DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks

1st Mudavath Vishnuvardhan
Electrical Engineering
IIT Bombay
Mumbai, India
200070044@iitb.ac.in

2nd Arpita Gajanan Joshi
Energy Science and Engineering
IIT Bombay
Mumbai, India
20d170004@iitb.ac.in

***Abstract*—Despite the considerable improvement in the quality of built-in smartphone cameras, their inherent limitations, such as small sensor size, compact lenses, and the absence of specific hardware, hinder them from achieving results comparable to DSLR cameras. This study introduces a comprehensive deep learning approach to address this gap, aiming to transform ordinary photos into DSLR-quality images. The proposed method involves training a residual convolutional neural network to enhance both color rendition and image sharpness. Recognizing the inadequacy of the standard mean squared loss for perceptual image quality assessment, 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 adversarially. Additionally, we present DPED, a substantial dataset comprising real photos taken with three different phones and a high-end reflex camera. Both quantitative and qualitative evaluations demonstrate that the improved image quality is comparable to that of DSLR-captured photos, and the methodology is applicable to various digital cameras.**

***Index Terms*—Deep Learning for Image Enhancement, Adversarial Learning, Color Rendition Enhancement.**

I. Introduction

In recent years, there has been a notable advancement in the quality of compact camera sensors, elevating the field of mobile photography to unprecedented heights. Even budget-friendly devices can now capture reasonably impressive photos under favorable lighting conditions, thanks to the sophisticated software and hardware tools for post-processing. Despite these strides, mobile devices still lag behind DSLR counterparts in terms of artistic quality. The superior photo resolution, color rendition, and reduced noise achieved by larger sensors and high-aperture optics, coupled with additional sensors for precise parameter adjustments, contribute to the unattainability of DSLR camera quality for compact mobile devices.

While various tools for automatic image enhancement exist, many are limited to adjusting global parameters like contrast or brightness. These tools often neglect texture quality and image semantics, relying on predefined rules that may not account for the unique characteristics of each device. Consequently, manual image correction through specialized retouching software remains the dominant approach to photo post-processing.

A. Related work

Automatic Image Quality Enhancement Challenges: The realm of computer vision has made significant strides in solving various sub-tasks through deep learning techniques; however, the holistic problem of automatic image quality enhancement remains largely unaddressed. Many successful applications in this domain focus on image-to-image translation, typically aiming to eliminate artificially added artifacts from original images.

Sub-Tasks and Solutions in Computer Vision: Several sub-tasks within computer vision have seen success with deep learning techniques. For instance, image super-resolution involves restoring an original image from a down-scaled version using convolutional neural network (CNN) architectures and mean squared error (MSE) loss. Image deblurring and dehazing target the removal of added haze or blur, utilizing MSE as a loss function and employing CNN architectures with varying convolutional layers. Image denoising and sparse inpainting, aiming to eliminate noise and artifacts, use different CNN architectures and loss functions.

Image Colorization Challenges and Solutions: Image colorization, which focuses on restoring colors removed from the original image, traditionally relied on predicting new pixel values based on hand-crafted features. More recent approaches have leveraged generative adversarial networks or complex CNN architectures with multinomial cross-entropy loss for improved performance.

Image Adjustment Approaches: Addressing image color, contrast, and exposure adjustments, some works proposed algorithms for automatic exposure correction using hand-designed features and predefined rules. Another approach involves a general algorithm that utilizes local descriptions of image pixels to reproduce various photographic styles. In a different perspective, an algorithm retrieves images with similar content from a database and applies their styles to the target picture. Notably, these adjustments are implicitly incorporated into an end-to-end transformation learning approach by design.

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 the limitations, section 5 concludes the paper. Finally, the appendix shows the results of the proposed method.

Fig. 1. Example of an enhanced image.

II. DSLR Photo Enhancement Dataset

To address the challenge of translating images from lower quality captured by smartphone cameras to higher quality resembling those taken by professional DSLR cameras, we present a substantial real-world dataset named the "DSLR Photo Enhancement Dataset" (DPED). This dataset serves the broader purpose of general photo quality enhancement. DPED comprises photos captured concurrently in real-world scenarios by three smartphones and one DSLR camera.

To ensure simultaneous photo capture by all cameras, the devices were securely mounted on a tripod and remotely activated through a wireless control system. Over a span of 3 weeks, a total of more than 22,000 photos were collected, encompassing 4,549 photos from a Sony smartphone, 5,727 from an iPhone, and 6,015 each from Canon and BlackBerry cameras. The photo sessions occurred during daylight hours in diverse locations, encompassing various lighting and weather conditions. All photos were captured in automatic mode, maintaining default settings for all cameras throughout the entire data collection process.

