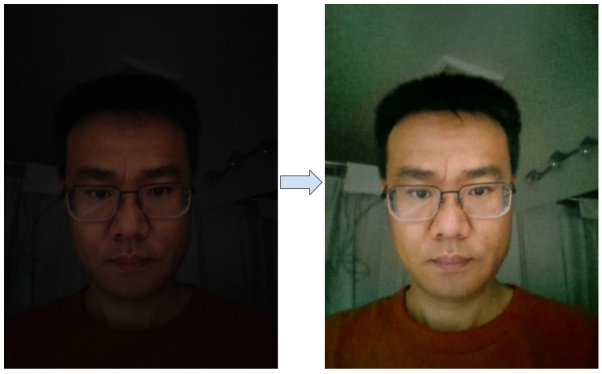
An Efficient Hybrid Model for Low-light Image Enhancement in Mobile Devices

Zhicheng Fu[1], Miao Song[1], Chao Ma[1], Joe Nasti[1], vivek tyagi[1], Grant Lloyd[1], Wei Tang[2]
Lenovo Research[1], University of Illinois Chicago[2]
{zcfu,MSONG2,chaoma,joenasti,vivekt,grantlloyd}@motorola.com, tangw@uic.edu

Abstract

*With the help of continuous optimizations in hardware and software, smartphones can now capture vivid, detailed macro pictures as well as high-resolution videos. However, taking photos/videos in a low-light environment with smartphones would still result in underexposed and bad-quality photos/videos due to their physical limitations — small sensor size, compact lenses, and the lack of specific hardware and software. A variety of low-light enhancement techniques have been proposed, but their effectiveness is limited by their high complexity and the limited computational resources of smartphones. In this paper, we present an efficient hybrid solution, named as LLNet, to generate a high-resolution enhanced image given the corresponding high-resolution low-light image on mobile devices. LLNet consists of two main parts: 1) a lightweight convolutional neural network for features restoration that takes a low-resolution low-light image scaled down from the high-resolution input and predicts an enhanced low-resolution output; 2) a non-trainable transformation estimation model that approximates a linear transformation between the low-resolution input and predicted low-resolution output. By applying the estimated transformation on a high-resolution low-light image, the corresponding enhanced image can be predicted efficiently. To support the development of this learning-based solution, we introduce a dataset of normal-exposure low-light images, with corresponding long-exposure reference images, and all the images were captured by smartphones under real-world low-light scenes. Experiments demonstrate that LLNet can provide a real-time (around 32ms) smartphone preview (1440*1080 resolution) with outstanding image enhancement under low-light environments with affordable resources consumption. One real viewfinder video demo is attached as supplementary material to indicate the practicality of LLNet on real smartphones.*



(a) The selfie were captured around 0.7 lux.

Figure 1. A 1440*1080 selfie captured under low-light by smartphones is significantly enhanced in real-time (around 32ms) by our proposed method-LLNet. The left image is the low-light image while right one is processed by LLNet

1. Introduction

With the rapid improvements in camera sensors quality, smartphones have given point-and-shoot cameras and digital single-lens reflex cameras (DSLRs) a run for their money. Although smartphones do hold their own against DSLRs in most aspects, the one area where they fared badly is low-light photography. Due to limited computational resources and the requirement of the fastest possible processing time, image processing algorithms are under significant performance pressure to produce high-quality media in low-light environments. This challenge in low light for mobile devices is well known in the computational photography community but remains open. By now, one of the most mature solutions in the industry is based on multiple-exposure frames fusion, which requires end-users to hold 2s-4s to capture multiple exposure frames with the same contents later to be blended. But it is hard to hold still long enough to take a good picture in dim light, and such multiple-exposure frames fusion solutions cannot be applied to real-time tasks, such as preview and video in smartphones.

Although low-light image enhancement algorithms have been the focus of a great deal of research, most sophisti-

cated algorithms are too computationally expensive to be integrated into mobile devices. To make the "expensive" quantitative, we take an Android phone with specific hardware to evaluate the computational costs of low-light enhancement algorithms on smartphones. The specifications of the Android phone are described in Section 2. According to the evaluation for low-light images with the Android phone, conventional algorithms [13, 26, 28] hold fast running time, but are limited by conditions of usage and not achieving commercial-level image quality. From the perspective of image quality, many deep learning-based architectures [3,4,9,19,38] have proved their outstanding capacity of enhancing low-light images. However, such work incurs a heavy computational cost that scales linearly with the size of the input image, usually because of the large number of stacked convolutions and non-linearities that must be evaluated at full resolution. The problem will limit their practicability in smartphones. Therefore, developing an approach to utilize the outstanding performance of CNN-based architectures while minimizing the cost of computational resources, has been highly demanded in the smartphone industry.

