

Retinex-inspired Unrolling with Cooperative Prior Architecture Search for Low-light Image Enhancement

Risheng Liu^{1,3*}, Long Ma^{2,3}, Jiaao Zhang^{2,3}, Xin Fan^{1,3}, Zhongxuan Luo^{2,3}

¹International School of Information Science & Engineering, Dalian University of Technology

²School of Software Technology, Dalian University of Technology

³Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province

{rsliu, xin.fan, zxlue}@dlut.edu.cn, {longma, jiaaozhang}@mail.dlut.edu.cn

Abstract

Low-light image enhancement plays very important roles in low-level vision areas. Recent works have built a great deal of deep learning models to address this task. However, these approaches mostly rely on significant architecture engineering and suffer from high computational burden. In this paper, we propose a new method, named Retinex-inspired Unrolling with Architecture Search (RUAS), to construct lightweight yet effective enhancement network for low-light images in real-world scenario. Specifically, building upon Retinex rule, RUAS first establishes models to characterize the intrinsic underexposed structure of low-light images and unroll their optimization processes to construct our holistic propagation structure. Then by designing a cooperative reference-free learning strategy to discover low-light prior architectures from a compact search space, RUAS is able to obtain a top-performing image enhancement network, which is with fast speed and requires few computational resources. Extensive experiments verify the superiority of our RUAS framework against recently proposed state-of-the-art methods. The project page is available at <http://dutmedia.org/RUAS/>.

1. Introduction

High quality images are critical to a large amount of computer vision and machine learning applications, such as object detection [18], tracking [19], and segmentation [42], just name a few. Unfortunately, images captured in low-light environments usually suffer multiple degradations, including but not limited to poor visibility, low contrast and unexpected noise. Therefore, it is necessary to enhance low-light images before further processing and analysis. Indeed, Low-light Image Enhancement (LIE) techniques have

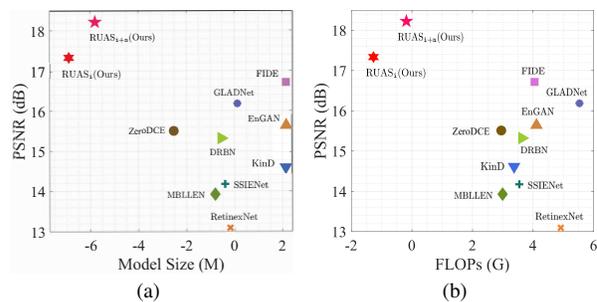


Figure 1. Averaged quantitative performance (PSNR) vs model size (a), FLOPs (b) on LOL dataset [2]. We plot results of our methods (i.e. RUAS_i and RUAS_{i+n}) and some recently proposed state-of-the-art CNN-based approaches, including MBLLEN [23], GLADNet [36], RetinexNet [2], EnGAN [12], SSIENet [48], KinD [49], ZeroDCE [8], FIDE [37], and DRBN [40]. Here a log scale is used on the x-axis for illustration.

gained a lot of traction recently and generally fall into two major categories: classical and deep learning methods.

In the past decades, classical LIE methods often perform histogram equalization [28, 4, 43] or gamma correction [11, 29, 35] to enhance low-light images. There are also various classical methods consider Retinex theory [26] and introduce different prior regularized optimization models to characterize the structures of the illumination and reflectance image layers [6, 9, 13, 46, 20]. However, these hand-crafted constraints/priors are not adaptive enough and their results may present intensive noises and/or suffer from over- and under- enhancement.

In recent years, great progress has been made on designing CNN-based models for LIE problems. Among these well-known approaches, most CNN-based solutions rely on paired data for supervised training [2, 49, 21, 37]. In addition, dozens of methods also train their networks without paired supervision [40, 12, 8]. However, the performances of these deep learning methods heavily rely on their elaborately designed architectures and carefully selected paired/unpaired training data. Moreover, most of these

*Corresponding author.

existing CNN-based methods tend to obtain unsatisfactory visual results in the face of various light intensities and intensive noises under the real-world scenario. The main reason is that these relevant approaches lack of physical constraints in principle, which makes it difficult to capture the inherent weak light image structure.

To partially address the above issues, we propose a principled framework to construct enhancement networks by infusing knowledge of low-light images and searching lightweight prior architectures, named Retinex-inspired Unrolling with Architecture Search (RUAS). More concretely, taking Retinex rule into consideration, we first design optimization models to exploit the latent structures of the low-light image in the real-world noisy scenario. Then by unrolling the corresponding optimization processes, we establish the holistic propagation structure of our enhancement network. Finally, we provide a reference-free bilevel learning strategy to cooperatively search prior architectures for the illumination map and desired image. Our contributions can be summarized as follows:

- In contrast to existing CNN-based LIE methods requiring substantial efforts to heuristically design the whole neural network, RUAS first provides a principled manner to build our fundamental network structure and then automatically discover the embedded atomic prior architectures.
- We develop a cooperative bilevel search strategy for RUAS, which is able to simultaneously discover architectures from a compact search space for both illumination estimation and noise removal. Furthermore, our strategy does not require any paired/unpaired supervisions during the whole search process.
- RUAS offers flexibility in searching prior architectures for different kinds of low-light scenarios. Extensive experiments also show that our established enhancement networks are memory and computation efficient, and can perform favorably against state-of-the-art approaches (see Fig. 1).

2. Related Work

2.1. CNNs for Low-light Image Enhancement

In recent years, with the development of deep learning, the problem of LIE has achieved a significant performance boost. Actually, there has been a great deal of interest in CNN architectures designed for solving the LIE problem. For example, LLNet [22] utilized a variant of the stacked sparse denoising autoencoder to brighten the low-light images. EnGAN [12] designed an attention module on U-Net [27] and can be trained with only low/normal-light images (unnecessarily paired). The paper in [37] developed

a frequency-based decomposition-and-enhancement network based on the attention to context encoding module. RetinexNet [2] combined the Retinex theory with CNNs to estimate the illumination map and enhance the low-light images. KinD [49] designed a similar network but connected the feature-level illumination and reflectance in the decomposition step. Wang *et al.* [34] designed an image-to-illumination network architecture based on the bilateral learning framework. Zhang *et al.* [48] established a self-supervised CNN to simultaneously output the illumination and reflectance. The work in [8] proposed a zero-reference curve estimation CNN to address the LIE task. A recursive band network was proposed in [40] and trained by a semi-supervised strategy. Nonetheless, discovering state-of-the-art neural network architectures requires substantial efforts.

