

FastLLVE: Real-Time Low-Light Video Enhancement with Intensity-Aware Lookup Table

Wenhao Li*

Shanghai Jiao Tong University
Shanghai, China
wenhaoli.233.411@gmail.com

Guangyang Wu*

Shanghai Jiao Tong University
Shanghai, China
wu.guang.young@gmail.com

Wenyi Wang

University of Electronic Science and
Technology of China
Chengdu, China
wangwenyi@uestc.edu.cn

Peiran Ren

Alibaba Damo Academy
Hangzhou, China
peiran_r@sohu.com

Xiaohong Liu†

Shanghai Jiao Tong University
Shanghai, China
xiaohongliu@sjtu.edu.cn

ABSTRACT

Low-Light Video Enhancement (LLVE) has received considerable attention in recent years. One of the critical requirements of LLVE is inter-frame brightness consistency, which is essential for maintaining the temporal coherence of the enhanced video. However, most existing single-image-based methods fail to address this issue, resulting in flickering effect that degrades the overall quality after enhancement. Moreover, 3D Convolution Neural Network (CNN)-based methods, which are designed for video to maintain inter-frame consistency, are computationally expensive, making them impractical for real-time applications. To address these issues, we propose an efficient pipeline named *FastLLVE* that leverages the Look-Up-Table (LUT) technique to maintain inter-frame brightness consistency effectively. Specifically, we design a learnable Intensity-Aware LUT (IA-LUT) module for adaptive enhancement, which addresses the low-dynamic problem in low-light scenarios. This enables FastLLVE to perform low-latency and low-complexity enhancement operations while maintaining high-quality results. Experimental results on benchmark datasets demonstrate that our method achieves the State-Of-The-Art (SOTA) performance in terms of both image quality and inter-frame brightness consistency. More importantly, our FastLLVE can process 1,080p videos at 50+ Frames Per Second (FPS), which is 2 \times faster than SOTA CNN-based methods in inference time, making it a promising solution for real-time applications. The code is available at <https://github.com/Wenhao-Li-777/FastLLVE>.

CCS CONCEPTS

• Computing methodologies → Reconstruction.

*Both authors contributed equally to this research.

†Corresponding Author: xiaohongliu@sjtu.edu.cn

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0108-5/23/10...\$15.00

<https://doi.org/10.1145/3581783.3611933>

KEYWORDS

Low-light video enhancement, lookup table, brightness consistency

ACM Reference Format:

Wenhao Li, Guangyang Wu, Wenyi Wang, Peiran Ren, and Xiaohong Liu. 2023. FastLLVE: Real-Time Low-Light Video Enhancement with Intensity-Aware Lookup Table. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23)*, October 29–November 3, 2023, Ottawa, ON, Canada. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3581783.3611933>

1 INTRODUCTION

Low-Light Video Enhancement (LLVE) is a longstanding task aiming at transforming low-light videos into normal-light videos with better visibility, which has received considerable attention in recent years. In low-light conditions, videos often suffer from deteriorated texture and low contrast, leading to poor visibility and significant degradation of high-level vision tasks. Unlike traditional methods based on higher ISO and exposure that can cause noise and motion blur [4], LLVE offers an effective solution to improve the visual quality of videos captured in extremely low-light conditions. Moreover, it can serve as a fundamental enhancement module for a wide range of applications, e.g., visual surveillance [46], autonomous driving [18], and unmanned aerial vehicle [31].

