Adaptive Unfolding Total Variation Network for Low-Light Image Enhancement

Chuanjun Zheng   Daming Shi*   Wentian Shi
Shenzhen University
{zhengchuanjun2019,shiwentian2018}@email.szu.edu.cn, dshi@szu.edu.cn

Abstract

Real-world low-light images suffer from two main degradations, namely, inevitable noise and poor visibility. Since the noise exhibits different levels, its estimation has been implemented in recent works when enhancing low-light images from raw Bayer space. When it comes to sRGB color space, the noise estimation becomes more complicated due to the effect of the image processing pipeline. Nevertheless, most existing enhancing algorithms in sRGB space only focus on the low visibility problem or suppress the noise under a hypothetical noise level, leading them impractical due to the lack of robustness. To address this issue, we propose an adaptive unfolding total variation network (UTVNet), which approximates the noise level from the real sRGB low-light image by learning the balancing parameter in the model-based denoising method with total variation regularization. Meanwhile, we learn the noise level map by unrolling the corresponding minimization process for providing the inferences of smoothness and fidelity constraints. Guided by the noise level map, our UTVNet can recover finer details and is more capable to suppress noise in real captured low-light scenes. Extensive experiments on real-world low-light images clearly demonstrate the superior performance of UTVNet over state-of-the-art methods.

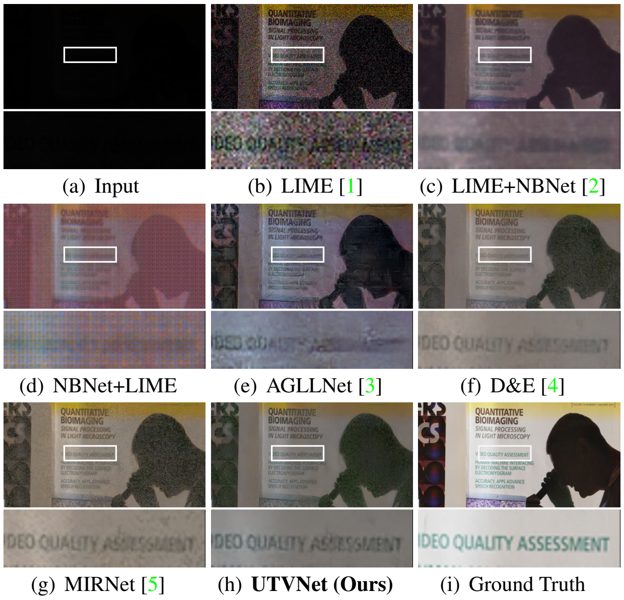

(a) Input (b) LIME [1] (c) LIME+NBNet [2]
(d) NBNet+LIME (e) AGLLNet [3] (f) D&E [4]
(g) MIRNet [5] (h) **UTVNet (Ours)** (i) Ground Truth

Figure 1. Performance in real captured low-light scene. (a) a sRGB image captured by a NikonD850 camera from ELD dataset [6]. (b-d) results of the luminance correction and denoising approaches. (f) and (g) are outputs of two recent methods without noise estimation. (e) the result of [3] trained on the synthetic low-light dataset to predict noise maps. (h) the result of proposed UTVNet. *All images were cropped to make their details more visible.*

1. Introduction

The insufficient light provided by low-light scenarios is the main limitation of capturing high-quality photographs. Besides enlarging the aperture and prolonging the exposure time to gather more light when taking images, another flexible solution is to increase the ISO setting of camera to adjust the analog gain or both analog and digital gains as the compensation for the brightness. However, a higher ISO setting will decrease the signal-to-noise ratio (SNR). The SNR should be carefully managed to control the noise for providing higher possibilities of post-processing [7]. Unfortunately, such a manipulation is difficult for non-expert users who lack the appropriate photography skills.

Restoration of real-world low-light images is a challenging process in computer vision, which focuses on recovering underlying clean images from observed noisy ones and correcting the luminance without amplifying the noise. Most existing methods [1, 8, 9] focus on enhancing noise-free underexposed images. Such methods may solve the low visibility problem, but the noise will amplified when adjusting the exposure as shown in Fig. 1 (b). Meanwhile, existing denoising approaches [10, 11, 2] may fail to suppress noise for low-light images, since most of them focus on images taken under normal luminance, the small magnitude between pixels in extremely low-light images hinder the detection of noise from details as shown in Fig. 1 (c, d).

*Corresponding author

As real-world low-light images exhibit different noise levels, recent low-light image restoration methods [4, 5] may lack of robustness without noise estimation as shown in Fig. 1 (f, g). In raw Bayer space, a physics-based noise formation model [6] has been proposed for low-light raw images. When converting raw data to sRGB, the distribution of noise is affected by the image processing pipeline (ISP). Existing method [3] learned noise maps by synthetic low-light image dataset, but the efficiency is degraded by the inaccurate noise model and the gap between synthetic images and real photographs as shown in Fig. 1 (e).

In this paper, an adaptive unfolding total variation network (UTVNet) is presented to restore the extremely low-light images in sRGB space. To boost the robustness in real captured scenes, we approximate the noise level inspired by the relationship between the noise level and the balancing parameter in model-based image denoising method. Specifically, we formulate an adaptive total variation (TV) regularization by introducing a learnable noise level map that serves as the balancing parameters for each pixel to control the trade-off between noise reduction and detail preservation. To learn the noise level map from the real low-light dataset, designing regularizers in loss function is infeasible. Therefore, we unfold the TV minimization process via the Alternating Direction Method of Multiplier (ADMM) [12] to separate the minimization problem into three subproblems with closed-form solutions, which provides the inferences of fidelity and smoothness constraints for learning the noise level map. Subsequently, a noise-free low-frequency layer is automatically generated along with the noise level map. Since the noise is smoothed out by the TV regularization in the low-frequency layer, the luminance correction is applied in this layer to avoid noise amplification. Meanwhile, the noise suppression process is guided by the approximated noise level map in the high-frequency layer. The main contributions of this work are as follows:

- By integrating a learnable nois level map, we formulate an adaptive TV regularization and propose a new noise level approximation method for real low-light photographs in sRGB color space.
- We propose an UTVNet that unfolds the total variation minimization algorithms to provide fidelity and smoothness constraints to learn the noise level map without designing special loss function.
- UTVNet is robust in restoring the real captured low-light images with various noise levels.