2.2. Neural Architecture Search (NAS)

In past years, NAS has attracted increasing attention due to the potential of finding effective and/or efficient architectures that outperform human expertise. Early attempts [50, 30] use evolutionary algorithms to optimize architectures and parameters of the networks. Another group of approaches [3, 31] employ reinforcement learning techniques to train a meta-controller to generate potential architectures. However, these two kinds of methods require a large amount of computations, resulting in inefficient search strategies. Recently, differentiable search methods [16, 39, 32] formulated the super-network into a differentiable form with respect to a set of architectural parameters, so that they can be optimized by gradient descent and the search cost can be reduced to several hours. The above search algorithms have achieved highly competitive performance in various high-level vision tasks, including classification [39], detection [33], and segmentation. Very recently, NAS algorithms [24, 44, 14] have also been applied to low-level vision problems, such as image denoising, restoration, and deraining, etc. Unfortunately, existing NAS strategies are fully data-driven, thus require a large number of well-prepared paired training data, which is generally impractical for the LIE problem. Furthermore, due to the lack of principled prior knowledge, architectures searched with the aforementioned methods are still not suitable to exploit the complex statistics of low-light images.

3. The Proposed Method

We first establish the enhancement network by unrolling the optimization process of Retinex-inspired models. Then we introduce a distillation cell based search space for the prior modules. Finally, a cooperative bilevel search strategy is proposed to discover desired architectures for illumination estimation and noise removal.

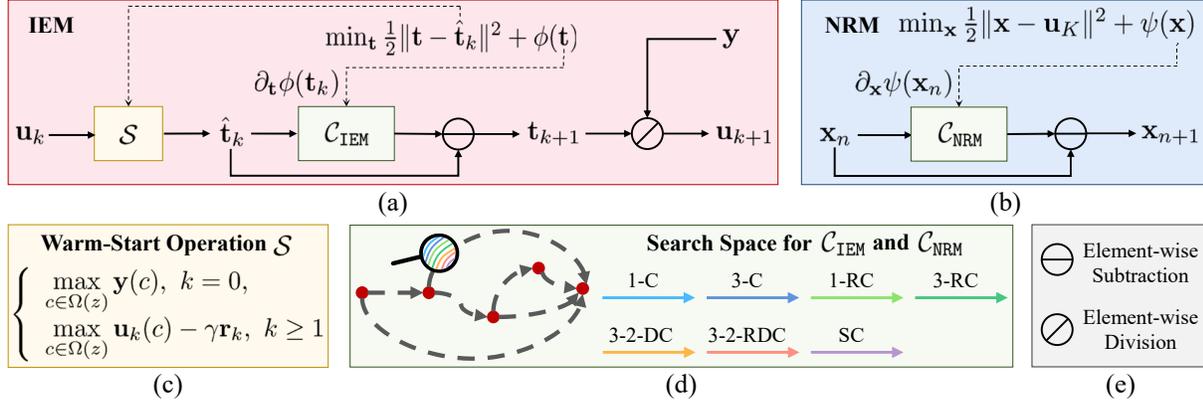


Figure 2. Illustrations of the fundamental components for RUAS. On the top row, we plot the diagrams of IEM (a) and NRM (b). On the bottom row, we show the warm-start operation (c), the search space for \mathcal{C}_{IEM} and \mathcal{C}_{NRM} (d) and two element-wise operations (e).

3.1. Retinex-Inspired Optimization Unrolling

Our RUAS enhancement network is built upon the following simple Retinex rule $\mathbf{y} = \mathbf{x} \otimes \mathbf{t}$, where \mathbf{y} and \mathbf{x} are the captured underexposed observation and the desired recovery, respectively. Furthermore, \mathbf{t} denotes the illumination map and the operator \otimes represents the element-wise multiplication. As shown in Fig. 2 (a) and (b), Illumination Estimation Module (IEM) is devised to estimate the illumination map \mathbf{t} and Noise Removal Module (NRM) is designed to suppress noise in some challenging low-light scenario. We next detail these two key components.

3.1.1 Illumination Estimation Module (IEM)

Define an intermediate image \mathbf{u} . At k -th stage of IEM, we first propose the following strategy to estimate an initial illumination map $\hat{\mathbf{t}}_k$, i.e., $\hat{\mathbf{t}}_k = \mathcal{S}(\mathbf{u}_k)$ with

$$\mathcal{S}(\mathbf{u}_k) := \begin{cases} \max_{c \in \Omega(z)} \mathbf{y}(c), & k = 0, \\ \max_{c \in \Omega(z)} \mathbf{u}_k(c) - \gamma \mathbf{r}_k, & k \geq 1. \end{cases} \quad (1)$$

Here \mathbf{u}_k is obtained by $\mathbf{u}_k = \mathbf{y} \oslash \mathbf{t}_k$, where \mathbf{t}_k is the estimated illumination map in the last stage and \oslash denotes the element-wise division. Furthermore, $\Omega(z)$ is a region centered at pixel z and c is the location index within this region (for three color channels). The principle behind this term is that the illumination is at least the maximal value of a certain location and can be used to handle non-uniform illuminations. As for the residual $\mathbf{r}_k = \mathbf{u}_k - \mathbf{y}$ (with penalty parameter $0 < \gamma \leq 1$), we actually introduce this term to adaptively suppress some overexposed pixels for $\hat{\mathbf{t}}_k$ during the propagation.