Matching algorithm: The images captured simultaneously are not perfectly aligned due to varying viewing angles and positions of the cameras. To address this misalignment, we employed additional non-linear transformations to generate fixed-resolution images that serve as input for our network. The algorithm proceeds as follows For each pair of images from the phone and DSLR, we compute and match SIFT keypoints across them. Utilizing RANSAC, we estimate a homography and then crop both images to their intersection part. Subsequently, we downscale the DSLR image crop to match the size of the phone crop.

Training a Convolutional Neural Network (CNN) directly on aligned high-resolution images is impractical. Therefore, we extracted patches of size 100×100 pixels from these photos. Preliminary experiments indicated that larger patch sizes did not significantly improve performance but demanded substantially more computational resources. Patches were extracted using a non-overlapping sliding window, moving in parallel along both images from each phone-DSLR pair. The window's position on the phone image was further adjusted by shifts and rotations based on cross-correlation metrics. To ensure accuracy, only patches with cross-correlation greater than 0.9 were included in the dataset. Approximately 100 original images were reserved for testing, while the remaining photos were utilized for training and validation. This process resulted in 139,000, 160,000, and 162,000 training patches and 2,400 to 4,300 test patches for BlackBerry-Canon, iPhone-Canon, and Sony-Canon pairs, respectively. It's essential to note that both training and test patches are precisely matched, with potential shifts not exceeding 5 pixels. Hereafter, we consider these patches of size 3×100×100 as the input data for our CNNs.

III. Method

Given a low-quality photo Is (source image), the goal of the considered enhancement task is to reproduce the image It (target image) taken by a DSLR camera. A deep residual CNN F_W parameterized by weights **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:

$$W^* = argmin \frac{1}{N} \sum_{n=1}^{N} L(F_W(I_s^j), I_t^j), \tag{1}$$

where 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

A. Loss function

The main difficulty of the image enhancement task is that input and target photos cannot be matched densely (i.e., pixelto-pixel): different optics and sensors cause specific local nonlinear 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.

1) color loss: To measure the color difference between the enhanced and target images, we propose applying a Gaussian blur 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:

$$L_{color}(X, Y) = ||X_b - Y_b||_2^2 \tag{2}$$

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

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

and the 2D Gaussian blur operator is given by

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

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 (6 2 pixels). For higher shifts (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.

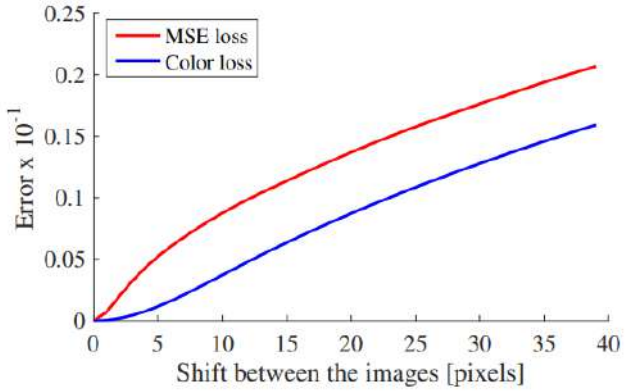

Fig. 2. MSE vs Color loss.

2) Texture loss: Instead of using a pre-defined loss function, we build upon generative adversarial networks (GANs) 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:

$$L_{texture} = -\sum_i logD(F_W(I_s), I_t), \quad (5)$$

where F_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) Content loss: Inspired by, 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:

$$L_{content} = \frac{1}{C_j H_j W_j} ||\psi_j(F_W(I_s)) - \psi_j(I_t)||, \quad (6)$$

where C_j , H_j and W_j denotes the number, height and width of the feature maps, and $F_W(I_s)$ the enhanced image.

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

$$L_{tv} = \frac{1}{CHW} ||\nabla_x(F_W(I_s)) + \nabla_y(F_W(I_s))||, \quad (7)$$

where C, H and W are the dimensions of the generated image $F_W(I_s)$. 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.

5) Total loss: Our final loss is defined as a weighted sum of previous losses with the following coefficient:

$$L_{total} = L_{content} + 0.4 * L_{texture} + 0.1 * L_{color} + 400 * L_{tv}, \quad (8)$$

where the content loss is based on the features produced by the $relu_{54}$ layer of the VGG-19 network. The coefficients were chosen based on preliminary experiments on the DPED training data.

B. Generator and Discriminator CNN

The comprehensive architecture of the proposed Convolutional Neural Networks (CNNs) is depicted in Figure 7. The image transformation network is entirely convolutional, initiating with a 9×9 layer, succeeded by four residual blocks. Each residual block comprises two 3×3 layers interspersed with batch-normalization layers. Following the residual blocks, two additional layers with 3×3 kernels and one with 9×9 kernels are employed. All layers within the transformation network have 64 channels and are succeeded by a Rectified Linear Unit (ReLU) activation function, except for the final layer, which applies a scaled hyperbolic tangent (tanh) to the outputs.