Like other typical video tasks, such as Video Frame Interpolation [33, 34, 42] and Video Super-Resolution [8, 19–21, 32, 48], LLVE also demands temporal stability. Additionally, the inherently ill-posed nature of LLVE makes it a more challenging task. As a result, although Low-Light Image Enhancement (LLIE) have demonstrated remarkable performance, recursively applying these image-based methods to video frames isn't feasible. Because it is time-consuming and may result in flickering effect in the enhanced video. As revealed in [10], the flickering problem is caused by the inconsistency in brightness between adjacent frames. To address this issue, recent LLVE methods have leveraged temporal alignment [38] and 3D Convolution (3D-Conv) [10, 23] to establish the spatial-temporal relationship in video. They have also adopted the self-consistency [2, 51] as an auxiliary loss to guide the network in maintaining brightness consistency. However, alignment-based methods, which aim to estimate the corresponding pixels between adjacent frames, are prone to errors and can lead to object distortion

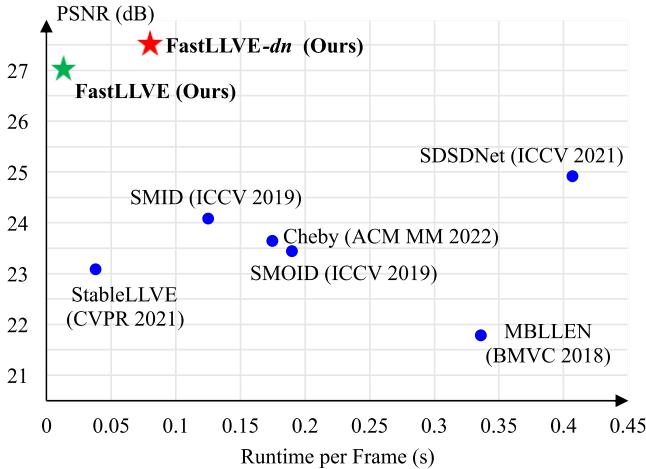


Figure 1: Comparisons of effectiveness and efficiency. Our method outperforms the current SOTA method (*i.e.*, SDSDNet) by a large margin in terms of PSNR, and faster than the most efficient method (*i.e.*, StableLLVE). The average PSNR is evaluated on SDSD test dataset [38], and runtime is evaluated on 1,080p videos with a Nvidia RTX 3090 GPU.

in the enhanced video. In contrast, 3D-Conv is capable of capturing comprehensive spatial-temporal information, but at the cost of greater computational complexity. Therefore, previous methods have found it challenging to strike a balance between efficiency and performance. To sum up, a considerate LLVE method should address the following challenging issues:

- **Ill-posed problem.** In low-light videos, the low dynamic range of the color space can result in similar color inputs appearing for different target colors. This phenomenon leads to the one-to-many mapping problem which is challenging to solve in complex scenarios. To address this problem, previous methods [10, 23, 38] have leveraged global context information and local consistency to enhance different colors. Despite their respective efficacy, these methods are plagued by instability with respect to color handling due to their heavy reliance on the precision and reliability of context extraction.
- **Brightness consistency.** Maintaining brightness consistency in the output video is crucial for achieving high perceptual quality in LLVE. However, current alignment-based method [38] often fails to achieve accurate alignment between adjacent frames, leading to unstable output for LLVE. Otherwise, self-consistency loss functions [2, 51] used to improve the stability of these methods are also unable to address the fundamental instability problem. This limitation hinders their ability to effectively improve their overall visual quality.
- **Efficiency.** Although 3D-Conv methods have shown significant improvement in video enhancement tasks by exploiting comprehensive spatial-temporal information [10, 23], they are associated with heavy computational complexity. This makes them impractical for real-world applications that require real-time enhancement.

To address the above issues, we propose a novel framework named FastLLVE. Our approach establishes a stable and adaptive Look-Up-Table (LUT) to enable real-time LLVE. In particular, we