With the illumination warm-start $\hat{\mathbf{t}}_k$, we further refine \mathbf{t} by solving the following model inspired by the work in [9]: $\min_{\mathbf{t}} \frac{1}{2} \|\mathbf{t} - \hat{\mathbf{t}}_k\|^2 + \phi(\mathbf{t})$, where $\phi(\cdot)$ represents a regularization term of \mathbf{t} . Different from classical iterative optimization methods, which interact with the prior term directly, we

just write a schematic gradient descent scheme¹

$$\mathbf{t}_{k+1} = \hat{\mathbf{t}}_k - \partial_{\mathbf{t}} \phi(\mathbf{t}_k), \quad (2)$$

and parameterize $\partial_{\mathbf{t}} \phi(\mathbf{t}_k)$ by a CNN architecture $\mathcal{C}_{\text{IEM}}(\mathbf{t}_k)$. By performing K stages of the above calculations, we can obtain $\mathbf{u}_K = \mathbf{y} \oslash \mathbf{t}_K$ as the output of IEM. Indeed, the choice of parameterizing each iteration as a separate CNN offers tremendous flexibility. We will demonstrate how to discover proper architectures for the above optimization process at the end of this section.

3.1.2 Noise Removal Module (NRM)

It has been recognized that intensive noise in underexposed images cannot be simply removed by pre-/post-processing with existing denoising methods. Therefore, we intend to introduce another optimization unrolling module (NRM) to suppress noises in real-world low-light scenarios. Similar to IEM, we define a regularized model: $\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{u}_K\|^2 + \psi(\mathbf{x})$, where ψ denotes the prior regularization on \mathbf{x} . By adopting the same unrolling strategy used in IEM, we can update our desired image \mathbf{x} by

$$\mathbf{x}_{n+1} = \mathbf{u}_K - \partial_{\mathbf{x}} \psi(\mathbf{x}_n). \quad (3)$$

Here we write $\mathcal{C}_{\text{NRM}}(\mathbf{x}_n)$ as the parameterization (i.e., CNN architecture) of $\partial_{\mathbf{x}} \psi(\mathbf{x}_n)$ and denote the output of NRM (with N stages) as \mathbf{x}_N in parallel.

3.2. Cooperative Architecture Search

In this part, we present a new search strategy to cooperatively discover architectures for both IEM and NRM.

3.2.1 Compact Search Space for Low-light Priors

We start with defining the search space for low-light prior modules (\mathcal{C}_{IEM} and \mathcal{C}_{NRM}). By employing feature distillation

¹Please notice that here we just skip the learning rate (i.e., set it as 1).

techniques [17], we define our search space as a distillation cell, which is a directed acyclic graph with five nodes and each node connects to the next and the last nodes (see Fig. 2 (d)). In fact, each node in the cell is a latent representation and each direct edge is associated with some operation. The connection to the last node just realizes the feature information distillation. The candidate operations include 1×1 and 3×3 Convolution (1-C and 3-C), 1×1 and 3×3 Residual Convolution (1-RC and 3-RC), 3×3 Dilation Convolution with dilation rate of 2 (3-2-DC), 3×3 Residual Dilation Convolution with dilation rate of 2 (3-2-RDC), and Skip Connection (SC). By adopting the continuous relaxation technique used in differentiable NAS literature [16, 39, 15], we introduce the vectorized form $\alpha = \{\alpha_t, \alpha_n\}$ to encode architectures in our search space (denoted as \mathcal{A}) for \mathcal{C}_{IEM} and \mathcal{C}_{NRM} , respectively. Denote by $\omega = \{\omega_t, \omega_n\}$ the weight parameters associated with the architecture α . Then the search task reduces to jointly learn α and ω within all the mixed operations.

3.2.2 Differentiable Search with Cooperation

The above search space can make our entire framework differentiable to both layer weights ω and hyper-parameters α , so that the most straightforward idea is to apply gradient-based NAS approaches for our problem. However, these classical methods can only learn ω and α in an end-to-end fashion, which completely ignore the important light enhancement factors (e.g., illuminations and noises). Our main idea to address this issue is to discover architectures that can properly reveal low-light prior information for underexposed images in real-world noisy scenarios. This is achieved by searching architectures for IEM and NRM by cooperation. Specifically, we formulate the search process of these two modules as a cooperative game and aim to solve the following model for α_t (IEM) and α_n (NRM):

$$\min_{\alpha_n \in \mathcal{A}} \left\{ \min_{\alpha_t \in \mathcal{A}} \mathcal{L}_{\text{val}}(\alpha_t, \alpha_n; \omega_t^*, \omega_n^*) \right\}. \quad (4)$$

We denote \mathcal{L}_{val} as a cooperative loss on the validation dataset, i.e.,

$$\mathcal{L}_{\text{val}} := \mathcal{L}_{\text{val}}^t(\alpha_t; \omega_t^*) + \beta \mathcal{L}_{\text{val}}^n(\alpha_n(\alpha_t); \omega_n^*), \quad (5)$$

where $\mathcal{L}_{\text{val}}^t$ and $\mathcal{L}_{\text{val}}^n$ respectively denote the losses on IEM and NRM and $\beta \geq 0$ is a trade-off parameter. Since NRM is defined based on the output of IRM (see Fig. 2), here we should also consider α_t as parameters of α_n in $\mathcal{L}_{\text{val}}^n$. In fact, by analogy with the generative adversarial learning task [7], it should be understood that the optimization problem in Eq. (4) actually considers a cooperative (“min-min”), rather than an adversarial (“min-max”) objective.

As for ω_t^* (and ω_n^*), we assume that they are only associated with the architecture α_t (and α_n). That is, they can

Algorithm 1 Cooperative Architecture Search Strategy

Input: The search space \mathcal{A} , the training and validation datasets \mathcal{D}_{tr} and \mathcal{D}_{val} and necessary parameters.

Output: The searched architecture of RUAS.