2. Related Work

Low-light image enhancement. Most low-light image enhancement approaches only point to underexposure without considering noise suppression, like learning-based methods [9, 13, 14], and retinex-based methods [1, 15], in which most of the cases are noise-free, therefore, these methods handle the low-visibility problem only. Some other methods take noise suppression into account in raw Bayer space. Chen *et al.* [16] proposed a raw low-light image dataset to learning the mapping from raw short-exposure images to the long-exposure references. Wei *et al.* [6] proposed a physics-based noise formation model for low-light raw image denoising. However, as the noise is affected by the ISP during the conversion from raw to sRGB, modeling noise is more difficult in the sRGB color space. Lv *et al.* [3] constructed a large sRGB synthetic dataset to learn noise maps for low-light images. The performance is related to the gap between synthetic images and real photographs. Xu *et al.* [4] suppressed the noise for low-light images by learning a low-frequency layer of the input image using the reference generated by a guided filter [17]. Zamir *et al.* [5] proposed a multi-scale residual block that selects useful sets of kernels from each branch using a self-attention approach. Without noise estimation, such noise suppression networks may lack robustness in real captured scenes.

Image denoising. Some blind denoising methods can handle the real-world image denoising tasks, these methods often combine noise estimation with non-blind denoising. Xu *et al.* [18] proposed a multi-channel weighted unclear norm minimization model for real color image denoising. Nam *et al.* [19] proposed a cross-channel noise model in sRGB space which took the correlation between the color channels into consideration. Among learning-based methods, Zhang *et al.* [10] stretched the noise level as a noise-level map to deal with spatially variant noise. Cheng *et al.* [2] proposed a network that separated signal and noise by learning a set of reconstruction basis in the feature space. Guo *et al.* [11] generated synthetic images and adopted a sub-network to learn the mapping from estimated noise level to ground truth. Instead of generating such a dataset, our UTVNet aims to learn the noise level approximation directly from real low-light images.

Deep unrolling architecture. Model-based denoising methods are flexible in handling noise with various levels. Most model-based denoising methods can be formulated as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{y}\|^2 + \lambda\mathbf{\Phi}(\mathbf{x}), \tag{1}$$

where $\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{y}\|^2$ is the data fidelity term, σ is the noise level , λ is the balancing parameters, $\mathbf{\Phi}(\boldsymbol{x})$ is the regularization term, $\hat{\mathbf{x}}$ is the solution to the problem, $\mathbf{x}$ and $\mathbf{y}$ are clean image and observed image, respectively. The performance of these methods is associated with image priors, such as self-similarity [20], sparsity [21], and low-rank [22]. However, most model-based methods focus only on how to carefully design the denoising priors, the parameter λ is neglected. It is often set as an empirical value. Recently, deep unrolling methods have been proposed that explicitly

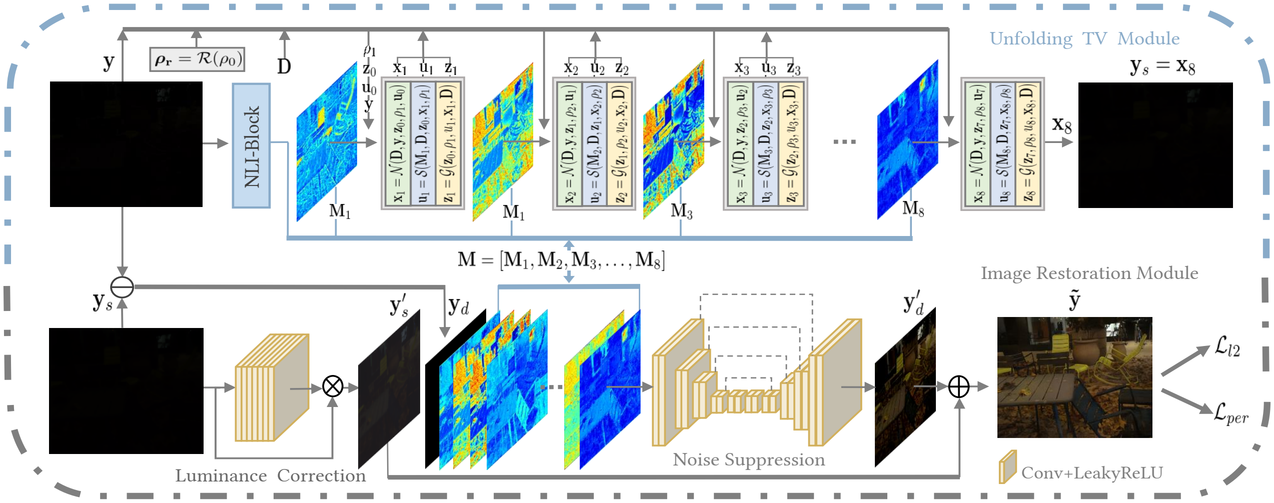

Unfolding TV Module
y
$\rho_r = \mathcal{R}(\rho_0)$
D
NLI-Block
ρ_1
z_0
u_0
y
x_1 u_1 z_1
$x_1 = \mathcal{N}(D, y, z_0, \rho_1, u_0)$
$u_1 = \mathcal{S}(M_1, D, z_0, x_1, \rho_1)$
$z_1 = \mathcal{G}(z_0, \rho_1, u_1, x_1, D)$
M_1
x_2 u_2 z_2
$x_2 = \mathcal{N}(D, y, z_1, \rho_2, u_1)$
$u_2 = \mathcal{S}(M_2, D, z_1, x_2, \rho_2)$
$z_2 = \mathcal{G}(z_1, \rho_2, u_2, x_2, D)$
M_1
x_3 u_3 z_3
$x_3 = \mathcal{N}(D, y, z_2, \rho_3, u_2)$
$u_3 = \mathcal{S}(M_3, D, z_2, x_3, \rho_3)$
$z_3 = \mathcal{G}(z_2, \rho_3, u_3, x_3, D)$
M_3
...
M_8
$x_8 = \mathcal{N}(D, y, z_7, \rho_8, u_7)$
$u_8 = \mathcal{S}(M_8, D, z_7, x_8, \rho_8)$
$z_8 = \mathcal{G}(z_7, \rho_8, u_8, x_8, D)$
x_8
$y_s = x_8$
$M = [M_1, M_2, M_3, \ldots, M_8]$
Image Restoration Module
y_s
y'_s
y_d
Luminance Correction
Noise Suppression
y'_d
$\tilde{y}$
$\mathcal{L}_{l2}$
$\mathcal{L}_{per}$
Conv+LeakyReLU

Figure 2. The proposed UTVNet consists of an unfolding TV module and an image restoration module. First, the given image $\mathbf{y}$ is fed into unfolding TV module to estimate noise level map $\mathbf{M}$ and generate noise-free smooth layer $\mathbf{y_s}$. Then, in image restoration module, $\mathbf{y_s}$ is used to correct the luminance, and the noise suppression in detail layer $\mathbf{y_d}$ is guided by the approximated noise level map $\mathbf{M}$. We describe the details of the two modules in Section 3.

unroll iterative optimization algorithms into learnable deep architectures combined with convolutional neural networks (CNNs). ADMM [23] has been used in [24] to unroll the minimization process for image layer separation, and the convergence has been analysed in [25]. Meanwhile, the half-quadratic splitting algorithm [26] has been utilized in [27, 10, 28, 29] to unfold the minimization problems for image denoising and super-resolution. Different from those methods that utilize CNNs to fit the physical priors in minimization problems, our UTVNet adopts the TV regularization and unfolds the minimization process for the fidelity and smoothness inferences.