design an Intensity-Aware LUT (IA-LUT) to transform RGB colors from one color space to another, which can handle the one-to-many mapping issue that commonly arises in LLVE. Unlike traditional LUTs where one-to-one mapping relationships for color values are stored, our IA-LUT stores the one-to-many mapping relationships, with respect to learnable enhancement intensities for every pixels. To improve the generalization ability of our method, we follow the parameterization approach [50] and combine a set of basis LUTs with dynamic weights. Importantly, our approach maintains the inter-frame brightness consistency by nature, as the pixel-wise LUT-based transformation is consistent with all pixels having the same RGB values and enhancement intensity. In addition, our method is computationally efficient and suitable for real-time video enhancement. To address the issue of noise that the LUT might fail to deal with, particularly in extremely low-light conditions, we simplify a common denoising method [3] to incorporate a plug-in refinement module for denoising denoted as FastLLVE-*dn*, which further improves the performance at the expense of some efficiency. It is worth noting that other denoising methods can readily replace the used one. As demonstrated in Figure 1, both of our two models outperform existing methods by a significant margin in terms of Peak Signal-to-Noise Ratio (PSNR), while the FastLLVE achieves the real-time processing speed of over 50 Frames Per Second (FPS).

The contributions of this paper can be summarized as follows:

- We propose a novel LUT-based framework, named FastLLVE, for real-time low-light video enhancement.
- We design a novel and lightweight Intensity-Aware LUT, which accounts for the one-to-many mapping problem in LLVE.
- Extensive experiments show that the FastLLVE achieves the SOTA results on benchmarks in most cases, with over 50 FPS inference speed.

2 RELATED WORKS

2.1 Low-light Image Enhancement

Researches on low-light enhancement started with traditional LLIE methods including Histogram Equalization [9, 27, 37] and Retinex theory [5, 6, 16, 40]. Then deep-learning approaches [15, 23, 26, 28, 39, 47, 53] have shown the great superiority on effectiveness, efficiency and generalization ability. Lv *et al.* [23] present a multi-branch network, which extracts rich features from different levels, to enhance low-light images via multiple subnets. Wang *et al.* [39] introduce intermediate illumination rather than directly learn an image-to-image mapping. Pan *et al.* [28] propose a new model learning to estimate pixel-wise adjustment curves and recurrently reconstruct the output. Zhou *et al.* [53] specially design a network for joint low-light enhancement and deblurring.

2.2 Low-light Video Enhancement

LLVE, an extension of LLIE, imposes an additional requirement of brightness consistency, as outlined in [15]. Existing LLVE methods address this challenge through three common solutions, namely 3D Convolution, Feature Alignment, and Self-consistency. Lv *et al.* [23] exchange all 2D-Conv layers of their proposed LLIE network into 3D-Conv layers to achieve the processing of low-light videos. Jiang *et al.* [10] train a LLVE network based on 3D U-Net. Instead of 3D-Conv, Wang *et al.* [38] align adjacent frames into the