In this paper, we present a hybrid model, named as LLNet, which is capable of learning a rich variety of low-light image enhancements and can be applied on high-resolution (1440*1080) inputs in real-time. Figure 1 presents the promising outputs by LLNet. In particular, we achieve this through three main steps: 1) after scaling high-resolution low-light images down to low-resolution images, we train a lightweight convolutional neural network to perform enhancement on the low-resolution low-light images for better image quality and fast running time; 2) with the low-resolution input and low-resolution predicted output, we design an approach to approximate the linear transformation between them; and 3) by applying a non-trainable estimated transformation model on high-resolution low-light images, the desired high-resolution enhanced image can be predicted in real-time. Taken together, these three steps allow us to perform the bulk of our processing at a low resolution with a CNN-based architecture for better image quality while saving substantial compute cost, yet using the low-resolution output to approximate a high-resolution equivalent in real-time.

In addition, to support the development of our learning-based LLNet, we have collected a new dataset of low-light images captured by smartphones with various ambient scenes. Each low-light image pair has the normal exposure image as the input and the corresponding long-exposure image as the ground truth. LLNet delivers promising results on the new dataset: low-light images are improved with better noise reduction and correct color transformation.

According to the experiments, the hybrid architecture-LLNet demonstrates its capability of delivering good quality results that are comparable to/or better than previous work, and the outstanding performance of real-time processing on mobile devices. In particular, video demos in the supplementary material indicate that LLnet can provide real-time preview enhancement around 30 Hz under low-light environments on the Android phone with specific configurations described in Section 2. The main contribution of our work can be summarized in the following perspectives:

(a) Input-0.8 lux (b) CLAHE [28] (c) Retinex [26] (d) LIME [13]

(e) SID [3] (f) ZeroDCE [12] (g) HDRNet [9] (h) Our LLNet

Figure 2. A visual comparison between state-of-the-art algorithms and LLNet on a single low-light image captured with smartphones for ablation study.

- We propose an efficient hybrid model (LLNet) with the combination of a lite convolutional neural network and a non-trainable linear transformation estimation model, to enhance low-light images in mobile devices.
- A new dataset of 3,000 low-light images and the corresponding ground truths is presented. All the images are captured by smartphones under real-world low-light scenes.
- We perform evaluations on LLNet using the new dataset as well as ablation study by end-users dogfooding and demonstrate the superiority of our model qualitatively and quantitatively.

2. Related Work

A variety of sophisticated algorithms have been proposed to enhance the overall quality of low-light images. The most intuitive and simplest way to restore the visibility of dark regions is by directly amplifying the low-light image, such as gamma correction [11, 29] which increases the brightness of dark regions while compressing bright pixels. However, this type of operation could distort saturation and contrast reduction. Histogram equalization strategies [6, 17, 22, 28] can avoid the above problem by forcing

the output image to fall in some specific range. However, in nature, they focus on contrast enhancement instead of exploiting real illumination causes, having the risk of over- and under-enhancement. To have better improvement of overall image quality, more advanced methods have been developed with more complex analysis and processing operations, such as the inverse dark channel prior [7, 24], the wavelet transform [39], the Retinex model [26], and illumination map estimation [13]. Although these methods have indicated their promising results in some specific cases, the low-light images suffering from severe noise and color distortion are still beyond the operating conditions of such methods. Figure 2 demonstrates the comparison among these algorithms.

Different from traditional image processing methods with a specific focus with a constrained input scope, Convolutional Neural Networks (CNN) based methods started to demonstrate their superiority in image enhancement tasks with the concept of end-to-end learning. Such methods mostly benefit from using a large volume of images in the training process, to learn the corresponding high-quality features, and there is no longer a need of defining the features and do feature engineering as traditional image processing methods do. For general image enhancement, Yan et al. [35] proposed the innovative deep-learning-based method for photo adjustment. Chen et al. [4] developed a fully convolutional network to approximate existing filters for image enhancement. More recently, a lot of deep convolutional networks have achieved significant progress on low-light image processing such as GLADNet [31], RetinexNet [33], KinD [36], UPED [30], pixel2pixel [19], CycleGAN [38], DPE [5], DPED [18], SID [3], EnlightnGAN [20], Zero-DCE [12], and HDRNet [9]. Among these CNN-based solutions, the most related one is HDRNet [9] which also leverages the hybrid methodology with a small CNN network and a c++ based bilateral grid model for achieving the real-time processing enhancement. However, LLNet and HDRNet differ in two aspects:: (1) different domain areas: HDRNet mainly focuses on high dynamic range (HDR) imaging while LLnet is designed to improve low-light images. Figure 2g indicates that HDRNet could produce washout effects, lower contrast, and less color restoration than LLNet. (2) different processing procedures: HDRNet uses CNN-based networks to learn the transformation mapping between inputs and ground truth, and LLNet takes a CNN-based network to generate predicted images and estimate the transformation model between inputs and predicted images.