3. Methodology

In this section, we introduce the proposed UTVNet. An overview of the process is shown in Fig. 2. The network consists of two modules, including an unfolding TV module and an image restoration module. To approximate the noise level, we start with the analysis of the noise level and the balancing parameter in model-based denoising methods. Then we formulate an adaptive TV regularization and describe each module of the network in detail below.

3.1. Noise Level vs. Balancing Parameter

As described in previous work [10], when λ is combined into σ in Eq. (1), setting noise level σ also plays the role of setting λ to control the trade-off between noise reduction and detail preservation. Inspired by this relationship, as for certain regularization terms, if we combine λ and σ as the balancing parameter in Eq. (1), the noise level can be approximated by learning such balancing parameter via unrolling architecture. In our model, we adopt the TV regularization term for three reasons. First, TV regularization can better preserve sharp edges [30, 31] and also be used in enhancing images with poor visibility [32]. Second, there are no extra parameters of the original TV regularization, accordingly, the trade-off is controlled only by the balancing parameter, which is easy and accurate for approximation. Third, instead of using CNNs to learn the explicit mapping of solutions to the minimization problem, one can easily find closed-form solutions for the TV regularization term, which provide the fidelity and smoothness constraints without using special loss functions.

3.2. Adaptive Total Variation Regularization

We first define operator $\mathbf{D}$ as a collection of two sub-operators $\mathbf{D} = [\mathbf{D}_x^T \ \mathbf{D}_y^T]^T$, where $\mathbf{D}_x, \mathbf{D}_y$ are the first-order forward finite-difference operators along the horizontal and vertical directions respectively. Thus, the anisotropic TV regularization term can be written as $\|\mathbf{Dx}\|_1$. Meanwhile, the balancing parameter λ can be combined into the regularization term, written as $\|\lambda\mathbf{Dx}\|_1$. Since the real-world noise exhibits different patterns, following [10, 11], we stretch λ into a map $\hat{\mathbf{M}} = [\mathbf{M}_x^T \ \mathbf{M}_y^T]^T$, where the size of $\mathbf{M}_x$ and $\mathbf{M}_y$ are the same as $\mathbf{x}$. Therefore, the adaptive TV regularization minimization problem is formulated as

$$\underset{\mathbf{x}}{\text{minimize}} \quad \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2 + \|\hat{\mathbf{M}} \circ (\mathbf{Dx})\|_1, \qquad (2)$$

where "∘" denotes component-wise multiplication. In our UTVNet, we set $\mathbf{M}_x = \mathbf{M}_y = \mathbf{M}$ as the noise level map that we need to approximate in the network. By introducing intermediate variable $\mathbf{u} = \mathbf{Dx}$, we have the augmented

Lagrangian function of Eq. (2)

$$L(\mathbf{x},\mathbf{u},\mathbf{z})=\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2+\|\hat{\mathbf{M}}\circ\mathbf{u}\|_1-\mathbf{z}^T(\mathbf{u}-\mathbf{D}\mathbf{x})+\frac{\rho_r}{2}\|\mathbf{u}-\mathbf{D}\mathbf{x}\|^2, \tag{3}$$

where $\mathbf{z}$ is the Lagrange multiplier, ρ_r is a regularization parameter. Through ADMM, the problem can be solved by iteratively updating $\mathbf{x}$, $\mathbf{u}$, and $\mathbf{z}$, other variables treated as constants. Consequently, the unfolding inference can be obtained by solving the below sub-problems

$$\begin{cases} \mathbf{x}_{k+1}=\arg\min\limits_{\mathbf{x}}\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2-\mathbf{z}_k^T\left(\mathbf{u}_k-\mathbf{D}\mathbf{x}\right)+\frac{\rho_r}{2}\left\|\mathbf{u}_k-\mathbf{D}\mathbf{x}\right\|^2, \\ \mathbf{u}_{k+1}=\arg\min\limits_{\mathbf{u}}\|\hat{\mathbf{M}}\circ\mathbf{u}\|_1-\mathbf{z}_k^T(\mathbf{u}-\mathbf{D}\mathbf{x}_{k+1})+\frac{\rho_r}{2}\|\mathbf{u}-\mathbf{D}\mathbf{x}_{k+1}\|^2, \\ \mathbf{z}_{k+1}=\mathbf{z}_k-\rho_r\left(\mathbf{u}_{k+1}-\mathbf{D}\mathbf{x}_{k+1}\right), \end{cases} \tag{4}$$

where k denotes the k-th iteration. Next, the fast Fourier transform (FFT) and the shrinkage function can be utilized to solve x-subproblem and u-subproblem respectively [33, 31]. Hence, the closed-form solutions to the sub-problems in Eq. (4) is

$$\begin{cases} \mathbf{x}_{k+1}=\mathcal{F}^{-1}\left[\frac{\mathcal{F}\left[\mathbf{y}+\rho_r\mathbf{D}^T\mathbf{u}_k-\mathbf{D}^T\mathbf{z}_k\right]}{1+\rho_r\left(|\mathcal{F}[\mathbf{D}_x]|^2+|\mathcal{F}[\mathbf{D}_y]|^2\right)}\right], \\ \mathbf{u}_{k+1}=\max\left\{|\mathbf{v}_{k+1}|-\frac{\hat{\mathbf{M}}}{\rho_r},0\right\}\cdot\operatorname{sign}\left(\mathbf{v}_{k+1}\right), \\ \mathbf{z}_{k+1}=\mathbf{z}_k-\rho_r\left(\mathbf{u}_{k+1}-\mathbf{D}\mathbf{x}_{k+1}\right), \end{cases} \tag{5}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote FFT and inverse FFT, respectively, $sign(\cdot)$ denotes the sign function, $\mathbf{v}$ is defined as $\mathbf{v}_{k+1}=\mathbf{D}\mathbf{x}_{k+1}+(1/\rho_r)\,\mathbf{z}_k$. By using the above three closed-form solutions iteratively, the minimization problem in Eq. (2) can be solved. We then analyze the relationship between three closed-from solutions and design unfolding TV module to learn the noise level map $\mathbf{M}$.

3.3. Unfolding Total Variation Module

Based on three closed-form solutions, we design an unfolding TV Module as shown in Fig. 2. First, image $\mathbf{y}$ is fed into noise level initialization block (NLI-block) to predict noise level map $\mathbf{M}$. Then, we unfold the TV minimization problem and iteratively utilize Eq. (5) to guarantee the smoothness and fidelity constraints for learning $\mathbf{M}$ end-to-end and generating noise-free smooth layer $\mathbf{y_s}$.