- 1: Initialize $\alpha = \{\alpha_t, \alpha_n\}$ and $\omega = \{\omega_t, \omega_n\}$.
 - 2: **while** not converged **do**
 - 3: // Update α_t and ω_t for IEM.
 - 4: **while** not converged **do**
 - 5: $\alpha_t^\dagger \leftarrow \alpha_t - \nabla_{\alpha_t} \mathcal{L}_{\text{val}}^t(\alpha_t; \omega_t - \nabla_{\omega_t} \mathcal{L}_{\text{tr}}^t) - \beta \nabla_{\alpha_t} \mathcal{L}_{\text{val}}^n(\alpha_n(\alpha_t); \omega_n)$.
 - 6: $\omega_t^\dagger \leftarrow \omega_t - \nabla_{\omega_t} \mathcal{L}_{\text{tr}}^t(\omega_t; \alpha_t^\dagger)$.
 - 7: **end while**
 - 8: // Update α_n and ω_n for NRM.
 - 9: **while** not converged **do**
 - 10: $\alpha_n^\dagger \leftarrow \alpha_n - \nabla_{\alpha_n} \mathcal{L}_{\text{val}}^n(\alpha_n(\alpha_t^\dagger); \omega_n - \nabla_{\omega_n} \mathcal{L}_{\text{tr}}^n)$.
 - 11: $\omega_n^\dagger \leftarrow \omega_n - \nabla_{\omega_n} \mathcal{L}_{\text{tr}}^n(\omega_n; \alpha_n^\dagger)$.
 - 12: **end while**
 - 13: **end while**
 - 14: **return** Architecture derived based on α_t^* and α_n^* .
-

be obtained by minimizing the following models

$$\begin{cases} \omega_t^* = \arg \min_{\omega_t} \mathcal{L}_{\text{tr}}^t(\omega_t; \alpha_t), \\ \omega_n^* = \arg \min_{\omega_n} \mathcal{L}_{\text{tr}}^n(\omega_n; \alpha_n), \end{cases} \quad (6)$$

where $\mathcal{L}_{\text{tr}}^t$ and $\mathcal{L}_{\text{tr}}^n$ are the training losses for IEM and NRM, respectively.

Therefore, our search strategy implies a bilevel optimization problem with Eq. (4) and Eq. (6) as the upper-level and lower-level subproblems, respectively. Moreover, the upper-level subproblem in Eq. (4) should be further separated into two cooperative tasks during the search process.

3.2.3 Reference-free Bilevel Learning

We first specify our training and validation objectives based on a series of reference-free losses. Specifically, for IEM, we define a loss $\frac{1}{2} \|\text{net}_{\alpha_t, \omega_t}(\mathbf{y}) - \hat{\mathbf{t}}_0\|^2 + \eta_t \text{RTV}(\text{net}_{\alpha_t, \omega_t}(\mathbf{y}))$ on the training and validation dataset as $\mathcal{L}_{\text{tr}}^t$ and $\mathcal{L}_{\text{val}}^t$, respectively. Here the first term is the fidelity and $\text{RTV}(\cdot)$ denotes the relative total variation term [38] (with a parameter $\eta_t > 0$). In fact, this loss encourages IEM to output illuminations that can simultaneously preserve the overall structure and smooth the textural details. As for NRM, we introduce a similar loss $\frac{1}{2} \|\text{net}_{\alpha_n, \omega_n}(\mathbf{u}_K) - \mathbf{u}_K\|^2 + \eta_n \text{TV}(\text{net}_{\alpha_n, \omega_n}(\mathbf{u}_K))$ to define $\mathcal{L}_{\text{tr}}^n$ and $\mathcal{L}_{\text{val}}^n$, in which we utilize standard total variation $\text{TV}(\cdot)$ as our regularization [25] (with a parameter $\eta_n > 0$).

Then Alg. 1 summarizes the overall search process². It can be seen that IEM and NRM are searched alternative-

²Here we skip some regular numerical parameters (e.g., initialization, learning rate and stopping criterion) to simplify our notations.

Table 1. Quantitative results (PSNR, SSIM, LPIPS) of state-of-the-art methods and ours on the MIT-Adobe 5K and LOL datasets. The best result is in red whereas the second best one is in blue.

Datasets	Metrics	LIME	SDD	MBLLEN	GALDNet	RetinexNet	EnGAN	SSINet	KinD	DeepUPE	ZeroDCE	FIDE	DRBN	Ours
MIT	PSNR	17.788	17.617	15.587	16.728	12.685	15.014	10.324	17.169	18.779	16.463	17.170	15.954	20.830
	SSIM	0.826	0.792	0.713	0.764	0.644	0.768	0.620	0.696	0.822	0.764	0.696	0.704	0.854
	LPIPS	0.162	0.628	0.307	0.693	0.338	0.209	0.323	0.229	0.183	0.193	0.324	0.292	0.141
LOL	PSNR	14.916	15.484	13.931	16.188	13.096	15.644	14.176	14.616	13.041	15.512	16.718	15.324	18.226
	SSIM	0.516	0.578	0.489	0.605	0.429	0.578	0.534	0.636	0.483	0.553	0.673	0.699	0.717
	LPIPS	0.746	0.206	0.697	0.205	0.864	0.647	0.675	0.463	0.677	0.718	0.510	0.362	0.354

