based on preliminary experiments on the DPED training data.

3.2 System Architecture

This section describes the implementation details of the generator and discriminator CNNs, and gives an overview of the whole pipeline of the proposed solution.



Figure 8: From left to right, top to bottom: original iPhone photo and the same image after applying, respectively: APE, Dong et al. [12], Johnson et al. [24], our generator network, and the corresponding DSLR image.

3.2.1 Data preprocessing

Training CNN on the aligned high-resolution images is infeasible, thus patches of size 100×100 px 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 extract patches using a non-overlapping sliding window. The window was moving in parallel along 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. 100 original images were reserved for testing and validation respectively, the rest of the photos were used for training. This procedure resulted in 65K, 92K and 97K training and 2K validation and test patches for Sony-Canon, BlackBerry-Canon and iPhone-Canon pairs, respectively. In the following we assume that these patches of size $3 \times 100 \times 100$ constitute the input data to our CNNs.

3.2.2 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 convolutional 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 six convolutional layers each followed by *LeakyReLU* nonlinearity and batch normaliza-

tion. The number of channels in these layers increases from 64 to 256, doubling after each second layer, where a strided convolution with a step size of 2 is used to reduce the size of the feature maps. Sigmoidal activation function is applied to the outputs of the last fully-connected layer containing 1024 neurons and produces a probability that the first image in the pair is a DSLR photo.

3.2.3 Training details

The network was trained on an *Nvidia Titan X* GPU for 20K iterations using a batch size of 50. The parameters of the network were optimized using *Adam* [23] 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 the 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 with respect to 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.

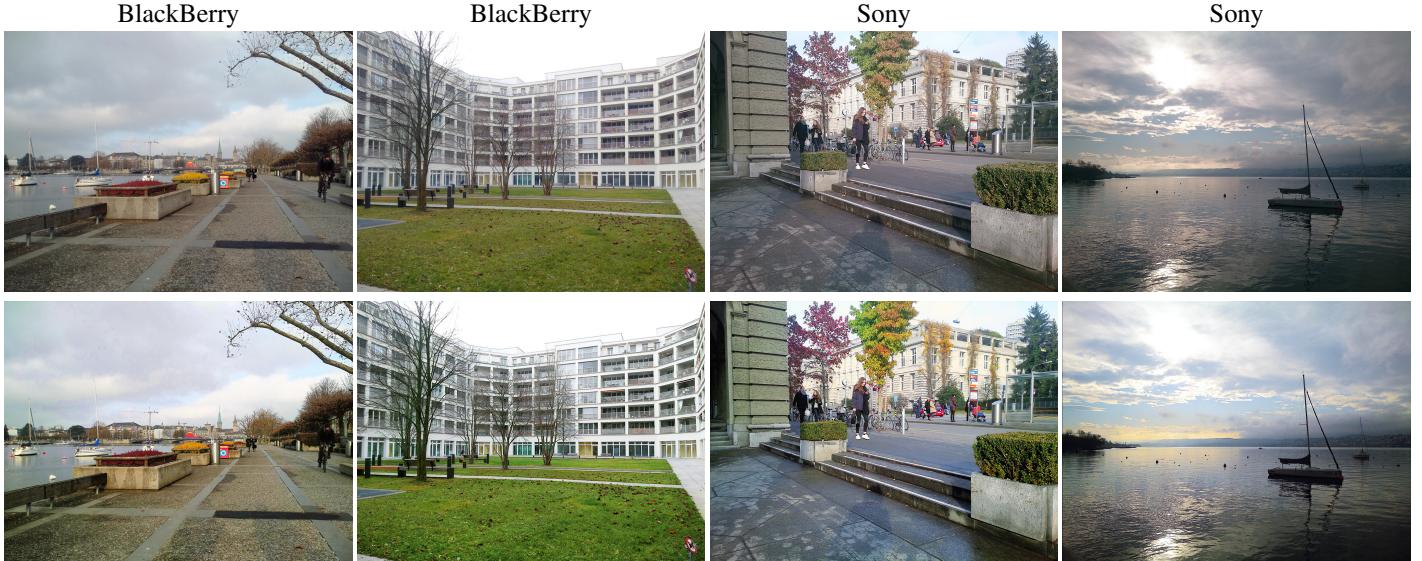


Figure 9: Four examples of original (top) vs. enhanced (bottom) images captured by BlackBerry and Sony cameras.