Unfolding TV Architecture. In Eq. (4), it can be observed that the data fidelity term and the TV regularization term are decoupled into the x- and u- subproblems. The parameter $\mathbf{M}$ only exists in the u-subproblem. Because $\mathbf{M}$ is also the hyper-parameter of the iterative algorithm, instead of learning a fixed $\mathbf{M}$ for all iterations, here we follow [29], in which learning a series of hyper-parameters varies in different iterations. Hence, we set $\mathbf{M}=[\mathbf{M}_1,\mathbf{M}_2,\mathbf{M}_3,\ldots,\mathbf{M}_K]$, $\boldsymbol{\rho}_\mathbf{r}=[\rho_1,\rho_2,\rho_3,\ldots,\rho_K]$, where K is the number of iteration times. For speed-accuracy trade-off, K is set to 8. Following the unfolding optimization, the unfolding TV architecture alternatively solves the $\mathbf{x}$, $\mathbf{u}$, $\mathbf{z}$ sub-problems by Eq. (5), they are written as below and illustrated in Fig. 2

$$\begin{cases} \mathbf{x}_k=\mathcal{N}(\mathbf{y},\mathbf{z}_{k-1},\rho_k,\mathbf{D},\mathbf{u}_{k-1}), \\ \mathbf{u}_k=\mathcal{S}(\mathbf{M}_k,\mathbf{D},\mathbf{z}_{k-1},\mathbf{x}_k,\rho_k), \\ \mathbf{z}_k=\mathcal{G}(\mathbf{D},\mathbf{z}_{k-1},\rho_k,\mathbf{u}_k,\mathbf{x}_k). \end{cases} \tag{6}$$

In the following descriptions, we will analyze the importance of three functions for learning the noise level map $\mathbf{M}$.

Function $\mathcal{N}(\cdot)$ and function $\mathcal{G}(\cdot)$ only contain the hyper-parameter ρ_k, but such two function are essential for the UTVNet. Function $\mathcal{N}(\cdot)$ aims to solve the fidelity term sub-problem, and guarantees the inference between the noise-free smooth layer $\mathbf{y}_s$ and the original image $\mathbf{y}$. Such inference provides the fidelity prior that speeds up the convergence of the whole network when training our model. Function $\mathcal{G}(\cdot)$ is associated with constraint $\mathbf{u}=\mathbf{D}\mathbf{x}$, as iteration goes, the Lagrange multiplier $\mathbf{z}$ should be updated. In PyTorch, torch.fft.fftn and torch.fft.ifftn are used to implement $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$.

Function $\mathcal{S}(\cdot)$ contains the approximated noise level map $\mathbf{M}$. The function can be regarded as a special smoothness constraint for the low-frequency layer $\mathbf{y_s}$, where it smooths out details and noise guided by the magnitude of noise level map $\mathbf{M}$ for each pixel. Using such a smoothness inference between the original image and the smoothed one, we can learn the parameter $\mathbf{M}$ for input images without designing a special loss function. As shown in Fig. 2, the parameter $\mathbf{M}$ varies in each iteration to control the trade-off between noise reduction and detail preservation.

There are two learning parameters in this architecture, $\boldsymbol{\rho}_\mathbf{r}$ and $\mathbf{M}$. $\boldsymbol{\rho}_\mathbf{r}$ is generated by $\boldsymbol{\rho}_\mathbf{r}=\mathcal{R}(\rho_0)$ in Fig. 2, where we adopt the hyper-parameter module in [29] that uses three 1×1 convolutions with two ReLU and one Softplus activation functions. To avoid dividing by extremely small ρ_k, we add $1e-6$ to the final output. The prediction of parameter $\mathbf{M}$ will be described in detail below.

NLI-Block. Before iteration, a coarse-to-fine noise level initialization block is proposed in Fig. 2 to predict the noise level in a coarse-to-fine way. The details of the NLI-block can be seen in Fig. 3. We first estimate global noise variation $\boldsymbol{\varepsilon}$ for three channels $\boldsymbol{\varepsilon}=\left[\varepsilon^r,\varepsilon^g,\varepsilon^b\right]$ as coarse approximated values according to [34]

$$\varepsilon^c=\sqrt{\frac{\pi}{2}}\frac{1}{6(w-2)(h-2)}\sum_{(i,j)}\left|\left(\mathbf{y}^\mathbf{c}\otimes\mathbf{N}\right)(i,j)\right|, \tag{7}$$

where $c\in\{r,g,b\}$ denotes the RGB channels, "$\otimes$" denotes the convolution operator. (i,j) denotes indices of the pixel position. h and w are the height and width of $\mathbf{y}^c$, respectively. $\mathbf{N}$ is defined as

$$\mathbf{N}=\left[\begin{array}{ccc} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{array}\right]. \tag{8}$$

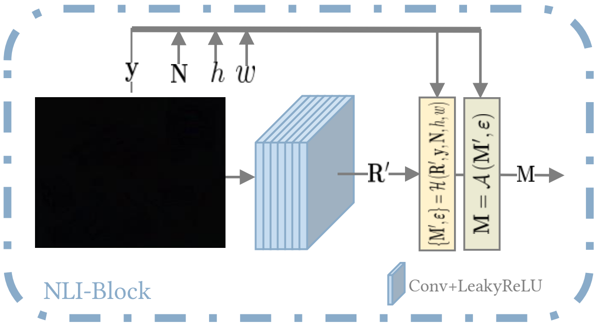


Figure 3. Illustration of the NLI-Block. The details of function $\mathcal{H}(\cdot)$ and function $\mathcal{A}(\cdot)$ are described in Section 3.3.

Then, we adopt a seven dilated convolution layers with LeakyReLU activation function for the first six layers to predict the residual $\mathbf{R}' = [\mathbf{R}'_1, \mathbf{R}'_2, \dots, \mathbf{R}'_K]$, which can be used to modify the global noise variation ε and generate precise noise level values for each pixel. Hence, the noise level map $\mathbf{M}' = [\mathbf{M}'_1, \mathbf{M}'_2, \dots, \mathbf{M}'_K]$ is formulated as

$$\mathbf{M}' = \left[\mathbf{R}'_1 + \varepsilon, \mathbf{R}'_2 + \varepsilon, \dots, \mathbf{R}'_K + \varepsilon\right]. \tag{9}$$