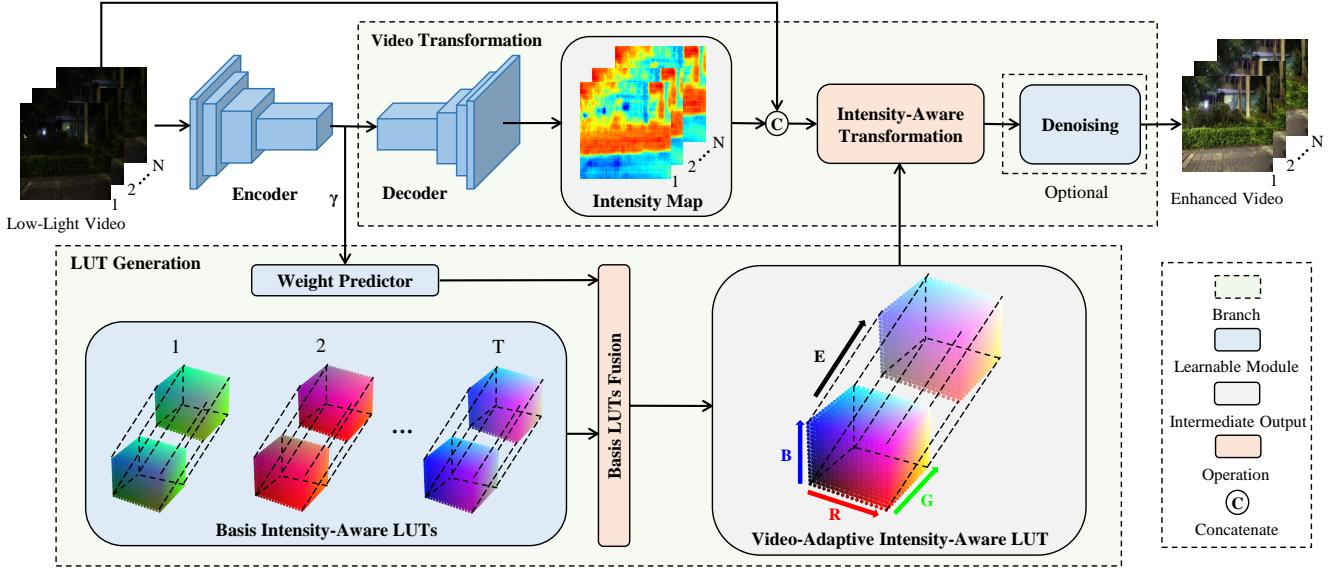


Figure 2: The architecture of the proposed network based on the designed Intensity-Aware LUT (IA-LUT). The lightweight encoder-decoder network extracts spatial and temporal features for building a video-adaptive IA-LUT and generates intensity map related to the input video. Then weight predictor utilizes the feature vector γ from the encoder to predict weights that guide the fusion of T basis IA-LUTs. Through IA-LUT transformation, the input video concatenated with intensity map transforms to the enhanced normal-light video. Finally, an optional denoising module can help the IA-LUT deal with noise.

middle frame for lighting enhancement and noise reduction based on Retinex theory [16]. In order to improve efficiency, some methods use 2D-Conv with self-consistency as an auxiliary loss. Chen *et al.* [2] randomly select two frames from the same low-light video to train a deep twin network, using self-consistency loss to make the network robust to noise and small changes in the scene. Rather than select similar frames, Zhang *et al.* [51] choose to simulate adjacent frames and ground truths by warping the input image and its corresponding ground truth based on the predicted optical flow, so as to artificially synthesize similar data pairs for self-consistency loss. However, self-consistency is a weak and unstable constraint which cannot solve the fundamental problem of brightness consistency.

2.3 LUT for Image Enhancement

A 3D-LUT is a 3-dimensional grid of values, which maps the input color values to the corresponding output color values. By applying such a transformation to an image or video, it is possible to achieve a wide range of color and tonal effects, from subtle color grading to dramatic color transformations. LUT has already been a classic and commonly used pixel adjustment tool in ISP system [11] and image editing software because of its high efficiency for modeling color transforms. Recently, deep-learning methods based on LUT are proposed in image enhancement tasks. Zeng *et al.* [50] first leverage a lightweight CNN to predict the weights for integrating multiple basis LUTs, and the constructed image-adaptive LUT is utilized to achieve image enhancement. Wang *et al.* [41] further propose a learnable spatial-aware LUT which considers the global scene and local spatial information. Yang *et al.* [44] realize the importance of the sampling strategy so that they design a non-uniform sampling strategy based on learnable adaptive sampling

intervals to replace the sub-optimal uniform sampling strategy. At the same time, Yang *et al.* [45] also try to combine 1D LUTs and 3D LUT to promote each other and achieve a more lightweight 3D LUT with better performance. To the best of our knowledge, LUT has not been adopted in LLVE tasks. In this paper, we will introduce how LUT is naturally suitable for LLVE and enables real-time applications.