All these sophisticated algorithms have proved their outstanding performance on image enhancement under required conditions. However, due to the demands of large computational costs, methods are often too expensive to be integrated into mobile devices. To quantify computational costs in mobile devices, we take an Android phone with specific hardware to evaluate the performance of algorithms against 1440*1080 low-light images. The specifications of the Android phone are described as *Qualcomm SM8450 chipset with Adreno 660 GPU. 8G memory, Camera Sensor: OV32B40, Resolution: 32MP, Aperture: f/2.25, Pixel Size: 0.7um, Sensor size: 1/3.15", Focus: Fixed, FOV (diag): 73.24°*.

With the Android phone, we have evaluated specific CNN-based algorithms mentioned above. From the perspective of image quality, compared with traditional algorithms, CNN-based solutions indicate their better performance on low-light image improvement, Figure 2 shows the evaluation example between CNN-based algorithms and conventional solutions. However, such end-to-end learning-based algorithms are too expensive to be integrated into mobile devices with limited computational resources. For example, SID [3] would cost about 2.4 seconds and consume 1.8G memory to finish the inference on a 1440×1080 image. Inspired by the existing low-light image enhancement research, LLNet strikes an appropriate balance between image quality and computational costs by the hybrid combination of a lightweight convolutional neural network and a transformation estimation model for processing high-resolution low-light images.

3. LLNet Dataset

Although there are many existing datasets [3, 14, 18, 33] for evaluating the performance of methods targeting low-light images, these images cannot accurately reflect the real semantic information captured by smartphone cameras under real low-light environments. For the Google HDR+ dataset [14], most images were captured during the day. The images of LOL [33] were taken with multiple-exposure levels during the day to simulate low/normal light images. But these images cannot represent accurate light distribution, noise distribution, and color distortion of real low-light images captured by smartphone cameras.

In this session, we present a new dataset for training and bench-marking single-image processing of low-light images in JPEG formats with 1440*1080 resolution. The dataset of LLNet contains 3000 default exposure images captured by smartphone with default camera sensor setting, each with a corresponding long-exposure reference image as the ground truth. Note that the exposure for the low-light images was set as 1/10 seconds, and the corresponding reference (ground truth) images were captured with 300 times longer exposure as 30 seconds. Since exposure times for the reference images are necessarily long, all the scenes in the dataset are static. The blurred images caused by shaking have been deleted by manual filtering. Since we focus on researching single low-light image enhancement, the required images should be captured by a single camera shot without

the fusion of multiple images captured by multiple cameras. Hence, all images were captured with the front-facing camera of smartphones and were scaled to be 1440*1080 for training and testing.

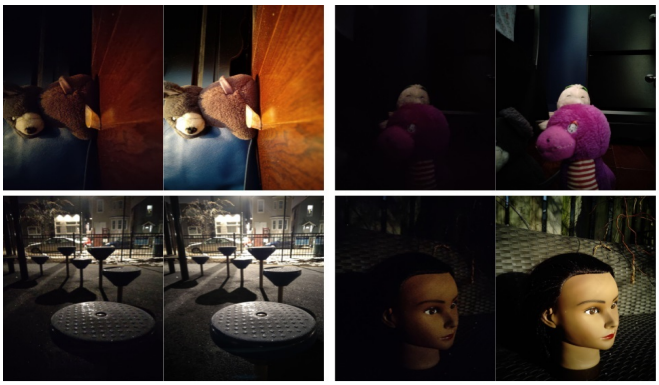


Figure 3. Example images in the LLNet dataset. The left one is the default-exposure low-light image, and its corresponding reference (ground truth) image is shown in right. The illuminance at the camera is generally between 0.5 and 5 lux for both indoor and outdoor.

The dataset covers both indoor and outdoor scenes. The outdoor images were generally captured at night, early morning, under moonlight or street lighting. For the indoor images, they were captured in closed rooms with regular lights turned off and with faint indirect illumination set up for this purpose. The illuminance at the camera in both outdoor and indoor scenes is generally between 0.5 lux and 10 lux. In addition, each captured image should generally be shot at 20 – 60 cm in front of the facing camera, and the foreground object(s) should occupy the large majority of the frame as much as possible. We also ensure the captured images should be rich in color and have sharp edges and textures. In addition, we ensure each scene should be also unique, which a single scene should not be shot in multiple lighting conditions or light levels. But rather, the scene content should be varied with lighting conditions, light level, etc. A few pairs of sample images are shown in Figure 3. For the whole dataset, 80% of images were captured under 0.5 - 5 lux, and the rest of them were taken under 5-10 lux.