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. The method is called from the Photos app with automatic Enhance function and performs image improvement without taking any parameters.

Dong et al. [12] is a fundamental baseline super-resolution method that is based on a standard 3-layers CNN and MSE loss function and maps from low resolution / corrupted image to the restored image.

Johnson et al. [24] – one of the latest state of the art in photorealistic 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 improve the quality of the image using professional software (Adobe Photoshop CS6). The task was to increase the color, sharpness and general look and feel of the image.

Figure 8 illustrates the ensemble of enhancement methods we consider for comparison in our experiments. Dong et al. [12] and Johnson et al. [24] are trained using the same train image pairs as for our solution for each of the smartphones from our DPED.

4.2 Quantitative evaluation

We first quantitatively compare APE, Dong et al. [12], Johnson et al. [24] 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 3. 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 with respect to the reference, the latter measures structural similarity which is known to be a strong cue for perceived quality [14]. First, one can note that our method is the best in terms of

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

Phone	APE		Dong et al. [12]		Johnson et al. [24]		Ours	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
iPhone	17.28	0.8631	19.90	0.8813	19.65	0.9003	19.16	0.9114
BlackBerry	18.91	0.8922	19.34	0.8998	19.31	0.9229	19.51	0.9259
Sony	19.45	0.9168	22.00	0.9291	21.07	0.9378	21.97	0.9400

SSIM, 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 Dong et al.’s method [12] and ours equivalent, while outperforming other methods. In Fig. 8 we show visual results in comparison with the original iPhone photo and the target DSLR image (Canon camera). More results are included 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 of images. Users were instructed to ignore precise picture composition errors (e.g., field of view, perspective variation, etc).

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 the proposed method.

At every question, the user is shown two pictures from different categories (original, DSLR or enhanced). For every of 3 phones 9 images are shown (the example of enhanced images for the BlackBerry phone are shown in Fig. 11). This results in 27 questions for every phone and 81 questions in total.

(ii) For the dataset taken by the iPhone we compared:

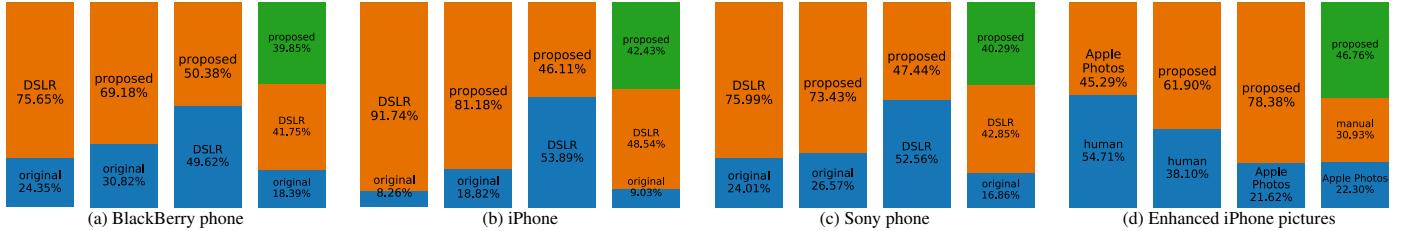


Figure 10: User study: results of pairwise comparisons. In every subfigure, the first three bars show the result of the pairwise experiments, while the last bar shows the distribution of the aggregated scores.



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



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

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

We again considered 9 images that resulted in 27 binary selection questions.

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 computer science background.

The study results are presented in Figure 10. In this figure, 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 a-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 between the picture produced by a DSLR and a picture enhanced by our method, the images are picked almost randomly (see the third bar in subfigures a-c). This means that it is difficult for the subjects to distinguish between the quality of the images taken by a DSLR camera and the high quality images as enhanced by the proposed method which targets DSLR quality.