We define such methods for two reasons. First, the small pixel values make it harder to learn the noise level map $\mathbf{M}$ directly in extremely low-light images. The modification of the global noise variation provides a better initialization and accelerate the speed of convergence for the unfolding TV module. Second, as indicated in [10], a larger noise level may guarantee the role of $\mathbf{M}$ in controlling the trade-off between noise reduction and detail preservation. The global noise variation also provides a basic magnitude for learning the parameter $\mathbf{M}$, which ensures an accurate approximation of the noise level. As the values in the noise level map should be positive, the final output $\mathbf{M}^c_k(i,j)$ is defined as

$$\mathbf{M}^c_k(i,j) = \begin{cases} \mathbf{M}'^c_k(i,j) & \mathbf{M}'^c_k(i,j) > 0, \\ \varepsilon^c & \mathbf{M}'^c_k(i,j) \leq 0, \end{cases} \tag{10}$$

$k \in \{1, 2, 3, \dots, K\}$ denotes k-th iteration. For brevity, in Fig. 3, Eq. (7) and Eq. (9) are denoted as Eq. (11a), meanwhile, Eq. (10) is denoted as Eq. (11b)

$$\begin{cases} \{\mathbf{M}', \varepsilon\} = \mathcal{H}\left(\mathbf{R}', \mathbf{y}, \mathbf{N}, h, w\right), & \text{(11a)} \\ \mathbf{M} = \mathcal{A}\left(\mathbf{M}', \varepsilon\right). & \text{(11b)} \end{cases}$$

$\mathcal{A}(\cdot)$ is similar to ReLU but contains an important modification. Different from ReLU that totally sets all negative values as zero, we set all negative values to the global noise variation ε. Such a treatment also provides an adaptive higher noise level value that guarantees the role of $\mathbf{M}$ as the balancing parameter λ. It is notable that the smoothed-out detail can be recovered when suppressing the noise guided by $\mathbf{M}$ in the image restoration module. The effectiveness of the NLI-block will be discussed in Section 4.3.

3.4. Low-light Image Restoration Module

The image restoration module consists of a luminance correction sub-network and a noise suppression sub-network. Through unrolling the TV module, the noise-free smooth layer $\mathbf{y}_s$ and the approximated noise level map $\mathbf{M}$ are generated. As shown in Fig. 2, $\mathbf{M}$ is utilized for noise suppression on the detail layer $\mathbf{y}_d = \mathbf{y} - \mathbf{y}_s$. Meanwhile, the smooth layer $\mathbf{y}_s$ is the input of the luminance correction sub-network. The details are described below.

Luminance correction sub-network. The luminance correction sub-network generates luminance amplification for each pixel. By multiplying the luminance amplification with $\mathbf{y}_s$, we get the luminance corrected smooth layer $\mathbf{y}'_s$. The sub-network adopts seven dilated convolution layers with LeakyReLU in the first six layers.

Noise suppression sub-network. As noted in [35, 10], taking both the noisy image and the noise level map as input helps generalize the learned model for blind denoising. For low-light image restoration, we also adopt a method that takes $\mathbf{y}_d$ and the approximated noise level map $\mathbf{M}$ as input for noise suppression and detail restoration of the original low-light image. As shown in Fig. 2, we adopt U-Net [36] architecture with four scales, and the channels of the convolution layer in each scale are 32, 64, 128 and 256, respectively. Average pooling and fully-connected layers are used to process global features and connected them with former layers. 2×2 convolution and transposed convolution are adopted in downscaling and upscaling layers. For all convolution layers except the last, we use LeakyReLU as the activation function.

3.5. End-to-End Training

As the TV minimization problem is solved in unfolding TV module when approximating the noise level map, the network can be trained end-to-end. We optimize weights and biases by minimizing the L_2 loss $\mathcal{L}_{l2}$ and the perceptual loss $\mathcal{L}_{per}$ with a weight of 1 and 0.12, respectively.

4. Experiment

Implementation detail. The proposed UTVNet[1] is implemented in PyTorch framework, before training, we initialize parameters for UTVNet. We set $\rho_0 = 2$, $\mathbf{x}_0 = \mathbf{y}$, $\mathbf{u}_0 = \mathbf{D}\mathbf{x}_0$, $\mathbf{z}_0 = 0$, and $K = 8$. To optimize parameters of UTVNet, we adopt the ADAM optimizer [12] for 320 epochs with an initial learning rate of 10^{-4}. The learning rate is reduced by half after 240 epochs. During training, the images are resized to have a long-edge of 512 pixels to avoid unnatural deformation. All the experiments are performed on a single NVIDIA TITAN Xp GPU.

Dataset. To suppress noise and enhance real-world low-light images in the sRGB domain, we train the UTVNet

[1] Code: https://github.com/CharlieZCJ/UTVNet

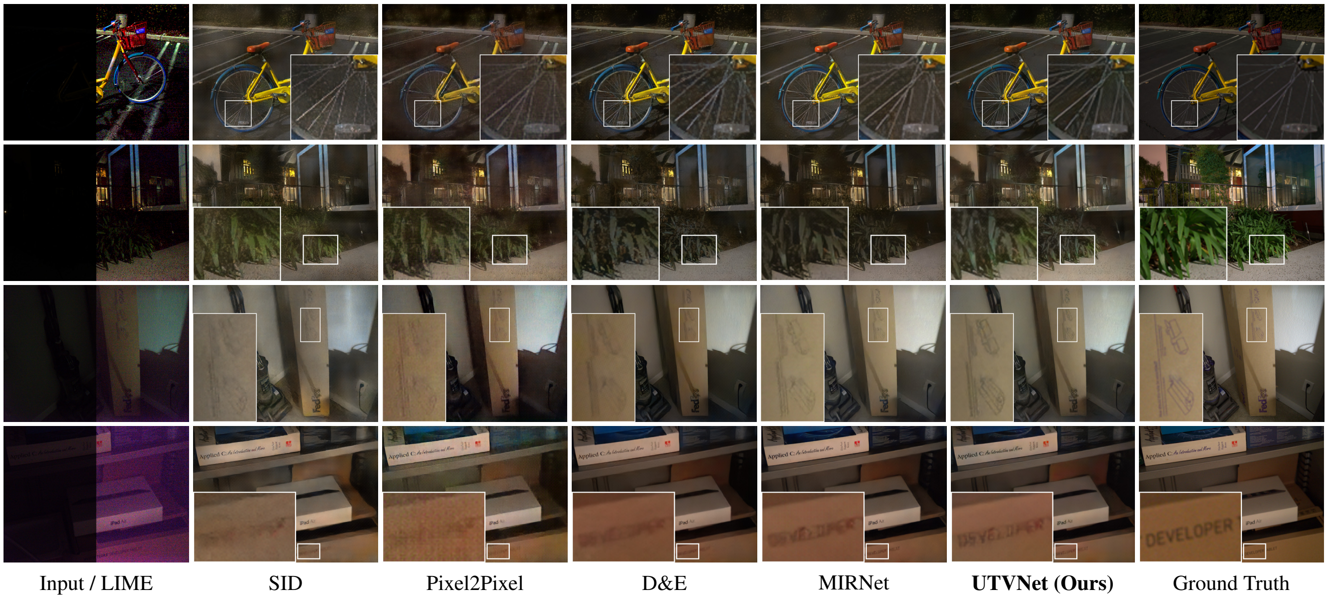


Figure 4. Visual results of different methods. UTVNet outperforms other methods in finer details preservation. *(Best viewed with zoom)*

Table 1. Quantitative comparison of different methods on sRGB-SID dataset. * indicates the released pre-trained model on the same dataset. The best result is highlighted.