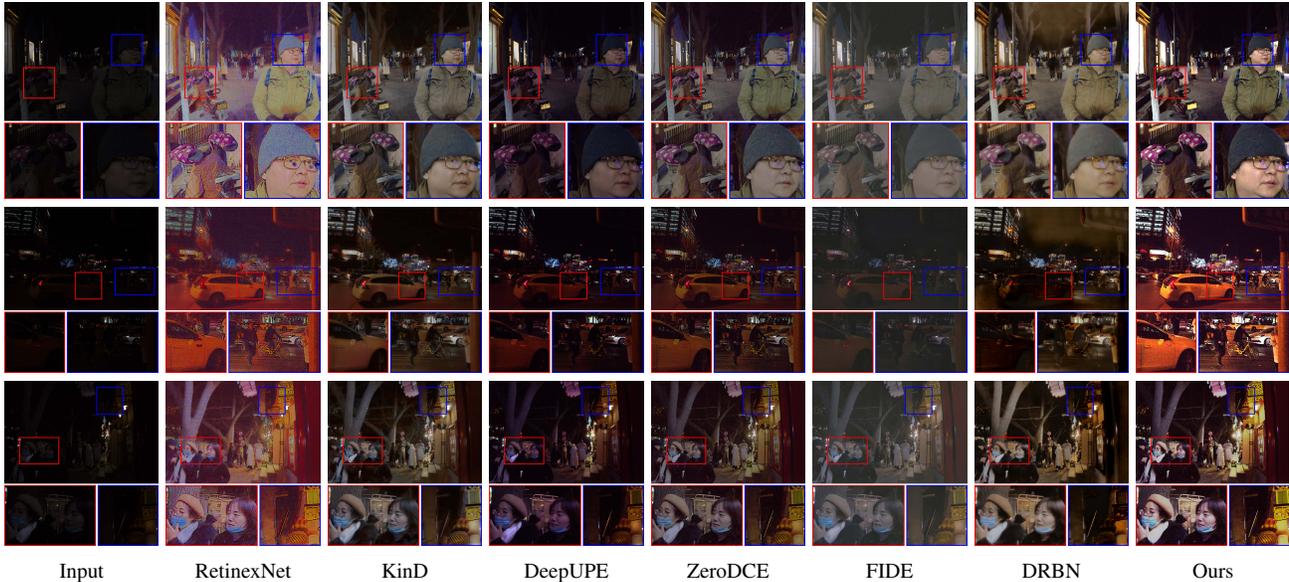


Figure 3. Visual results of state-of-the-art methods and ours on the DarkFace dataset. Red boxes indicate the obvious differences.

ly and simultaneously. That is, we update α_t (with the current α_n) for IEM (Steps 4-7) and update α_n (based on the updated α_t) for NRM (Steps 9-12). As for each module, we just adopt the widely used one-step finite difference technique [5] to approximately calculate gradients for the upper-level variables (Steps 5 and 10).

4. Experimental Results

4.1. Implementation Details

We sampled 500 low-light images randomly from the MIT-Adobe 5K [1] for searching and training, and sampled 100 image pairs for testing. As for the LOL Dataset [2], 100 image pairs were randomly sampled for evaluating and the remaining 689 pairs for searching and training. We adopted PSNR, SSIM and LPIPS [47] as our evaluated metrics. We also evaluated visual performance in the DarkFace [41] and ExtremelyDarkFace (used as the sub-challenge in the CVPR 2020 UG2+Challenge³) datasets. All the experiments are performed on a PC with a single TITAN X GPU.

³<http://cvpr2020.ug2challenge.org>

In the prior architecture search phase, we consider the same search space (with 3 fundamental cell structures) for IEM and NRM, but define their cells with different channel widths (i.e., 3 for IEM and 6 for NRM). The gradients of the architecture and weight parameters are computed following standard differential NAS techniques [16]. As for the numerical parameters, we set the maximum epoch as 20, the batch size as 1, and chose the initial learning rate as 3×10^{-4} . The momentum parameter was randomly sampled from (0.5, 0.999) and the weight decay was set as 10^{-3} . As for the training phase (with searched architecture), we use the same training losses as that in the search phase.

For the search time, our RUAS only spends 5 hours on a single TITAN X GPU, which is far less than standard NAS works. Moreover, we only need 3 minutes to train network parameters for evaluation. It is also faster than most existing network-based enhancement methods.

4.2. Comparison with State-of-the-arts

To prove the effectiveness of the RUAS, we compared RUAS with twelve recently-proposed state-of-the-art low-light image enhancement approaches including LIME [9],

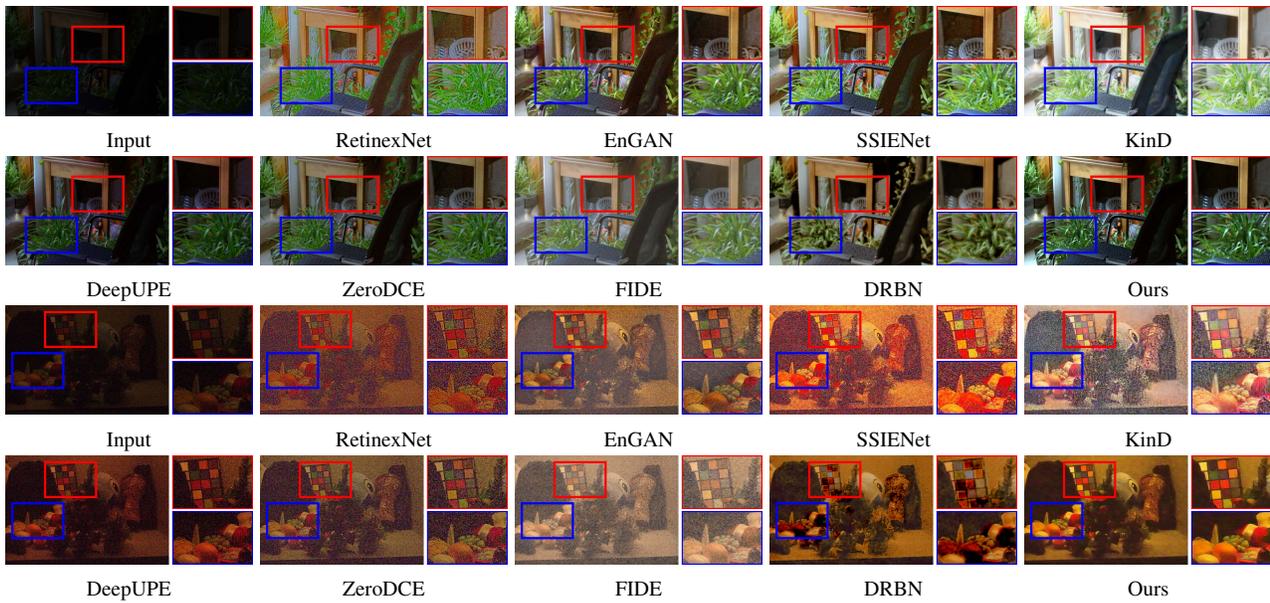


Figure 4. Visual results of state-of-the-art methods and ours on the LOL dataset. Red and blue boxes indicate the obvious differences.



Figure 5. Visual results of state-of-the-art methods and ours on the ExtremelyDarkFace dataset. Red boxes indicate the obvious differences.

SDD [10], MBLLEN [23], GLADNet [36], RetinexNet [2], EnGAN [12], SSIENet [48], KinD [49], DeepUPE [34], ZeroDCE [8], FIDE [37], and DRBN [40].