The subfigure d) shows the results of comparison of 9 iPhone images enhanced by the proposed method, a human working with a professional retouching software and a default Apple Photos one-click enhancer. Although human enhancement turns out to be slightly preferred to the automatic enhancement of Apple Photos, the images enhanced by the proposed CNN are picked the most often, outperforming even the results of manual retouching.

We can conclude that our results produce on pair quality images to the DSLR, 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 major artifacts occurring on the processed images include color deviations and too high contrast level (see first image in Fig. 12). Although they often cause rather plausible visual effects, in some situations this can lead to content changes that may look artificial, i.e. green asphalt in the second image of Fig. 12. Another notable problem is noise amplification – due to the nature of generative adversarial networks, they can effectively restore high frequency-components. However, high-frequency noise is emphasized too. Fig. 12 (third image) shows a case where the high noise in the original image is amplified in the enhanced image. Note that this noise issue occurred only on photos from the poor iPhone 3Gs camera and not on the

photos from the better Sony and BlackBerry cameras.

5 Conclusion

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] 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. [1](#)
- [2] 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. [1](#)
- [3] 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](#)
- [4] 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](#)
- [5] 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](#)
- [6] 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](#)
- [7] 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. [4](#)
- [8] 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](#)
- [9] L. Yuan and J. Sun. *Automatic Exposure Correction of Consumer Photographs*, pages 771–785. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. [2](#)
- [10] 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. [1](#)
- [11] 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. [1](#)
- [12] 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](#)
- [13] 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. [1](#)
- [14] 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](#)
- [15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. [3](#)
- [16] 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. [1](#)
- [17] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *CoRR*, abs/1608.03981, 2016. [1](#)
- [18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arxiv*, 2016. [2](#)
- [19] Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. [1, 2](#)
- [20] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. *ECCV*, 2016. [2](#)
- [21] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016. [1, 4](#)
- [22] 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. [1](#)
- [23] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. [5](#)
- [24] 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, 5, 6](#)
- [25] P. Svoboda, M. Hradis, D. Barina, and P. Zemčík. Compression artifacts removal using convolutional neural networks. *CoRR*, abs/1605.00366, 2016. [1](#)
- [26] R. Timofte, V. DeSmet, and L. VanGool. *A+: Adjusted Anchored Neighborhood Regression for Fast Super-Resolution*, pages 111–126. Springer International Publishing, Cham, 2015. [1](#)
- [27] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. [3](#)
- [28] 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 Appendix. Results of the proposed method: iPhone



Figure 13: Image results of our method for iPhone DPED test images.



Figure 14: Image results of our method for iPhone DPED test images.

7 Appendix. Results of the proposed method: BlackBerry



Figure 15: Image results of our method for BlackBerry DPED test images.



Figure 16: Image results of our method for BlackBerry DPED test images.

8 Appendix. Results of the proposed method: Sony



Figure 17: Image results of our method for Sony DPED test images.



Figure 18: Image results of our method for Sony DPED test images.

9 Appendix. Loss analysis

In this section, we study the contribution of different terms of the proposed perceptual loss function. For this purpose, we consider four different loss combinations: 1) the proposed one [color + content + texture], 2) [content + texture] loss, 3) [MSE + texture] loss and 4) [MSE] loss. For each of these target loss combinations, a CNN was trained on the DPED dataset and tested on its validation subset. The results of this experiment are provided in the Table 3 and visual results are shown in Fig. 19. As one can see, the adversarial network that stands behind the texture loss can cause significant color deviations, and the additional MSE term cannot effectively suppress them since it is not precise in this task (images are not perfectly aligned). Content loss shows better results in this case since it is less sensitive to image mismatches. Adding an extra color term further improves the resulting images, making the colors more saturated and closer to the target. Single MSE demonstrates high PSNR and SSIM values and natural color rendition while causing strong artifacts and slightly degrading image sharpness. Overall our proposed [color + content + texture] loss leads to the best visual results while at the same time achieves both top PSNR and SSIM scores.

Table 3: PSNR/SSIM scores for different loss functions.