3 METHOD

This section provides an overview of the structural intricacies of FastLLVE, as shown in Figure 2. Input video frames are first encoded into latent features through a lightweight encoder network. Afterwards, the latent features are parallel fed into two modules, namely LUT Generation Module and Video Transformation Module. Specifically, a video-adaptive LUT is generated through the LUT Generation Module, while an intensity map is generated for video transformation. Then, each pixel is enhanced via the IA-LUT transformation with its RGB values and enhancement intensity as the index. Finally, the transformed video is feed into a denoising module for further enhancement. Sections 3.1 and 3.2 will focus on the LUT Generation Module and Video Transformation Module, respectively. More structure details about the feature encoder network and denoising module can be found in appendix.

3.1 LUT Generation Module

Definition. Although low-light pixels from various areas may appear similar in RGB, they correspond to distinct enhancement intensities during the low-light enhancement process. In Figure 3, we visualize several intensity maps where even pixels from extremely low-light videos have different enhancement intensities. Traditional

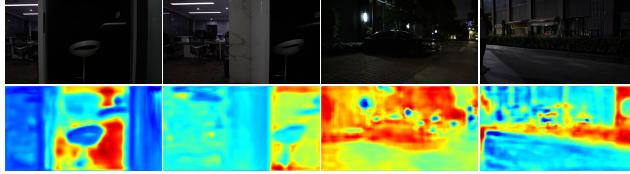


Figure 3: Visualization of the intensity map from extremely low-light videos. The first line consists of input frames and the second line consists of the corresponding intensity maps.

3-dimensional LUTs only save one-to-one mapping relationships for color transformation, which fails on solving the ill-posed problem of low-light pixels with similar color. In order to address this issue, we add a new dimension denoting the enhancement intensity, and the corresponding LUT is denoted as Intensity-Aware LUT. It can store several color spaces for one-to-many mapping relationships and facilitates finer color transformation for LLVE. It is worth noting that only a sampled sparse discrete input space is saved in IA-LUT to avoid introducing massive parameters, which can result in heavy memory burden and great training difficulty. And due to the sparse discrete 4D input space, the LUT transformation of the IA-LUT should be implemented using quadrilinear interpolation.

Let $\mathcal{V} : [0, 1]^4 \rightarrow [0, 1]^3$ be a function defined by the IA-LUT, we have

$$\mathcal{V}(r, g, b, e) = [r', g', b'] \quad (1)$$

where r, g, b, e indicate the input red, green, blue colors and enhancement intensity, and r', g', b' are the mapped color values. Let L be the number of grid points in each dimension of the IA-LUT, and $C_x = [r_i, g_j, b_k, e_m]$ stands for the index of grid point $x = [i, j, k, m]$, where $0 \leq i, j, k, m \leq L - 1$. For this grid point x , the stored values in IA-LUT for color mapping are represented as $C'_x = [r_x, g_x, b_x]$. If the input indices $[r, g, b, e]$ can not be mapped to any grid point, we will apply quadrilinear interpolation in the nearest unit lattice. For brevity, we here let

$$\Omega_x = (r_i, r_{i+1}) \times (g_j, g_{j+1}) \times (b_k, b_{k+1}) \times (e_m, e_{m+1}) \quad (2)$$

as the unit lattice at grid point $x \in [0, 1, \dots, L - 1]^4$, where we have

$$r_{i+1} > r_i, g_{j+1} > g_j, b_{k+1} > b_k \text{ and } e_{m+1} > e_m. \quad (3)$$

Then, the quadrilinear interpolation process I_{Ω_x} in the unit lattice Ω_x is formulated as:

$$I_{\Omega_x}(r, g, b, e) = \left[\sum_{n=1}^{2^4} O_x^n \cdot r_x^n, \sum_{n=1}^{2^4} O_x^n \cdot g_x^n, \sum_{n=1}^{2^4} O_x^n \cdot b_x^n \right], \quad (4)$$

where coefficients $O_x^n, n \in [1, 2, \dots, 16]$ indicates the offsets of the input index to the nearest 2^4 sampling grids of lattice Ω_x . In conclusion, the IA-LUT \mathcal{V} can be formulated as:

$$\mathcal{V}(r, g, b, e) = \begin{cases} C'_x, & \text{if } [r, g, b, e] = [r_i, g_j, b_k, e_m], \\ I_{\Omega_x}(r, g, b, e), & \text{if } [r, g, b, e] \in \Omega_x, \\ [0, 0, 0], & \text{otherwise.} \end{cases} \quad (5)$$

In Figure 4, we illustrate the quadrilinear interpolation process, and the detailed formulation of coefficients can be found in appendix.