Method	**PSNR**	**SSIM**	**L2$_{Lab}$**	**LPIPS**
LIME [1]	15.515	0.3011	14.040	0.5230
LIME [1] + NBNet [11]	16.280	0.3817	14.191	0.5087
NBNet [2] + LIME [1]	13.455	0.4416	15.920	0.6120
White-box [8]	15.941	0.4149	13.755	0.5569
White-box [8] + NBNet [11]	14.736	0.4938	13.624	0.5391
NBNet [2] + White-box [8]	13.875	0.4467	15.094	0.5964
DPED [37]	16.528	0.5144	12.841	0.4924
DeepUPE [9]	13.412	0.2565	12.411	0.5188
CSRNet [13]	19.613	0.5696	12.183	0.4779
DeepLPF [38]	19.645	0.6472	11.847	0.4207
Pixel2Pixel [39]	21.182	0.6073	11.626	0.4959
SID [16]	21.212	0.6601	10.998	0.4026
AGLLNet [3]	18.641	0.6158	13.649	0.4472
D&E* [4]	21.934	0.6950	10.859	0.3744
D&E [4]	22.157	0.6922	10.751	0.3794
MIRNet [5]	22.345	0.7031	10.750	0.3562
UTVNet (Ours)	**22.691**	**0.7179**	**10.581**	**0.3417**

with the sRGB-SID dataset proposed by [4]. The sRGB-SID dataset is generated from SID [16] raw dataset to model real-world noisy low-light sRGB images. It contains 4,198 image pairs for training and 1,196 image pairs for testing.

4.1. Results on sRGB-SID dataset

We compare the proposed UTVNet with 11 recent methods, including LIME [1], DPED [37], D&E [4], CSRNet [13], SID [16], DeepUPE [9], DeepLPF [38], White-Box [8], Pixel2Pixel [39], AGLLNet [3], MIRNet [5]. For Pixel2Pixel, SID, DeepLPF, CSRNet, D&E, MIRNet, we re-train their models for comparison. Because of the limitation on memory capacity, the number of features in the first layer of MIRNet is limited at 20. Meanwhile, the perceptual loss were adopted when training all the methods.

Quantitative and visual comparisons. We compare UTVNet with state-of-the-art methods in terms of PSNR, SSIM [40], Mean $L2$ error in L*a*b* space (lower is better) and LPIPS [41] (lower is better). The results are presented in Table 1. The first six rows of Table 1 show that the exposure correction methods provide unsatisfactory performance even when combined with the real-world denoising method NBNet [2]. Compared with all learning-based exposure correction methods and image restoration approaches, our UTVNet achieves the best performance in four metrics. The visual results of the proposed approach and the state-of-the-art methods can be seen in Fig. 4. One can see that only correcting the exposure by LIME may fail to recover images with noise, because it is hard to recover original details from small pixel values disrupted by noise. SID and Pixel2Pixel both introduce artifacts. Without noise level estimation, D&E and MIRNet may provide over-smooth or still noisy results. Our UTVNet outperforms all methods in recovering finer details and colors.

Computational cost. We analyze the computational complexity of some recently proposed methods and our UTVNet on sRGB-SID dataset with a resolution of 1024 × 768 pixels. The parameter size, MACs, execution time, and PSNR results are shown in Table 3. Compared with D&E and MIRNet, our UTVNet achieves the best performance with a balanced parameter size and lower MAC operations.

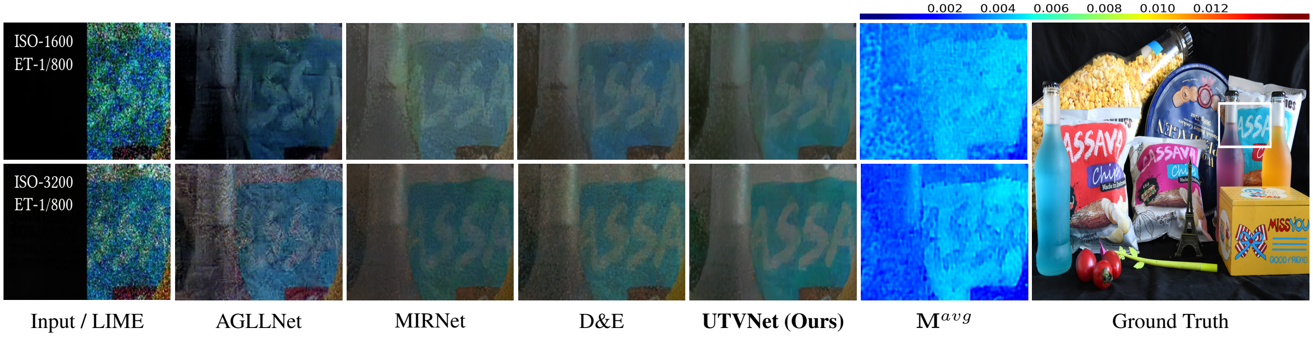


Figure 5. Evaluating the performance at different ISO levels (1600 vs. 3200) under exposure time (ET) $1/800s$ from the ELD dataset. M^{avg} is the approximated average of noise level maps in all iterations by the proposed UTVNet. *(Best viewed with zoom)*

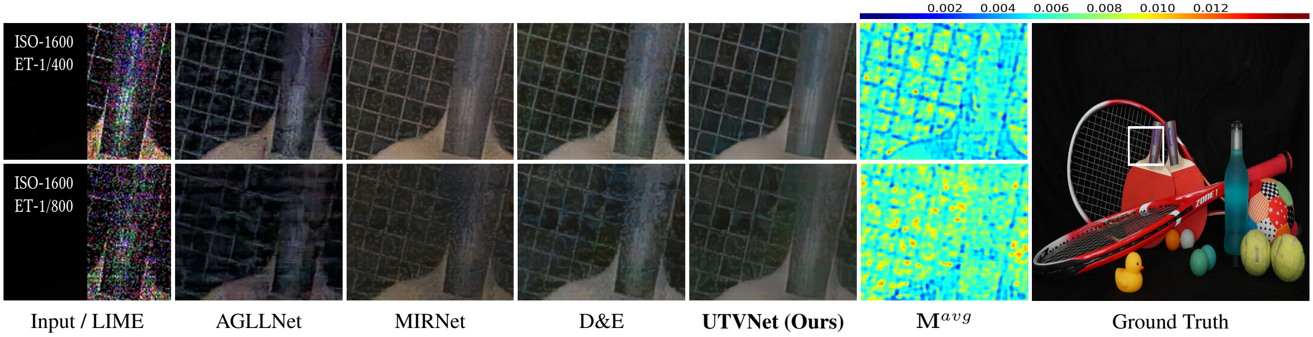


Figure 6. Evaluating the performance at different exposure times ($1/400s$ vs. $1/800s$) under ISO 1600 from the ELD dataset. M^{avg} is the approximated average of noise level maps in all iterations by the proposed UTVNet. *(Best viewed with zoom)*

Table 2. Quantitative results of different methods that trained on sRGB-SID or other synthetic datasets and tested on ELD dataset.