Phone	Color + Content + Texture		Content + Texture		MSE + Texture		MSE	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
iPhone	20.95	0.9144	19.54	0.9039	20.53	0.9029	21.23	0.9111
BlackBerry	20.29	0.9247	19.03	0.9192	19.96	0.9163	20.39	0.9223
Sony	22.09	0.9367	21.64	0.9388	21.48	0.9229	21.65	0.9393

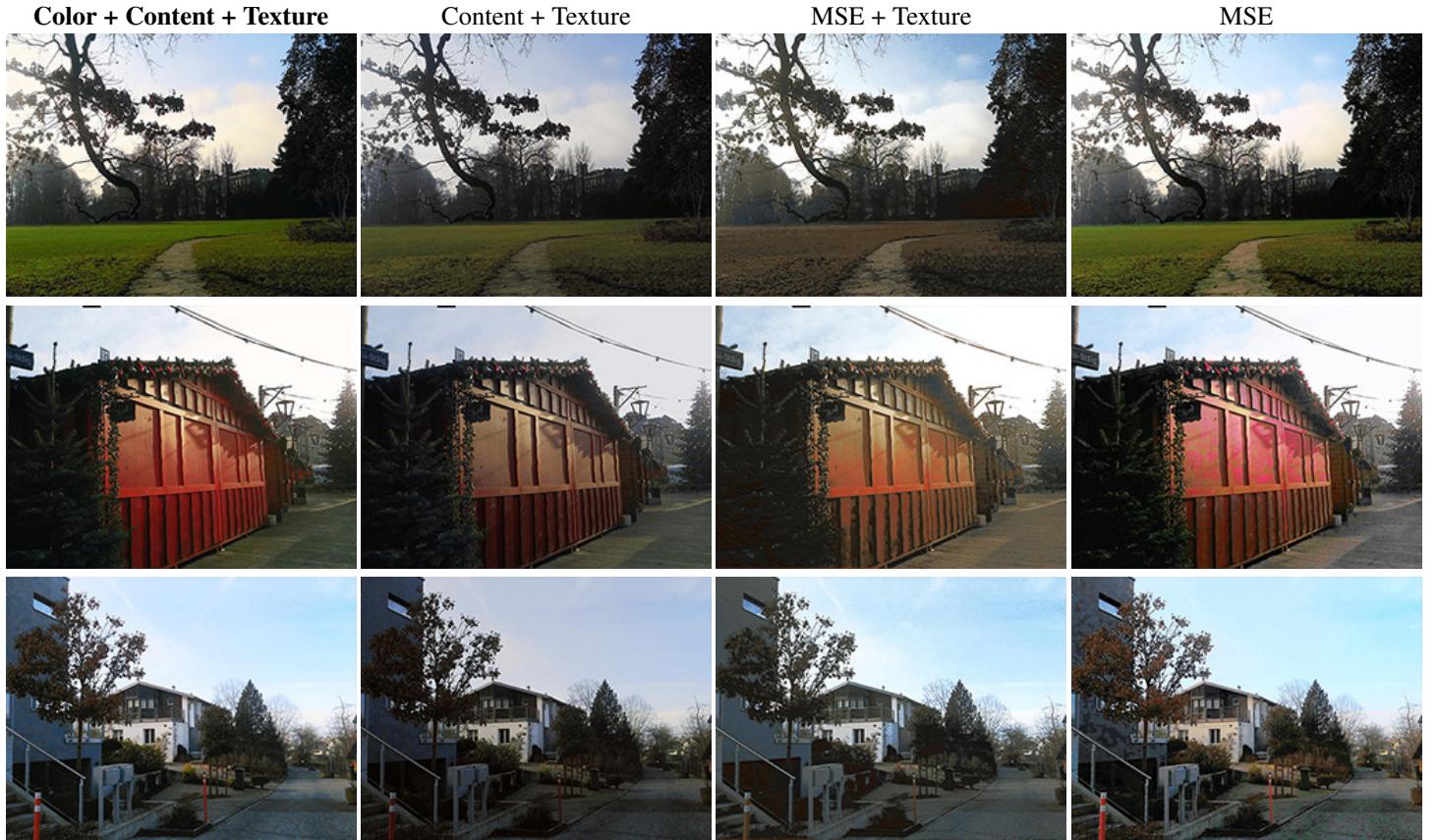


Figure 19: Result images for iPhone camera for 4 different target loss functions.