DSLR Photos on Mobile Devices with Deep Convolutional Network

PRESENTED BY:
Ashish Verma (300059114)
Gurpreet Singh (8908872)
Karan Mahajan (300082496)

This presentation is submitted to Professor Jochen Lang in partial fulfillment of the requirements for CSI 5139Q

Paper…

Andrey Ignatov
Nikolay Kobyshev
Kenneth Vanhoey
Radu Timofte
Luc Van Gool

2017 IEEE International Conference on Computer Vision (ICCV)

uOttawa

Presentation Overview

- Introduction
- Dataset
- Method
- Experiments
- Conclusion and Future Work

uOttawa

Introduction

- Mobile devices fall behind the DSLR cameras in terms of artistic image quality
- Physical limitations of the smartphone impeding to achieve the DSLR quality images
  - small sensor size
  - compact lenses
  - Lack of specific hardware
- People who can’t afford DSLR cameras can have the luxury of having DSLR quality images in the mobile device

Related Work

- Image super resolution
  - To restore the original image from its downscaled version
  - VGG based loss function and adversarial networks
- Image deblurring
  - To remove the artificially added blur from the images.
  - Proposed CNN architecture consist of 3-15 convolutional layers
- Image denoising
  - Helps in removal of noise and artifacts from the pictures.
  - 8 layer residual CNN using standard mean square error
- Image colorization
  - to recover colors which were removed from the original image.
  - generative adversarial networks

Main Contributions

- Learning Mapping function between photos from mobile devices and a DSLR camera
- Using multi-term loss function composed of color, texture and content terms
- Efficient image quality estimation
- Experiments measuring the quality of the enhanced photos over their originals and the DSLR counterparts

Dataset - DPED

Fig 1: The rig with the four DPED cameras

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

Fig 2: DPED Camera characteristics

Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey and Luc Van Gool. "DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks", in IEEE International Conference on Computer Vision (ICCV), 2017

Dataset - DPED

- DSLR Photo Enhancement Dataset
- Large-scale real world dataset
- Photos taken in the wild synchronously by three smartphones and one DSLR camera.
- Devices were mounted on a tripod and activated remotely by a wireless control system.
- 22K photos were collected during 3 weeks

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

Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey and Luc Van Gool. "DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks", in IEEE International Conference on Computer Vision (ICCV), 2017

Matching Algorithm

- Synchronously captured images are not perfectly aligned.
- The cameras have different viewing angles.
- Additional Non-linear transformations
- SIFT key points are computed and matched.
- Downscale the DSLR image crop to the size of the phone crop.

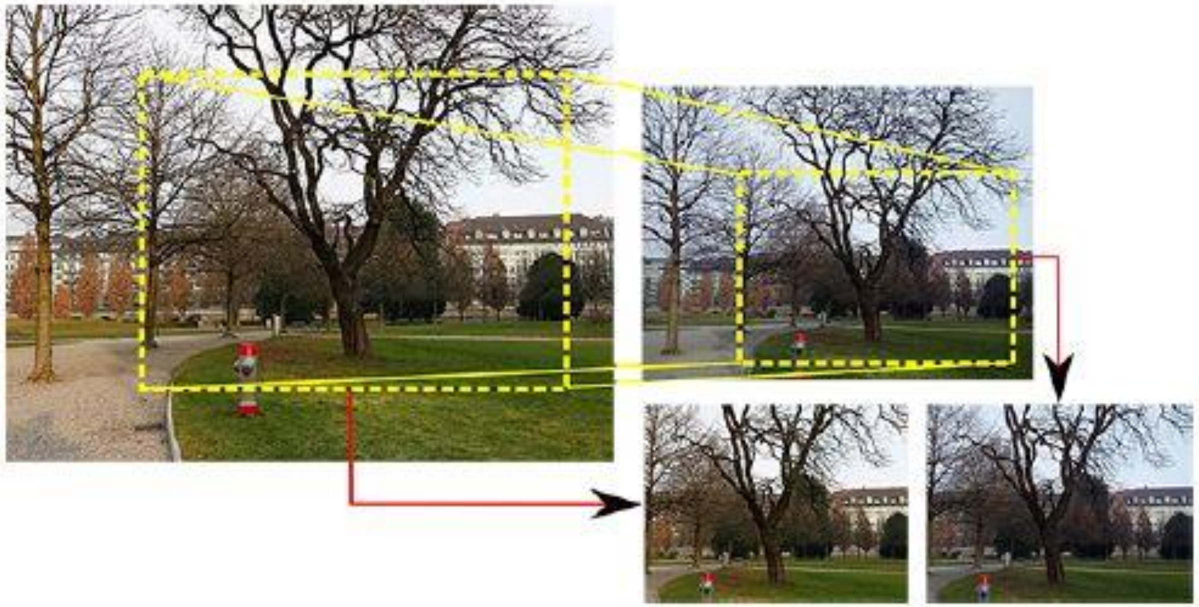


Fig 4: an overlapping region is determined by SIFT descriptor matching.

Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey and Luc Van Gool. "DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks", in IEEE International Conference on Computer Vision (ICCV), 2017

Matching Algorithm

- Training CNN on the aligned high-resolution images is infeasible
- Patches of size 100x100px extracted from the photos as the larger patch sizes do not lead to better performance and requires high computational resources.
- Patches with cross correlation greater than 0.9 were included in the dataset
- 100 original images were reserved for testing
- 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.

Method

- I_s be the source image (image from smartphone)
- I_t be the target image (image from DSLR)
- CNN F_W parametrized by weights W
- Learn translation function below

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

- Where N = number of image pairs, $\mathcal{L}$ = multi-term loss (combination of losses)

Why Multi-term loss?

- The source and target image can't be densely matched
- Reason : different devices use different optics and sensor, which leads to distortions and aberration
- Leads to non constant shift of pixels, even after perfect alignment

- Solution : Perceptual image quality can be decomposed as
  - Color quality
  - Texture quality
  - Content quality

$\mathcal{L}$ Multi-term loss = Color Loss

- To measure color difference between enhanced(source image with some modifications) and target images
- Apply Gaussian blur and compute Euclidean distance for obtained result

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

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

Xb and Yb are the blurred images of X and Y

2D Gaussian Blur

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

A = 0.053, µx,y = 0, and σx,y = 3

Color Loss : Why use Gaussian Blur?

- It (Blurring) removes high frequencies from image
- Thus making color comparison easy

- Evaluate difference in brightness, contrast and major colors between images
- This doesn't consider texture and content comparison

- Here σ (visual inspection) is set to smallest value so that texture and content are dropped.
- This loss is invariant to small distortions

2D Gaussian Blur

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

$A = 0.053$, $\mu_{x,y} = 0$, and $\sigma_{x,y} = 3$

$\mathcal{L}$ Multi-term loss = Texture Loss

- A separate network – generative adversarial network (GAN) - Discriminator
- Used on grayscale images to target texture processing only
- To learn metric for texture quality
- To predict if input image is real or not (Discriminates images)
- Trained to minimize cross entropy loss
- Trained separately and later on jointly with generator (F_W)
- Like color loss this is also shift invariant

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

Texture Loss : Why use GANs?

- Discriminative algorithms map features to labels. They are concerned solely with that correlation
- Used to determine Boolean decision like real or fake
- Generator creates enhanced images and discriminator decides them for being real or fake
- The discriminator is in a feedback loop with the ground truth of the images
- The generator is in a feedback loop with the discriminator.

Must read
https://skymind.ai/wiki/generative-adversarial-network-gan

$\mathcal{L}$ Multi-term loss = Content Loss

- Uses pre-trained VGG-19 network
- Doesn’t measure per pixel difference
- Used to preserve similar feature representation(semantics), content and perceptual quality
- Again its Euclidean distance between feature representation of enhanced and target images

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

- Where ψj is feature map and Cj, Hj and Wj denotes the number, height and width of the feature maps, and $F_W(Is)$ the enhanced image

$\mathcal{L}$ Multi-term loss = Total Variation Loss

- To enforce spatial smoothness of the produced images
- To remove noise observed in enhanced images

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

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

$\mathcal{L}$ Total Loss

- Final loss is weighted sum of previous losses

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

- The coefficients were chosen based on preliminary experiments on DPED training data