To minimize the misalignment issues, the smartphone was mounted on a tripod and activated remotely by a wireless control system as shown in Figure 4. We also developed an internal camera capture application to facilitate us to capture aligned images as much as possible. By using the application, a low-light image and its corresponding ground truth can be taken by pressing the wireless controller, and when the capture is complete, there will be an audible "beep" tone. Within this non-interrupted procedure, the smartphone was not touched between the default exposure and the long-exposure images, and the default exposure

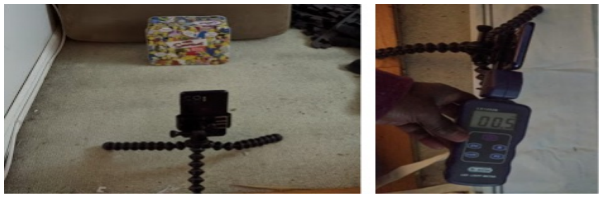


Figure 4. This illustrate what the fixture is and how we used this fixture to collect images with different lux levels. The right image indicates how to calibrate and record lux value when objects being photographed.

low-light image and its corresponding long-exposure reference image can be captured in sequence steadily for avoiding the misalignment issues as much as possible. Note that the captured images are saved as JPEG data after the mature ISP pipeline processing, by which we can benefit from perfect alignment with the help of optical image stabilization.

4. LLNet

For low-light images, the current ISP pipeline can't deliver a good enough result to satisfy customers' expectations for both snapshot and preview. Although some promising methods [2, 10, 25] based on multiple-frames fusion have proved their promising results for snapshot, they can not be applied on viewfinder cases in real-time. To ensure both snapshot and preview can benefit from low-light improvement algorithms, we have to accelerate the operations of processing low-light images. Inspired by the methodology of processing a low-resolution image and then using the low-resolution output to approximate a high-resolution equivalent [9, 15], we propose a hybrid solution, named as LLNet, to perform fast single image processing of low-light images for preview and snapshot. The high-level architecture of LLNet is illustrated in Figure 5. There are two main modules in LLNet: (1) the CNN-based features restoration module is designed to predict a low-resolution image with the excellent image quality from a low-resolution low-light image; (2) the transformation model estimation module is to approximate the transformation relationship between the low-resolution low-light input and low-resolution predicted output. Together with both modules, the estimated transformation model can be applied to a high-resolution low-light image to generate the enhanced high-resolution image efficiently.

4.1. CNN-Based Features Restoration Module

Specifically, this module is a fully convolutional network (FCN) [23] for performing the low-light image improvement with low-resolution input. Recent work has shown that pure FCNs can effectively represent many image processing algorithms [19, 34, 38]. To leverage the successful experience of previous work, we also take the fully-

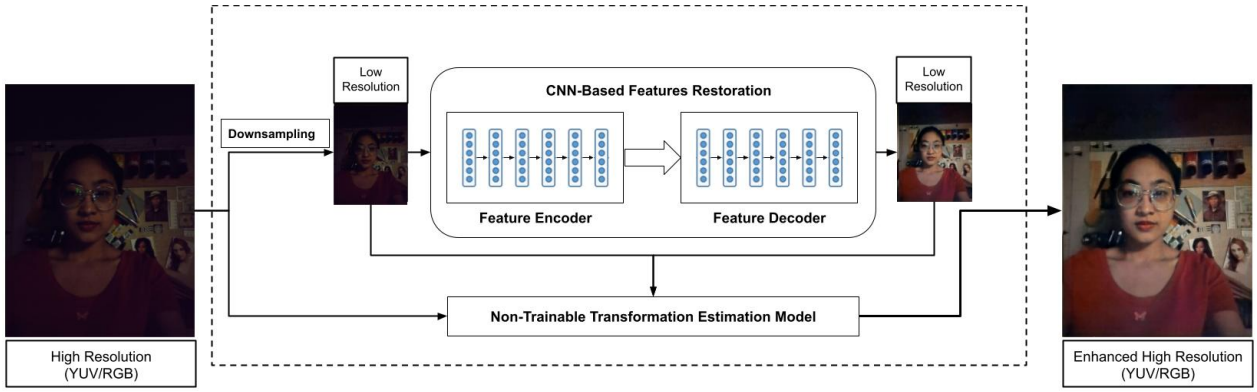


Figure 5. The high-level architecture of LLNet. Please note that we implemented this whole architecture with tensorflow to ensure the model can be fully delegated with GPU for efficient performance.