The discriminator CNN is composed of five convolutional layers, each succeeded by a LeakyReLU nonlinearity and batch normalization. The first, second, and fifth convolutional layers have striding with step sizes of 4, 2, and 2, respectively. The outputs of the last fully-connected layer, which contains 1024 neurons, undergo a sigmoidal activation function. This function produces a probability indicating whether the input image was captured by the target DSLR camera.

C. Training details

The network was trained on a Google Colab T4 GPU for 1K iterations using a batch size of 32. The parameters of the network were optimized using Adam modification of stochastic gradient descent with a learning rate of 5e-4. The whole pipeline and experimental setup was identical for all cameras.

In table 1 and 2, we have shown the results of our model training and testing on Iphone images

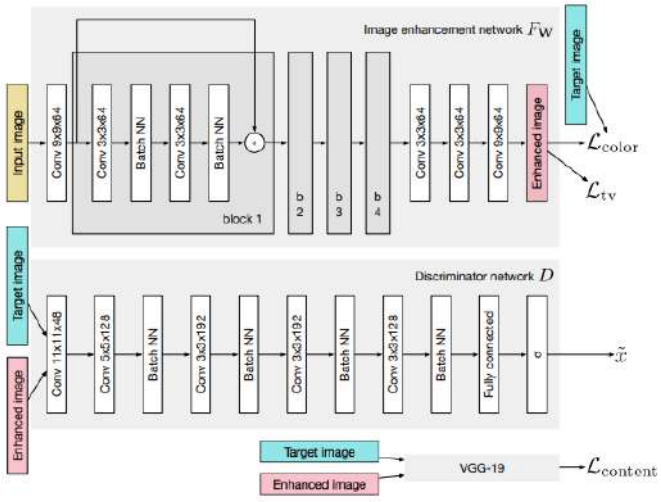

Fig. 3. The overall architecture of the proposed system .

IV. Limitations:

The fully-automated nature of the proposed image enhancement process leads to some inherent limitations. Common issues observed in the enhanced images include color inaccuracies, as seen in the depiction of ground and mountains in the first image of Figure 12, and excessively high contrast levels, noticeable in the second image of the same figure. While these artifacts can sometimes create visually acceptable outcomes, they can also lead to unnatural content alterations.

Another significant challenge is the amplification of noise, particularly in high-frequency components. This issue is a byproduct of the Generative Adversarial Networks (GANs) used, which, while effective in restoring high-frequency details, also tend to accentuate noise. This problem is more pronounced in images of lower quality, especially those taken with iPhones.

V. Conclusion

In our study, we developed a photo enhancement technique aimed at elevating the quality of images captured by standard smartphone cameras to the level of high-quality DSLR cameras. This technique is built on an end-to-end deep learning framework, incorporating a unique perceptual error function that amalgamates content, color, and texture losses. For the training and evaluation of our method, we used DPED, a comprehensive dataset comprised of authentic photographs taken from three distinct smartphone models and a high-end reflex camera. Additionally, we devised an effective approach to calibrate these images, making them conducive for image-to-image learning applications. Both quantitative and qualitative evaluations of our method show that it significantly enhances image quality, making it on par with photos taken by DSLR cameras, and is versatile enough to be applied to cameras of varying quality levels.

Fig. 4. Typical artifacts generated by the proposed method (2nd row) compared with original iPhone images (1st row) .

TABLE I
Training Progress of Photo Enhancement Model - Part 1

Step	Disc. Accuracy (Train)	Disc. Accuracy (Test)
0, iPhone	0.003594	0.4979
200, iPhone	0.5142	0.492
400, iPhone	0.5052	0.494
600, iPhone	0.512	0.497
800, iPhone	0.4963	0.4956

TABLE II
Training Progress of Photo Enhancement Model - Part 2

Step	Gen. Loss (Train)	Gen. Loss (Test)	Metrics (PSNR, MS-SSIM)
0, iPhone	3.355	784	10.76, 0.4239
200, iPhone	278.4	245.2	17.42, 0.8609
400, iPhone	213.3	206.1	18.24, 0.8739
600, iPhone	187.4	177.8	18.92, 0.8876
800, iPhone	177.2	177.7	18.8, 0.8903

References

[1] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey, and L. Van Gool, "DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks," arXiv preprint arXiv:1704.02470, April 2017, revised September 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1704.02470

[2] A. Ignatov, "DPED: Dataset for improving photo enhancement," GitHub repository, [Online]. Available: https://github.com/aiff22/DPED

[3] P. Hofmann, "DPED Dataset," Kaggle, [Online]. Available: https://www.kaggle.com/datasets/philiphofmann/dped-dataset

APPENDIX

A. Results of the Proposed Method

Fig. 5. Image 1

Fig. 6. Image 2

Fig. 7. Image 3

Fig. 8. Image 4

Fig. 9. Image 5