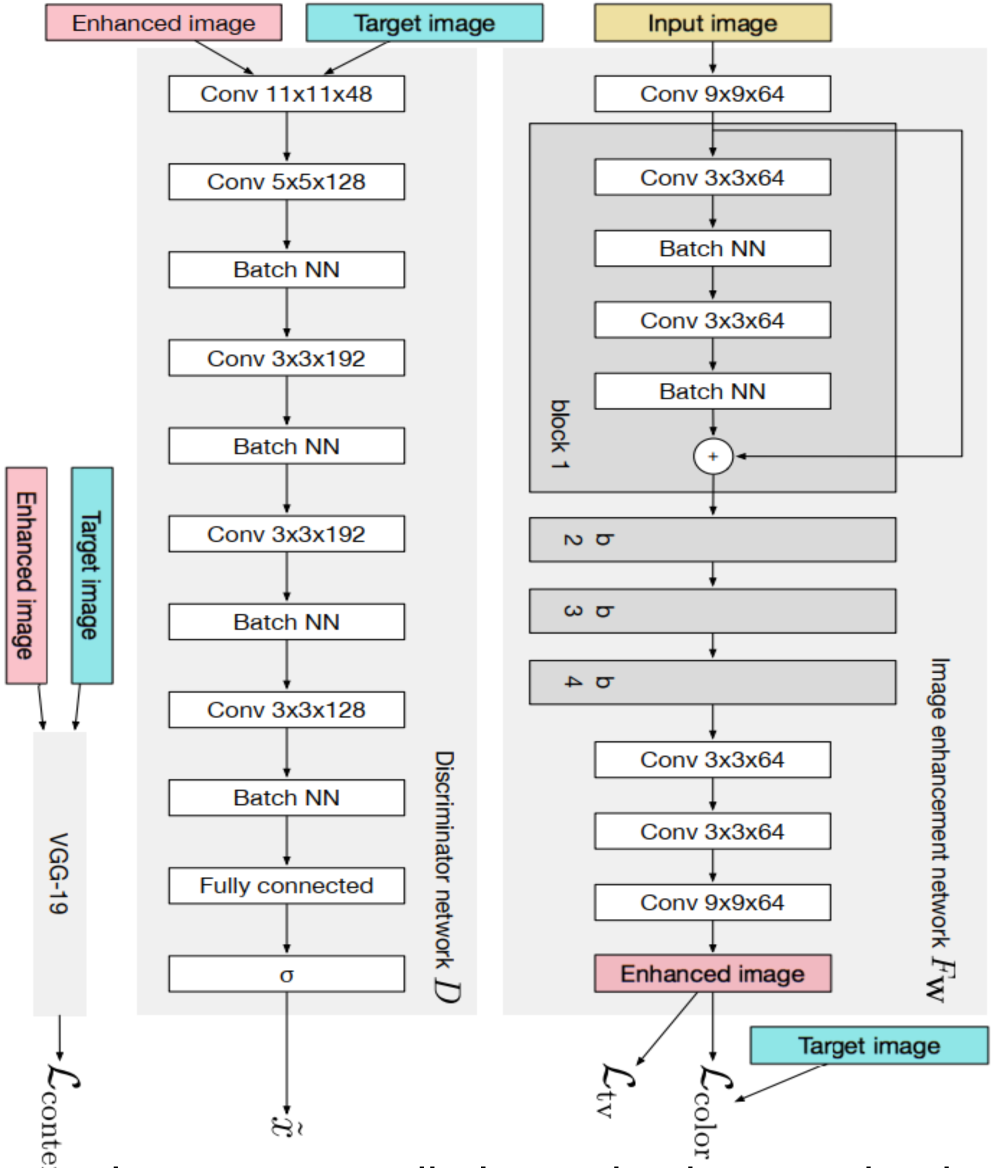
Overall Architecture : Transformation CNN

- Transformation network – Fully Convolutional
- 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.
- Two additional layers with kernels of size 3×3 and one with 9×9 kernels after the residual blocks.
- All layers have 64 channels *ReLU* activation function, except for the last one, where a scaled *tanh* is applied to the outputs.

Overall Architecture : Discriminative CNN(GAN)

- The first, second and fifth convolutional layers are strided with a step size of 4, 2 and 2.
- A Sigmoidal activation function is applied to the outputs of the last fully-connected layer containing 1024 neurons
- Produces a probability that input image was taken by the target DSLR camera

❖ Network parameters were optimized using *Adam* modification of stochastic gradient descent



Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey and Luc Van Gool. "DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks", in IEEE International Conference on Computer Vision (ICCV), 2017

Experiments

Some methods that we compared to are:

- Apple Phote Enhancer (APE) – It is a commercial product for improving visual results.
- Dong et al –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 –This method is based on deep residual network that is trained to minimize a VGG- based loss function

Fig 5: From left to right, top to bottom: original iPhone photo and the same image after applying, respectively: APE, Dong et al., Johnson et al., our generator network, and the corresponding DSLR image.

Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey and Luc Van Gool. "DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks", in IEEE International Conference on Computer Vision (ICCV), 2017

Quantitative Evaluation

Table 1: 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**

Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey and Luc Van Gool. "DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks", in IEEE International Conference on Computer Vision (ICCV), 2017

User Study

- Our goal is to produce DSLR-quality images for end user of smartphone cameras.
- 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.). Time limit was not an issue and users were allowed to zoom in/out at will.