Method	**SonyA7S2**			**NikonD850**			**CanonEOS700D**		
	PSNR	**SSIM**	**LPIPS**	**PSNR**	**SSIM**	**LPIPS**	**PSNR**	**SSIM**	**LPIPS**
LIME [1]	16.467	0.3960	0.5384	15.155	0.3034	0.5616	15.423	0.3177	0.5854
AGLLNet [3]	17.274	0.4915	0.4653	16.047	0.4564	0.5072	16.399	0.3569	0.6145
MIRNet [5]	17.301	0.5103	0.4643	17.692	0.5045	0.5020	16.201	0.3778	0.5926
D&E [4]	18.571	0.5783	0.4612	17.926	0.5532	0.4536	**16.639**	0.3729	0.5868
UTVNet (Ours)	**18.675**	**0.6054**	**0.4524**	**18.161**	**0.5910**	**0.4446**	16.496	**0.4233**	**0.5601**

4.2. Performance in Real Captured Scenes

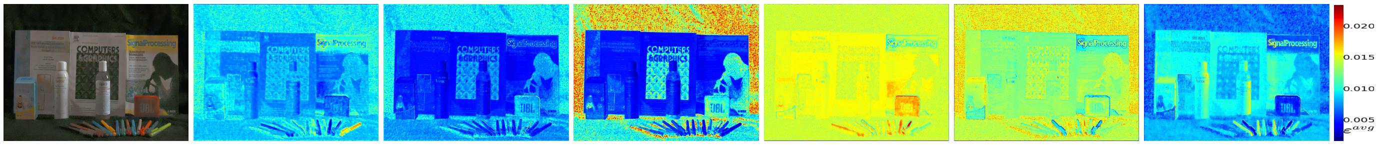
To substantiate the robustness in real captured low-light scenes of proposed UTVNet, we evaluate the performance of different methods with ELD [6] dataset. The extremely low-light images in ELD dataset covers 10 indoor scenes and were taken by 4 camera devices. In each scene, the noisy images were captured with different ISO levels and exposure times to guarantee noise and low-light conditions. Only the sRGB images were used in our experiment. We compare UTVNet with LIME, AGLLNet, MIRNet, and D&E. We adopted these module that were trained on other datasets to restore the images in the ELD dataset, where MIRNet, D&E, and our UTVNet were trained on the sRGB-SID dataset, AGLLNet was trained on the synthetic low-light dataset with Gaussian-Poisson mixed noise model.

Quantitative and visual comparisons. The quantitative results are shown in Table 2. Our UTVNet obtained competitive performance on PSNR, SSIM, and LPIPS scores.

In visual comparison, we visualize the average noise level map M^{avg} of all iterations in UTVNet. First, we evaluate the results under different ISO levels in Fig. 5. The outputs of AGLLNet suffer from artifacts. D&E achieves better color quality at ISO 1600 but is still noisy. When increasing the ISO to 3200, guided by the estimated noise level map, our UTVNet generates clearer result than the outputs of MIRNet and D&E. Next, we compare the visual results at different exposure times as shown in Fig. 6. The color is not recovered correctly by MIRNet. When reducing the exposure time from $1/400$ to $1/800$, fewer photons can be gathered by the camera. The proposed UTVNet introduces fewer artifacts than the state-of-the-art methods.

4.3. Analysis and Discussions

Ablation study. The results of the ablation study are shown in Table 4. The first two rows in Table 4 demonstrate the importance of the noise level map M. To estimate the

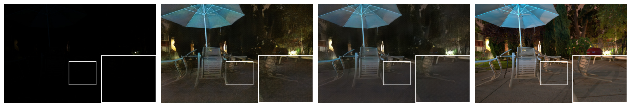


Output $\mathbf{M}_1$ $\mathbf{M}_2$ $\mathbf{M}_3$ $\mathbf{M}_6$ $\mathbf{M}_7$ $\mathbf{M}_8$

Figure 7. We visualize the average noise level maps of R, G, B channels in the first three and the last three iterations respectively. ε^{avg} in the color coordinate axis is the average global noise variance. *(Best viewed with zoom)*

Table 3. Analysis of computational cost. Our UTVNet achieves SOTA performance with a balanced computational requirement.

Method	Params (M)	Time (s)	MACs (G)	PSNR (dB)
DeepLPF	1.716	0.154	49.930	19.645
Pixel2Pixel	11.38	0.047	217.71	21.182
SID	7.760	0.060	164.62	21.212
D&E	8.621	0.234	417.36	22.157
MIRNet	3.107	0.756	929.43	22.345
UTVNet	7.745	0.199	273.76	22.691



Input D&E UTVNet Ground Truth

Figure 8. The example of failure case when restoring images.

effectiveness of the NLI-Block, we first remove the global noise variance ε, so that parameter $\mathbf{M}$ is directly learned from the convolution layers in the NLI-Block, which cause a performance drop as shown in the third row of Table 4. Because the pixel value is small in extremely low-light images, modifying the global noise variance is a better way to estimate the differences in noise levels among pixels. Similarly, replacing $\mathcal{A}(\cdot)$ with ReLU also causes a performance drop. The results verify the effectiveness of the way to approximate the noise level map and demonstrate that $\mathcal{A}(\cdot)$ provides a larger noise level value for guaranteeing the role of $\mathbf{M}$ as the balancing parameter.

Analysis of the unfolding TV module. We investigate the estimation of the noise level map in each iteration. The average noise level maps of three R, G, B channels are calculated for visualization in Fig. 7. In most of the iterations, the noise level values of sharp edges or object boundaries is lower than that of other regions, which means when smoothing the regions containing edges and boundaries, the output is mainly determined by the fidelity term. Such noise level maps are essential for noise suppression, as they guarantee the detail preservation after noise suppression, which is the main reason that our UTVNet can recover finer details compared with the state-of-the-art methods. We also analyze the iteration times, with results shown in Table 4. For a speed-accuracy trade-off, we adopt UTVNet with 8 iterations.

Table 4. Ablation study and execution time with different numbers of iteration on sRGB-SID dataset.