convolutional network as the foundation of this module. For the specific structure of this module, we have investigated some well-known convolutional architectures, such as U-Net [27], Convolution Pyramids [8] and ResNet [16]. Based on evaluation for computational resources cost, running time, and image quality, U-Net [27] emerged victoriously. As shown in Figure 5, the feature restoration network takes a low-resolution low-light image as the input, which is scaled from the high-resolution low-light image and predicts a low-resolution output with better semantic details, such as better brightness, better contrast, and better color saturation. To strike the optimal balance between image quality and computational costs, our network is designed to predict low-resolution images. However, the optimal low-resolution for achieving the optimal speed/quality trade-offs is different with various requirements and resources. In this paper, the depth of our U-Net [27] based network is 5, which indicates the height and width of the low-resolution must be both multiples of 32. Also, we take 1440*1080 low-light images as our target inputs, the aspect ratio of the low-resolution should be close to the aspect ratio of target inputs. Based on the constraints and our evaluations, the low-resolution 288*244 is the best practice for balancing computational costs and image quality.

4.2. Non-Trainable Transformation Estimation Module

Given a high-resolution low-light image I_{high} with 1440 by 1080, firstly we take the bi-linear interpolation to scale I_{high} down to be I_{low} with 288 by 224. The enhanced image I^o_{low} can be predicted by the CNN-based features restoration module. The challenge here is how to get the enhanced high-resolution output I^o_{high} based on the three images: I_{high}, I_{low}, and I^o_{low}. Noting that I_{low}, I^o_{low} and I^o_{high} can be treated as variants of I_{high}, we can infer that the demanded I^o_{high} is visually similar to I^o_{low} and preserves the edges and other semantic information from I_{high}. Therefore, we infer that an appropriate transformation model between I_{low} and I^o_{low} can be applied on I_{high} to estimate the demanded high-resolution output I^o_{high}. Alternatively, we can formulate this challenge as a specific joint upsampling problem.

In the literature of joint upsampling, guided filter [15] is one of the most widely used algorithms that has shown better performance regarding the trade-off between speed and accuracy of image quality. Most excellent works based on the guided filter have indicated their outstanding efficiency in estimating the transformation model between low-resolution images and delivering high-resolution images by applying the transformation model in various domains. Typically, an intuitive linear transformation model between input image I_{input} and output image I_{out} can be identified as Equation 1, where α is a real-valued scaling factor known as gain, and β is a real-valued offset known as the bias.

$$I_{out} = \alpha * I_{input} + \beta \tag{1}$$

In particular, guided filter expands the Equation 1 to a pixel-wised linear transformation model as Equation 2:

$$I^i_{out} = A^k * I^i_{input} + B^k, \forall i \in \omega_k \tag{2}$$

ω_k is the k-th local square window on I_{input}, and I^i_{input} is the i-th pixel inside ω_k. By applying this Equation 2 on I_{low} and I^o_{low}, the corresponding A_{low} and B_{low} can be estimated by minimizing a reconstruction error between I_{low} and I^o_{low}. And based on the observation that I^o_{high} is visually similar to I^o_{low} and preserves the edges and other semantic information from I_h, we can easily to scale A_{low} and B_{low} up to be A_{high} and B_{high} with bilinear interpolation. Then the high-resolution output I^o_{high} can be approximately generated by the linear transformation model: $I^o_{high} = A_{high} * I_{high} + B_{high}$. Based on procedures of estimating transformation model, we present the module for generating generating high-resolution, edge-preserving outputs with much lower computational costs.

4.3. Model Training

Note that LLNet is a hybrid model, which means that we treat the CNN-based features restoration module as a lightweight trainable network while the transformation model estimation module is a non-trainable network. Then we train the CNN-based network with 2700 low-resolution default-exposed images as inputs while the corresponding low-resolution long-exposure images as ground truths. Once the training is done, we would merge the trainable network and the non-trainable network to be the final LLNet. Based on this training strategy, we can achieve the final model with a fast training time while also keeping the promising predicted results. For 2700 training pairs, each epoch only costs around 40 seconds with batch size 8 on using NVIDIA Tesla GPUs.

We train the networks from scratch using the L_2 loss and the Adam optimizer [21]. In each iteration, we scale down a 288×224 patch for training. The learning rate is initially set to 10^{-4} and Training proceeds for 500 epochs.

4.4. Model Optimization

Since smartphones often have limited memory or computational power, to ensure LLNet can be run within these constraints, various optimizations are applied to LLNet. Noting that LLNet is implemented based on Tensorflow [1], the optimizations are proposed based on the compatibility among Tensorflow, GPU, and the implementation of LLNet. The main optimizations are described as follows:

- For implementation, instead of using Conv2D, we take SeparableConv2D as the default convolutional operation for decreasing the running time and trainable parameters of LLNet.