Generation. In order to automatically generate video-adaptive IA-LUT, as shown in Figure 2, we learn T learnable basis IA-LUTs and

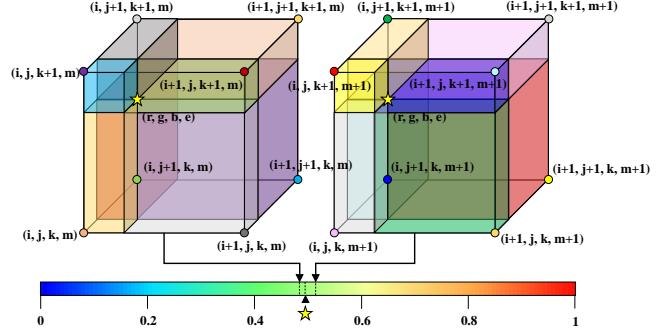


Figure 4: Illustration of the quadrilinear interpolation for one input pixel.

fuse them based on T video-dependent weights, where T is the number of basis LUTs. Compared with directly generating all elements of the video-adaptive IA-LUT via CNN, fusing several basis LUTs is more efficient and computationally inexpensive. More specifically, suppose the low-light video $Y \in \mathbb{R}^{N \times H \times W \times 3}$ with N frames of resolution $W \times H$ is taken as the input, at the beginning, a lightweight encoder with five 3D convolution layers, each with a $3 \times 3 \times 3$ kernel size, is used to capture the coarse understanding and some global attributes of the input video Y . The output of the encoder is resized to a compact feature vector $\gamma \in \mathbb{R}^{P \times Q}$, which serves as a guide to construct video content-dependent LUT parameters. The size of the feature vector γ is due to the two hyper-parameters P and Q , which denote the number of pixels and the number of channels before the resizing, respectively. In this paper, we set P to 16 and Q to 64 according to the structure of the encoder which can be found in the appendix. After the shared encoder, the weight predictor based on the fully-connected layer maps the compact feature vector γ into T dynamic video-dependent weights, which can be formulated as:

$$h_0 : \mathbb{R}^{P \times Q} \rightarrow \mathbb{R}^T, \quad (6)$$

where h_0 denotes the mapping from the feature vector γ to the video-dependent weights for fusion. Subsequently, another fully-connected layer is employed to map the video-dependent weights to all elements of the video-adaptive IA-LUT. The learnable parameters of this layer are encoded basis IA-LUTs. We refer to this mapping as h_1 and describe it as:

$$h_1 : \mathbb{R}^T \rightarrow \mathbb{R}^{L \times L \times L \times L \times 3}, \quad (7)$$

where $L \times L \times L \times L \times 3 = 3 \times L^4$ is the total number of elements of the generated video-adaptive IA-LUT, and the number 3 means that the IA-LUT stores the mapped red, green and blue color values, respectively. The elements of the basis IA-LUTs can be updated during the end-to-end training since they serve as the parameters of the fully-connected layer, which makes the basis LUTs learnable.

In general, besides the shared encoder, two fully-connected layers achieve the main mapping h from the feature vector γ to the generated video-adaptive IA-LUT, as shown below:

$$h : \gamma \xrightarrow{h_0} w \in \mathbb{R}^T \xrightarrow{h_1} C' \in \mathbb{R}^{L \times L \times L \times L \times 3}, \quad (8)$$

where $C' \in \mathbb{R}^{L \times L \times L \times L \times 3}$ denotes all the stored elements C'_x of the target video-adaptive IA-LUT, and $w \in \mathbb{R}^T$ represents the video-dependent weights obtained through the mapping h_0 . As shown

Table 1: Quantitative comparisons on SDSD and SMID test datasets. Top two numbers of each column are with the best in red and the second best in blue. "FastLLVE+dn" denotes our method with a simple denoising module.