Method	PSNR	SSIM	LPIPS	Time(s)
w/o $\mathbf{M}$	21.858	0.6938	0.3519	-
$\mathbf{M} \to \varepsilon$	21.483	0.6930	0.3547	-
w/o ε in NLI-Block	21.909	0.6974	0.3511	-
$\mathcal{A}(\cdot) \to$ ReLU	22.136	0.7050	0.3432	-
$k = 4$	22.476	0.6897	0.3479	**0.159**
$k = 6$	22.539	0.6985	0.3422	0.171
$k = 10$	22.647	**0.7203**	0.3379	0.220
$k = 12$	22.521	0.7124	**0.3368**	0.246
UTVNet ($k = 8$)	**22.691**	0.7179	0.3417	0.199

Limitation. As shown in Fig. 8, our UTVNet as well as other state-of-the-art approaches may fail to restore the details with similar color as the background, and introduce artificial halos. Such a problem is more difficult to tackle in sRGB color space compared with raw Bayer space.

5. Conclusion

In this paper, we propose an adaptive unfolding total variation network, referred to as UTVNet, for real-world low-light noisy image enhancement in the sRGB domain. Inspired by the relationship between the noise level and the balancing parameter in model-based denoising methods, we formulate an adaptive total variation regularization by introducing a learnable noise level map that serves as the balancing parameters for each pixel to control the trade-off between noise reduction and detail preservation. The TV minimization problem is unfolded with fidelity and smoothness constraints to learn the noise level map in an end-to-end manner, and generating a noise-free smooth layer. Furthermore, we also design a noise level initialization block to predict noise level maps in a coarse-to-fine way before the unfolding TV architecture. Extensive experimental results have verified the capability of the proposed UTVNet. We expect future work to make further improvements in image quality, especially in real captured low-light scenes.

Acknowledgments. This work is supported by Ministry of Science and Technology China (MOST) Major Program on New Generation of Artificial Intelligence 2030 No. 2018AAA0102200. It is also supported by Natural Science Foundation China (NSFC) Major Project No. 61827814 and Shenzhen Science and Technology Innovation Commission (SZSTI) Project No. JCYJ20190808153619413.

References

[1] X. Guo, Y. Li, and H. Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on Image Processing*, 26(2):982–993, 2017. 1, 2, 6, 7

[2] Shen Cheng, Yuzhi Wang, Haibin Huang, Donghao Liu, Haoqiang Fan, and Shuaicheng Liu. Nbnet: Noise basis learning for image denoising with subspace projection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4896–4906, June 2021. 1, 2, 6

[3] Feifan Lv, Yu Li, and Feng Lu. Attention guided low-light image enhancement with a large scale low-light simulation dataset. *International Journal of Computer Vision*, 129(7):2175–2193, 2021. 1, 2, 6, 7

[4] K. Xu, X. Yang, B. Yin, and R. W. H. Lau. Learning to restore low-light images via decomposition-and-enhancement. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2278–2287, 2020. 1, 2, 6, 7

[5] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *ECCV*, 2020. 1, 2, 6, 7

[6] Kaixuan Wei, Ying Fu, Jiaolong Yang, and Hua Huang. A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 7

[7] Samuel W 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. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016. 1

[8] Y. Hu, H. He, C. Xu, and and S. Lin B. Wang. Exposure: A white-box photo post-processing framework. *ACM Trans. Graph.*, 37(2), May 2018. 1, 6

[9] R. Wang, Q. Zhang, C. Fu, X. Shen, W. Zheng, and J. Jia. Underexposed photo enhancement using deep illumination estimation. pages 6842–6850, 2019. 1, 2, 6

[10] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 1, 2, 3, 5

[11] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang. Toward convolutional blind denoising of real photographs. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1712–1722, 2019. 1, 2, 3, 6

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2, 5

[13] Jingwen He, Yihao Liu, Yu Qiao, and Chao Dong. Conditional sequential modulation for efficient global image retouching. In *European Conference on Computer Vision*, pages 679–695. Springer, 2020. 2, 6

[14] M. Gharbi, andJ. T. Barron J. Chen, S. W. Hasinoff, and F. Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017. 2

[15] Zhenqiang Ying, Ge Li, Yurui Ren, Ronggang Wang, and Wenmin Wang. A new low-light image enhancement algorithm using camera response model. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017. 2

[16] C. Chen, Q. Chen, J. Xu, and V. Koltun. Learning to see in the dark. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3291–3300, 2018. 2, 6

[17] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2013. 2

[18] J. Xu, L. Zhang, D. Zhang, and X. Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. pages 1105–1113, 2017. 2

[19] S. Nam, Y. Hwang, Y. Matsushita, and S. J. Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. pages 1683–1691, 2016. 2

[20] A. Buades, B. Coll, and J. . Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65 vol. 2, 2005. 2

[21] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. 2

[22] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, 2014. 2

[23] Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1-2):165–199, 2017. 3

[24] R. Liu, Z. Jiang, X. Fan, and Z. Luo. Knowledge-driven deep unrolling for robust image layer separation. *IEEE Transactions on Neural Networks and Learning Systems*, 31(5):1653–1666, 2020. 3

[25] R. Liu, S. Cheng, L. Ma, X. Fan, and Z. Luo. Deep proximal unrolling: Algorithmic framework, convergence analysis and applications. *IEEE Transactions on Image Processing*, 28(10):5013–5026, 2019. 3

[26] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Transactions on Image Processing*, 19(9):2345–2356, 2010. 3

[27] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu. Denoising prior driven deep neural network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(10):2305–2318, 2019. 3

[28] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 3

[29] K. Zhang, L. Van Gool, and R. Timofte. Deep unfolding network for image super-resolution. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3214–3223, 2020. 3, 4

[30] Robert Acar and Curtis R Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse problems*, 10(6):1217, 1994. 3

[31] Yilun Wang, Junfeng Yang, Wotao Yin, and Yin Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008. 3, 4

[32] Kai-Fu Yang, Xian-Shi Zhang, and Yong-Jie Li. A biological vision inspired framework for image enhancement in poor visibility conditions. *IEEE Transactions on Image Processing*, 29:1493–1506, 2020. 3

[33] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen. An augmented lagrangian method for total variation video restoration. *IEEE Transactions on Image Processing*, 20(11):3097–3111, 2011. 4

[34] John Immerkær. Fast noise variance estimation. *Computer Vision and Image Understanding*, 64(2):300–302, 1996. 4

[35] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll. Burst denoising with kernel prediction networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2502–2510, 2018. 5

[36] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[37] 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, 2017. 6

[38] S. Moran, P. Marza, S. McDonagh, S. Parisot, and G. Slabaugh. Deeplpf: Deep local parametric filters for image enhancement. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12823–12832, 2020. 6

[39] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 6

[40] W. Zhou , 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

[41] Richard R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 6