Firstly, we evaluated these methods in some simple real-world scenarios. We reported the quantitative scores on the MIT-Adobe 5K dataset. As shown in the first three rows in Table 1, we can easily see that our method received the best

numerical scores. Limited to space, we provided the visual comparisons on this benchmark in the Supplemental Materials. Then we evaluated the visual performance in some challenging real-world scenarios. Fig. 3 demonstrated three groups of visual comparisons on DarkFace dataset [41]. Although some methods were able to enhance the brightness successfully, they failed to restore the clear image textures.

Table 2. The model size, FLOPs and running time (GPU-seconds for inference) of some recently proposed CNN-based methods (with manually designed architectures) and our searched RUAS_i and RUAS_{i+n}. The FLOPs and running time are reported on the LOL dataset. The best result is in red whereas the second best one is in blue.

Methods	MBLEN	GLADNet	RetinexNet	EnGAN	SSIENet	KinD	FIDE	DRBN	RUAS _i	RUAS _{i+n}
SIZE (M)	0.450	1.128	0.838	8.636	0.682	8.540	8.621	0.577	0.001	0.003
FLOPs (G)	19.956	252.141	136.015	61.010	34.607	29.130	57.240	37.790	0.281	0.832
TIME (S)	0.077	0.025	0.119	0.010	0.027	0.181	0.594	0.053	0.006	0.016



Figure 6. Ablation study of the effect of different warm-start strategies. Subfigures (b)-(d) plot the enhanced results and corresponding estimated illumination maps for different settings.

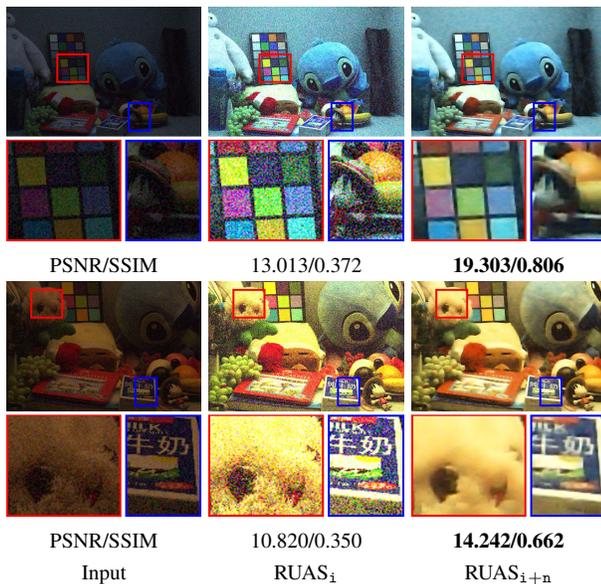


Figure 7. Ablation study of the contribution of NRM for low-light images in the noisy scenario. PSNR/SSIM are reported below each subfigure. Red and blue boxes indicate the obvious differences.

On the contrary, our RUAS can restore the brightness and the details perfectly at the same time.

Further, we evaluated our RUAS on the LOL dataset qualitatively and quantitatively, where the LOL dataset contained sensible noises to hinder the enhancement. The last three rows in Table 1 illustrated the quantitative comparisons. Obviously, our RUAS obtained a competitive PSNR, SSIM and LPIPS scores. As shown in Fig. 4, all compared methods failed to take on vivid and true colors. KinD and DRBN indeed removed noises but introduced some unknown artifacts. While our results presented vivid colors and removed undesired noises. Then some extreme exam-

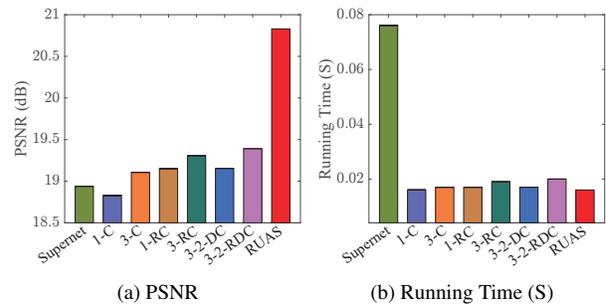


Figure 8. Comparison between naively determined architectures (with supernet and single type of convolution) and our searched RUAS on MIT-Adobe 5K dataset. Quantitative results (PSNR) and running time are plotted for these different architectures.

ples from the ExtremelyDarkFace dataset were showed in Fig. 5. These advanced deep networks being compared indeed improved the lightness, but lots of adverse artifacts became visible significantly. The recently-proposed methods, FIDE and DRBN even destroyed the color system, e.g., the overcoat should be red. By comparison, our RUAS improved brightness well and had great advantages in both detail restoration and noise removal.

Finally, we verified the memory and computation efficiency of our RUAS. Table 2 compared the model size, FLOPs, and running time among different state-of-the-art methods. As can be observed that the FLOPs and running time were calculated on 100 testing images with the size of 600×400 from the LOL dataset. Fortunately, our method just needed very small model size, FLOPs, and time. Note that our RUAS_{t+n} is our full version with the noise removal module. Even though its time consuming was a little higher than EnGAN, it is because that EnGAN ignored introducing the explicit noise removal module. In a word, Table 2 can fully verify the high-efficiency and fast speed of our RUAS.

4.3. Analysis and Discussions

Firstly, we evaluated the performance brought by three different warm-start strategies including fix warm-start as \hat{t}_0 , update \hat{t}_k w/o and w/ residual rectification. Fig. 6 provided the visual comparison of these warm-start strategies in terms of different components. We can observe that the updating strategy of w/o residual rectification indeed supplied the positive over-exposure suppression by comparing it with the naive fix warm-start strategy. Further, by introducing the mechanism of residual rectification, the result performed a more comfortable exposure (see the lampshade) than those by other strategies. In a word, we are able to confirm that our designed warm-start strategy really suppress the over-exposure during the propagations.

We then explored the effect of NRM. Fig. 7 provided the visual comparison on two examples that contained intensive noises hidden in the dark. After performing RUAS_i , image details were enhanced. However, there appeared the visible noises to harmfully damage the image quality. By introducing the NRM, our RUAS_{i+n} successfully removed the undesired noises to improve the visual quality and numerical scores. This experiment fully verified the necessity of introducing NRM in some complex real-world scenarios.