- Ensure all the operations in the model can be delegated by the GPU. A GPU carries out computations in a very efficient and optimized way, consuming less power and generating less heat than the same task run on a CPU. To leverage the GPU inference, the LLNet should be implemented with only GPU-supported operations.

- Quantizing 32-bit floating-point model to be 16-bit floating-point model resulting in a 2x reduction in model size.

- Ensure the number of channels for each layer to be a multiple of 4. On GPU, tensor data is sliced into 4-channels. Thus, computation on a tensor of shape [B,H,W,5] will perform about the same on a tensor of shape [B,H,W,8] but significantly worse than [B,H,W,4] [1].

5. Experiments

We evaluate the performance of LLNet from the perspectives of image quality and computational costs. LLNet is faster than standard neural network-based solutions and predicts much better-qualified images than conventional techniques on mobile devices.

5.1. Qualitative Results on Smartphones

To indicate the real-time processing ability of LLNet on smartphones, we have delivered our LLNet to commercial-ready smartphones. These smartphones have exact configurations as the Android phone described in Section 2. Video demos have been captured to illustrate the performance of LLNet on the camera preview. The demo videos can be found in supplementary materials. Figure 6 presents the screenshots of such video demos. The low-light data processed by the traditional ISP suffers from low visibility, severe noise, and color shift, but the result of applying LLNet, has much better visibility, good contrast, low noise, and well-adjusted color. And the average running time of preview with our LLNet is about 32ms, and peak memory is around 182MB.

(a) Lux is around 0.7. (b) Lux is around 1.0.

Figure 6. (a) and (b) are the screenshots taken from video demos for evaluating the performance of LLNet on preview of smartphones, compared with the original ISP.

In addition, a comprehensive evaluation of computational costs has been conducted on different Qualcomm Snapdragon Platforms, as shown in Table 1.

5.2. Qualitative and Quantitative Results in Dataset

For the LLNet dataset, we reserve 300 images (1440*1080 resolution) for validation and testing, and train on the remaining 2700. As the evaluation metrics, we employed Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) [32] to quantitatively evaluate the performance of our solution in terms of the color and structure similarity between the predicted results and the corresponding long-exposure images. As we know they are not absolutely indicative, but we can still use PSNR and SSIM val-

Qualcomm Platform	Time	Memory
SM8450 + Adreno660	32ms	182MB
SM8250 + Adreno650	40ms	186MB
SM7325 + Adreno642	48ms	230MB
SM7250 + Adreno620	60ms	228MB
SM6375 + Adreno619	70ms	180MB
SM6115 + Adreno610	105ms	141MB

Table 1. For 1440*1080 images, the computational costs on different Qualcomm Snapdragon Platforms. The results are achieved by averaging 150 iterations of executions with LLNet.

Method	PSNR	SSIM	Time	Memory
Input vs GT	15.02	0.50		
LIME [13]	16.77	0.26	480ms	140MB
SID [3]	24.34	0.67	2.4s	1.8GB
Zero-DCE [12]	19.35	0.63	690ms	1.6GB
HDRNet [9]	20.52	0.60	230ms	788MB
LLNet	24.32	0.67	32ms	182MB

Table 2. Quantitative comparison between LLNet and selected well-known algorithms against 300 testing images (1440*1080 resolution) of LLNet Dataset. For HDRNet, it has a dependent c++ based bilateral_slice module for operations acceleration. Since there is no publicly-available document for integrating this module into smartphones, the way we did integration can cause much more slower running time than claimed in the original paper.

ues to conclude whether proposed solutions could generate reasonably promising results. We evaluate the performance of LLNet with the following four state-of-the-art image enhancement methods: LIME [13], SID [3], Zero-DCE [12] and HDRNet [9]. Table 2 reports the results, where for each case, we re-trained the networks with LLNet dataset, and we produced their results using publicly-available implementation provided by the authors with recommended parameter setting. The computational costs for each method are evaluated on smartphones with the configurations described in Section 2. Figure 7 presents a visual comparison among these algorithms against the testing images. One visual comparison of end-users evaluation is illustrated in Figure 2. As the comparison in smartphones, LLNnet performs better from the perspective of achieving the optimal speed/quality trade-offs.

5.3. LLNet-Evaluation with Different Configurations

Most Appropriate Downsampling: as discussed in Section 4, the light-weight convolutional neural network of LLNet will predict a low-resolution enhanced image from a low-resolution low-light image. Alternatively, we need to scale a high-resolution low-light image down to be an appropriate low-resolution image for achieving the optimal speed/quality trade-offs that are different with various requirements and resources. To explore the most appropriate downsampling levels of input images for LLNet, we have evaluated the performance of LLNet with different downsampling levels, as shown in Table 3. And a visual comparison is also provided in Figure 8.