User Study

Fig 6: Iphone picture before and after applying deep CNN

Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey and Luc Van Gool. "DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks", in IEEE International Conference on Computer Vision (ICCV), 2017

Results

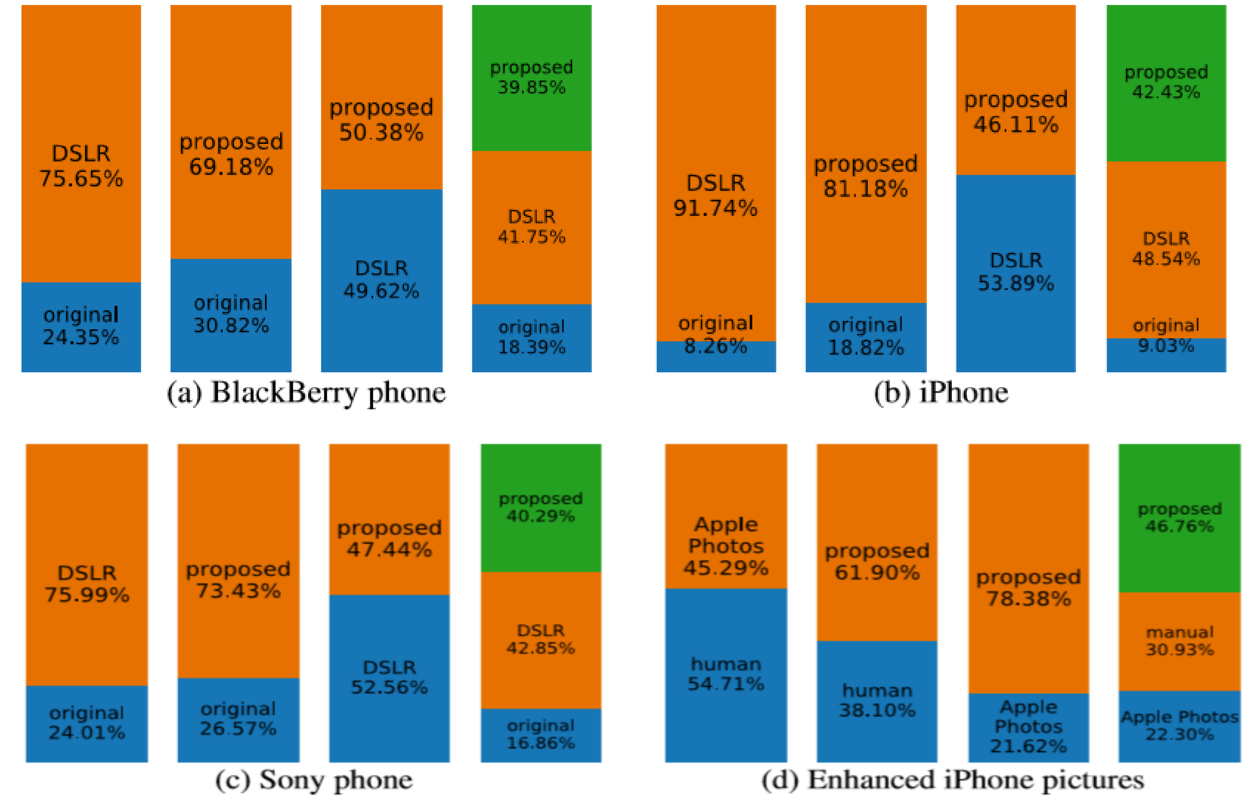


Fig 7 : results of pairwise comparisons. In every subfigure, the first three bars show the result of the pairwise experiments, while the last bar shows the distribution of the aggregated scores.

uOttawa

Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey and Luc Van Gool. "DSLR-Quality Photos on Mobile Devices with Deep Convolutional Networks", in IEEE International Conference on Computer Vision (ICCV), 2017

Conclusion

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

Limitations

Since the proposed enhancement process is fully-automated, some flaws are inevitable. Two typical artifacts that can appear on the processed images are

- Color deviations - Although they often cause rather plausible visual effects, in some situations this can lead to content changes that may look artificial.
- Noise amplification – due to the nature of GANs, they can effectively restore high frequency-components. However, high-frequency noise is emphasized too.

рахмат
danke
謝謝
ngiyabonga
Баярлалаа
spas
mersi
kia ora
barka
welalin
tack
teşekkür ederim
mahalo
tapadh leat
спасибо
faafetai lava
vinaka
blagodaram
dank je
misaotra
matondo
paldies
grazzi
спасибі
kiitos
dankie
thank you
gracias
хвала
nanni
nandri
dhanyavad
mauruuru
asante
manana
bedankt
bayarlalaa
gracie
hvala
köszönöm
aciu
dankon
akun
obrigada
tenki
murakoze
enkosi
djiere dieuf
tau
mochchakkeram
дякую
mamnun
chokrane
dziękuję
chnorakaloutioun
gratias ago
gracies
sulpay
go raibh maith agat
sobodi
dėkuji
sagolun
sukriya
kop khun krap
taiku
arigatô
takk
dakujem
trugarez
obrigado
mèsi
didi madloba
ありがとう
tanemirt
grazie
najis tuke
terima kasih
rahmet
diolch
dhanyavadagalu
shukriya
merce
мерси
kam sah hamnida
rahmat
তোমাকে ধন্যবাদ
감사합니다
xiexie
ευχαριστώ
merci

uOttawa