Subsequently, we analyzed the performance using different heuristically-designed architectures on the MIT-Adobe 5K dataset. As shown in Fig. 8, even though we adopted the complex supernet that contained massive parameters, the results performed the low PSNR and high time-consuming. As for other heuristically-designed architectures, the performance was also unideal. Briefly, these architectures may not effective enough. It is because that these architectures did not integrate the task cues/principles. By comparison, our searched architecture realized the highest PSNR, with an additional outcome (i.e., less inference time). In a word, this experiment indicated the necessity of searching for architecture and the superiority of our searched architecture.

Actually, the search strategy is a decisive factor for the searched architecture. To this end, we made an evaluation for different search strategies. We considered three search strategies based on how to search the IEM and NRM. The separate search strategy is to search these two parts one by one. That is, when searching the NRM, the searching procedure of the IEM has ended. The naive joint search is to view IEM and NRM as part of an entire architecture, and just need to search for all the architecture once. As shown in Fig. 9, our strategy is significantly effective according to the numerical scores compared with others. In addition, from the searched architecture, we can see that our searched NRM contained more residual convolution and skip connection, it is reasonable that this structure had been proved in some existing denoising works [45]. In other words, our cooperative search strategy indeed bridges the illumination estimation and denoising to realize a valuable collaboration.

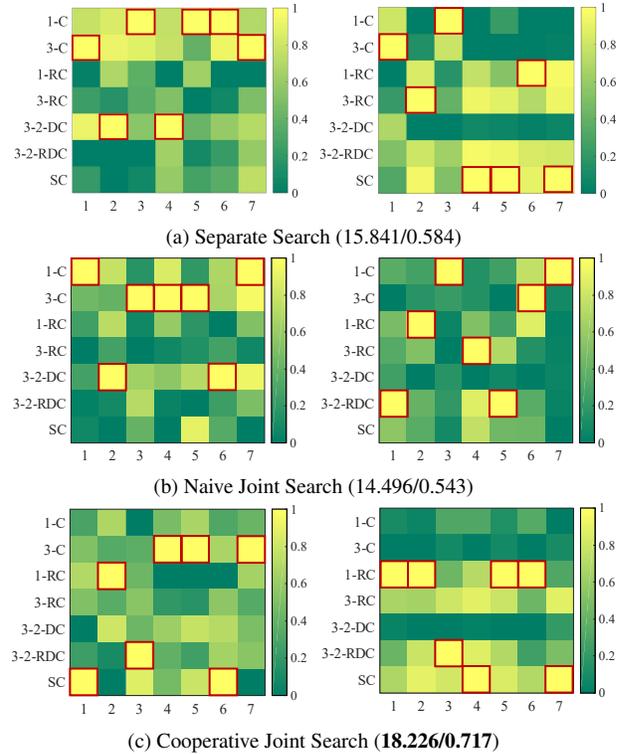


Figure 9. Heatmaps of these candidate architectures (i.e., α) in the last searching epoch. Red boxes indicate our searched architectures (with the highest score). Since we share cell for all the stages, thus only one cell is plotted. The left and right columns are the results of IEM and NRM, respectively. Quantitative results (PSNR/SSIM) are reported accordingly.

5. Conclusion

In this study, we proposed a new framework to integrate the principled optimization unrolling technique with a cooperative prior architecture search strategy for designing effective yet lightweight low-light enhancement network. We first established optimization models based on the Retinex rule to formulate the latent structures of the illumination map and our desired image. By unrolling the iteration process with abstract deep priors, we can obtain the holistic structure of our enhancement network. Then we developed a cooperative and reference-free strategy to discover specific architectures from a compact search space. Our experiments were performed on a series of challenging benchmark datasets and we derived new state-of-the-art results. In the future, we will apply our built architecture search based optimization unrolling technique for different vision tasks.

Acknowledgements: This work is partially supported by the National Natural Science Foundation of China (Nos. 61922019, 61733002, and 62027826), Liaoning Revitalization Talents Program (XLYC1807088), and the Fundamental Research Funds for the Central Universities.