Resolution After Downsampling	PSNR	SSIM	Time	Memory
1440*1080 (No Downsampling)	24.67	0.68	260ms	900MB
612*576	24.54	0.67	110ms	420MB
576*448	24.34	0.67	94ms	300MB
288*224	24.32	0.67	32ms	182MB

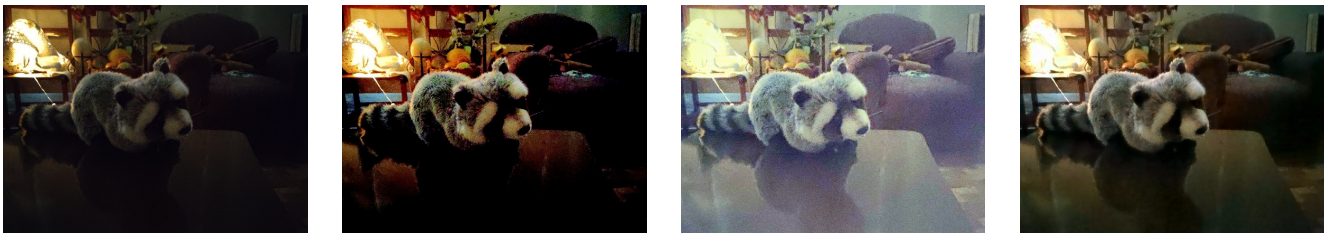
Table 3. Quantitative comparison among different downsampling leves for LLNet. The 1440*1080 resolution indicates that we only use the convolutional neural network to predict high-resolution images without the Transformation Estimation Module. From this table and manually evaluating image quality, we can conclude that larger resolution will result in better image quality as well as increasing the computational costs.

Loss Functions: different from most recent image restoration efforts [37] using L_1 as the optimal loss function, in this paper, we take the L_2 loss by as our default based on our evaluation which indicate L_2 can provide better image quality than L_2, especially in sharpness and color. In addition, we also evaluate many alternative loss functions, such as SSIM, and the combination of L_1 and MS-SSIM [37]. However, we have not observed systematic perceptual benefits for these loss functions. Figure 9 presents the visual comparison between L_1 and L_2.

6. Discussion and Limitation

To balance the image quality and computation costs, we need to scale the images down to be low-resolution images and feed them to the convolutional neural network for predicting. In this paper, we take 288*224 as the target input resolution of the network for achieving the most appropriate speed/quality trade-offs. However, for some extreme low-light images, noise becomes a dominant issue for the predicted images by our LLNet shown in Figure 8. Thus, striking the optimal balance between image quality and efficiency with better denoising is another challenge.

In addition, we expect future work to yield further improvements in image quality by systematically optimizing the network architecture and training procedure. For instance, in our current implementation of LLNet, we take SeparableConv2D to perform convolution in all layers. But how to mix Conv2D and SeparableConv2D for different layers to achieve a better image quality is still ongoing work.



(a) Input (b) LIME [13] (c) Retinex [26] (d) SID [3]

(e) Zero-DCE [12] (f) HDRNet [9] (g) LLNet (h) GT

Figure 7. Visual comparison with state-of-the-art methods on a test image (a) from our dataset. The test image were captured around 0.8 lux.

(a) Input (b) 288*224-(Default) (c) 576*448 (d) 672*576 (e) 1440*1080

Figure 8. A visual comparison among different downsampling levels of LLNet. Based on our evaluation, a larger resolution will result in better image quality.

(a) Input (b) LLNet+L_1 (c) LLNet+L_2

Figure 9. A visual comparison of LLNet with two different loss functions: L_1 and the default L_2. From the example, L_2 provides much better color saturation and sharpness.

7. Conclusion

We present an efficient hybrid architecture (LLNet) with the combination of a lite CNN and a non-trainable transformation estimation model that can perform low-light images (1440*1080) enhancement on smartphones in real-time. We also present the LLNet dataset to support the development of learning-based architecture. Benefiting from this hybrid architecture, LLNet can strike an appropriate balance between image quality and computational costs. Experiments demonstrate that LLNet is capable of delivering enhanced high-resolution outputs with good quality and affordable resources consumption on mobile devices.

Acknowledgements

We thank our team members for their awesome support on evaluation, data collection and phone integration. Special thanks to Yunming Wang, Minxun Peng, Hong Zhao, Nathaniel Mitchell and Nigil Valikodath for their valuable efforts. We also thank Thomas Merrell from Motorola camera team for his support on data collection tool building up.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 6

[2] P.J. Burt and R.J. Kolczynski. Enhanced image capture through fusion. In *1993 (4th) International Conference on Computer Vision*, pages 173–182, 1993. 4

[3] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3, 7, 8

[4] Qifeng Chen, Jia Xu, and Vladlen Koltun. Fast image processing with fully-convolutional networks. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2516–2525. IEEE Computer Society, 2017. 2, 3

[5] Yu-Sheng Chen, Yu-Ching Wang, Man-Hsin Kao, and Yung-Yu Chuang. Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6306–6314, 2018. 3

[6] D. Coltuc, P. Bolon, and J.-M. Chassery. Exact histogram specification. *IEEE Transactions on Image Processing*, 15(5):1143–1152, 2006. 2

[7] Xuan Dong, Guan Wang, Yi Pang, Weixin Li, Jiangtao Wen, Wei Meng, and Yao Lu. Fast efficient algorithm for enhancement of low lighting video. In *2011 IEEE International Conference on Multimedia and Expo*, pages 1–6, 2011. 3

[8] Zeev Farbman, Raanan Fattal, and Dani Lischinski. Convolution pyramids. *ACM Trans. Graph.*, 30(6):1–8, Dec. 2011. 5

[9] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):118, 2017. 2, 3, 4, 7, 8

[10] A. Ardeshir Goshtasby. Fusion of multi-exposure images. *Image and Vision Computing*, 23(6):611–618, 2005. 4

[11] Xu Guan, Su Jian, Pan Hongda, Zhang Zhiguo, and Gong Haibin. An image enhancement method based on gamma correction. In *2009 Second International Symposium on Computational Intelligence and Design*, volume 1, pages 60–63, 2009. 2

[12] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Cong Runmin. Zero-reference deep curve estimation for low-light image enhancement. *CVPR*, 2020. 2, 3, 7, 8

[13] Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2):982–993, 2017. 2, 3, 7, 8

[14] Sam Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *SIGGRAPH Asia*, 2016. 3

[15] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013. 4, 5

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 5

[17] Haidi Ibrahim and Nicholas Sia Pik Kong. Brightness preserving dynamic histogram equalization for image contrast enhancement. *IEEE Transactions on Consumer Electronics*, 53(4):1752–1758, 2007. 2

[18] A. Ignatov, N. Kobyshev, R. Timofte, and K. Vanhoey. Dslr-quality photos on mobile devices with deep convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3297–3305, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society. 3

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 2, 3, 4

[20] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Han Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE Transactions on Image Processing*, 30:2340–2349, 2021. 3

[21] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 6

[22] Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE Transactions on Image Processing*, 22(12):5372–5384, 2013. 2

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 4

[24] Henrik Malm, Magnus Oskarsson, Eric Warrant, Petrik Clarberg, Jon Hasselgren, and Calle Lejdfors. Adaptive enhancement and noise reduction in very low light-level video. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007. 3

[25] Tom Mertens, Jan Kautz, and Frank Van Reeth. Exposure fusion. In *15th Pacific Conference on Computer Graphics and Applications (PG'07)*, pages 382–390, 2007. 4

[26] Seonhee Park, Byeongho Moon, Seungyong Ko, Soohwan Yu, and Joonki Paik. Low-light image enhancement using variational optimization-based retinex model. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*, pages 70–71, 2017. 2, 3, 8

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation.

In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 5

[28] J.A. Stark. Adaptive image contrast enhancement using generalizations of histogram equalization. *IEEE Transactions on Image Processing*, 9(5):889–896, 2000. 2

[29] Magudeeswaran Veluchamy and Bharath Subramani. Image contrast and color enhancement using adaptive gamma correction and histogram equalization. *Optik*, 183:329–337, 2019. 2

[30] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6842–6850, 2019. 3

[31] Wenjing Wang, Chen Wei, Wenhan Yang, and Jiaying Liu. Gladnet: Low-light enhancement network with global awareness. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 751–755, 2018. 3

[32] Zhou 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, 2004. 6

[33] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *ArXiv*, abs/1808.04560, 2018. 3

[34] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 4

[35] Zhicheng Yan, Hao Zhang, Baoyuan Wang, Sylvain Paris, and Yizhou Yu. Automatic photo adjustment using deep neural networks. *ACM Transactions on Graphics*, 35, 05 2016. 3

[36] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 1632–1640, New York, NY, USA, 2019. Association for Computing Machinery. 3

[37] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, 2017. 7

[38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 2, 3, 4

[39] Artur Łoza, David R. Bull, Paul R. Hill, and Alin M. Achim. Automatic contrast enhancement of low-light images based on local statistics of wavelet coefficients. *Digital Signal Processing*, 23(6):1856–1866, 2013. 3