References

- [1] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR*, pages 97–104, 2011. [5](#)
- [2] Wei Chen, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *BMVC*, 2018. [1](#), [2](#), [5](#), [6](#)
- [3] An-Chieh Cheng, Chieh Hubert Lin, Da-Cheng Juan, Wei Wei, and Min Sun. Instanas: Instance-aware neural architecture search. In *AAAI*, pages 3577–3584, 2020. [2](#)
- [4] HD Cheng and XJ Shi. A simple and effective histogram equalization approach to image enhancement. *DSP*, 14(2):158–170, 2004. [1](#)
- [5] George E Forsythe and Wolfgang R Wasow. Finite difference methods. *Partial Differential*, 1960. [5](#)
- [6] Xueyang Fu, Yinghao Liao, Delu Zeng, Yue Huang, Xiaoping Zhang, and Xinghao Ding. A probabilistic method for image enhancement with simultaneous illumination and reflectance estimation. *IEEE TIP*, 24(12):4965–4977, 2015. [1](#)
- [7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014. [4](#)
- [8] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong. Zero-reference deep curve estimation for low-light image enhancement. In *CVPR*, pages 1780–1789, 2020. [1](#), [2](#), [6](#)
- [9] Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE TIP*, 26(2):982–993, 2017. [1](#), [3](#), [5](#)
- [10] Shijie Hao, Xu Han, Yanrong Guo, Xin Xu, and Meng Wang. Low-light image enhancement with semi-decoupled decomposition. *IEEE TMM*, 2020. [6](#)
- [11] Shih-Chia Huang, Fan-Chieh Cheng, and Yi-Sheng Chiu. Efficient contrast enhancement using adaptive gamma correction with weighting distribution. *IEEE TIP*, 22(3):1032–1041, 2012. [1](#)
- [12] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlighten: Deep light enhancement without paired supervision. *IEEE TIP*, 30:2340–2349, 2021. [1](#), [2](#), [6](#)
- [13] Mading Li, Jiaying Liu, Wenhan Yang, Xiaoyan Sun, and Zongming Guo. Structure-revealing low-light image enhancement via robust retinex model. *IEEE TIP*, 27(6):2828–2841, 2018. [1](#)
- [14] Ruoteng Li, Robby T Tan, and Loong-Fah Cheong. All in one bad weather removal using architectural search. In *CVPR*, pages 3175–3185, 2020. [2](#)
- [15] Hanwen Liang, Shifeng Zhang, Jiacheng Sun, Xingqiu He, Weiran Huang, Kechen Zhuang, and Zhenguo Li. Darts+: Improved differentiable architecture search with early stopping. *arXiv preprint arXiv:1909.06035*, 2019. [4](#)
- [16] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *ICLR*, 2018. [2](#), [4](#), [5](#)
- [17] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. *arXiv preprint arXiv:2009.11551*, 2020. [4](#)
- [18] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *IJCV*, 128(2):261–318, 2020. [1](#)
- [19] Risheng Liu, Qianru Chen, Yuansheng Yao, Xin Fan, and Zhongxuan Luo. Location-aware and regularization-adaptive correlation filters for robust visual tracking. *IEEE TNNLS*, 2020. [1](#)
- [20] Risheng Liu, Shichao Cheng, Long Ma, Xin Fan, and Zhongxuan Luo. Deep proximal unrolling: Algorithmic framework, convergence analysis and applications. *IEEE TIP*, 28(10):5013–5026, 2019. [1](#)
- [21] Risheng Liu, Long Ma, Yuxi Zhang, Xin Fan, and Zhongxuan Luo. Underexposed image correction via hybrid priors navigated deep propagation. *IEEE TNNLS*, 2021. [1](#)
- [22] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *PR*, 61:650–662, 2017. [2](#)
- [23] Feifan Lv and Feng Lu. Attention-guided low-light image enhancement. *arXiv preprint arXiv:1908.00682*, 2019. [1](#), [6](#)
- [24] Marcin Mozejko, Tomasz Latkowski, Lukasz Treszczotko, Michal Szafraniuk, and Krzysztof Trojanowski. Superkernel neural architecture search for image denoising. In *CVPR Workshops*, pages 484–485, 2020. [2](#)
- [25] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005. [4](#)
- [26] Zia-ur Rahman, Daniel J Jobson, and Glenn A Woodell. Retinex processing for automatic image enhancement. *JEI*, 13(1):100–111, 2004. [1](#)
- [27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015. [2](#)
- [28] Debdoot Sheet, Hrushikesh Garud, Amit Suveer, Manjunatha Mahadevappa, and Jyotirmoy Chatterjee. Brightness preserving dynamic fuzzy histogram equalization. *IEEE TCE*, 56(4), 2010. [1](#)
- [29] Himanshu Singh, Anil Kumar, LK Balyan, and Girish Kumar Singh. A novel optimally gamma corrected intensity span maximization approach for dark image enhancement. In *DSP*, pages 1–5, 2017. [1](#)
- [30] David So, Quoc Le, and Chen Liang. The evolved transformer. In *ICML*, pages 5877–5886, 2019. [2](#)
- [31] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, pages 10781–10790, 2020. [2](#)
- [32] Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuan-dong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, et al. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In *CVPR*, pages 12965–12974, 2020. [2](#)
- [33] Ning Wang, Yang Gao, Hao Chen, Peng Wang, Zhi Tian, Chunhua Shen, and Yanning Zhang. Nas-fcos: Fast neu-

- ral architecture search for object detection. In *CVPR*, pages 11943–11951, 2020. 2
- [34] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *CVPR*, pages 6849–6857, 2019. 2, 6
- [35] Wei Wang, Na Sun, and Michael K Ng. A variational gamma correction model for image contrast enhancement. *Inverse Problems & Imaging*, 13(3):461, 2019. 1
- [36] Wenjing Wang, Chen Wei, Wenhan Yang, and Jiaying Liu. Gladnet: Low-light enhancement network with global awareness. In *FG*, pages 751–755, 2018. 1, 6
- [37] Ke Xu, Xin Yang, Baocai Yin, and Rynson WH Lau. Learning to restore low-light images via decomposition-and-enhancement. In *CVPR*, pages 2281–2290, 2020. 1, 2, 6
- [38] Li Xu, Qiong Yan, Yang Xia, and Jiaya Jia. Structure extraction from texture via relative total variation. *ACM TOG*, 2012. 4
- [39] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. Pc-darts: Partial channel connections for memory-efficient differentiable architecture search. In *ICLR*, 2020. 2, 4
- [40] Wenhan Yang, Shiqi Wang, Yuming Fang, Yue Wang, and Jiaying Liu. From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement. In *CVPR*, pages 3063–3072, 2020. 1, 2, 6
- [41] Wenhan Yang, Ye Yuan, Wenqi Ren, Jiaying Liu, Walter J Scheirer, Zhangyang Wang, Taiheng Zhang, Qiaoyong Zhong, Di Xie, Shiliang Pu, et al. Advancing image understanding in poor visibility environments: A collective benchmark study. *IEEE TIP*, 29:5737–5752, 2020. 5, 6
- [42] Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *CVPR*, 2020. 1
- [43] Se Hwan Yun, Heon Kim Jin, and Suki Kim. Contrast enhancement using a weighted histogram equalization. In *ICCE*, volume 10, pages 203–204, 2011. 1
- [44] Haokui Zhang, Ying Li, Hao Chen, and Chunhua Shen. Memory-efficient hierarchical neural architecture search for image denoising. In *CVPR*, pages 3657–3666, 2020. 2
- [45] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE TIP*, 26(7):3142–3155, 2017. 8
- [46] Qing Zhang, Yongwei Nie, Lei Zhu, Chunxia Xiao, and Wei-Shi Zheng. Enhancing underexposed photos using perceptually bidirectional similarity. *IEEE TMM*, 2020. 1
- [47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5
- [48] Yu Zhang, Xiaoguang Di, Bin Zhang, and Chunhui Wang. Self-supervised image enhancement network: Training with low light images only. *arXiv preprint arXiv:2002.11300*, 2020. 1, 2, 6
- [49] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. In *ACM MM*, 2019. 1, 2, 6
- [50] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017. 2