Format	Method	SDSD				SMID				Runtime (s)
		PSNR	SSIM	AB (Var)↓	MABD↓	PSNR	SSIM	AB (Var)↓	MABD↓	
Image	MBLLEN [23]	21.79	0.65	\	\	22.67	0.68	\	\	0.336
	Cheby [28]	23.65	0.81	0.079	0.297	25.24	0.76	1.486	1.891	0.175
Video	SALVE [1]	18.03	0.69	0.125	0.246	16.73	0.60	1.984	3.501	0.182
	SMOID [10]	23.45	0.69	0.397	0.749	23.64	0.71	1.455	1.736	0.190
	SMID [2]	24.09	0.69	0.784	1.592	24.78	0.72	0.405	0.794	0.125
	SDSDNet [38]	24.92	0.73	0.181	0.193	26.03	0.75	0.737	0.944	0.407
	StableLLVE [51]	23.09	0.81	1.366	2.814	26.22	0.78	0.745	0.897	0.038
Video	FastLLVE	27.06	0.78	0.038	0.091	26.45	0.75	0.476	0.748	0.013
	FastLLVE+dn	27.55	0.86	0.033	0.040	27.62	0.80	0.065	0.050	0.080

above, the mapping h is actually a cascade of the mapping h_0 and h_1 . It's worth emphasizing that dividing the main mapping h into two parts, each realized through a fully-connected layer, is crucial to reduce the number of parameters, similar to the sampled input space of LUT. Using only one fully-connected layer to directly map the compact feature vector y to the generated LUT would lead to a significantly larger number of parameters, specifically $P \times Q \times 3 \times L^4$, compared to $T \times (P \times Q + 3 \times L^4)$. Therefore, dividing the mapping h by rank factorization and implementing it with two fully-connected layers can reduce the parameters, making the transformation easier to learn and optimize.

3.2 Video Transformation

As shown in Figure 2, we first estimate the intensity map, then perform look-up according to the RGB video and corresponding intensity map. In order to construct this map, a lightweight decoder with five deconvolution layers [49] of size $3 \times 3 \times 3$ is adopted to utilize the latent features from the encoder, resulting in an intensity map $I \in \mathbb{R}^{N \times H \times W \times 1}$. By concatenating the intensity map and the input video, we can perform look-up and interpolation as introduced in Section 3.1. We recursively apply the transformation on each frame of a video sequence, resulting in the video output with stable and consistent brightness.

In addition, as the LUT transformation is applied to each pixel independently and quadrilinear interpolation can be parallel processed, we implement the IA-LUT transformation via CUDA to accelerate the transformation and achieve the convenient end-to-end training. Specifically, we merge the lookup and interpolation operations into a single CUDA kernel to maximize the parallelism. Following Adaint [44], we also adopt binary search algorithm during lookup operation, because the logarithmic time complexity can make computational cost negligible, unless L is set to an unexpected large value. It is important to emphasize that the pixel-wise transformation, which is indexed only by the red, green, blue colors and enhancement intensity of each input pixel, is the key to the IA-LUT naturally maintaining the inter-frame brightness consistency.

Although the FastLLVE framework can naturally maintain inter-frame brightness consistency, and achieves the great performance at the same time, it should be pointed out that LUT is susceptible to noise. In the real world, images and videos captured in low-light

conditions inevitably contain noises. Therefore, we sacrifice some efficiency to add a simple denoising module as the post-processing, refining the enhanced normal-light video for better performance. Specifically, in this paper, we choose to follow the practice in [3] to design the additional denoising module. However, it is worth noting that almost all existing denoising methods, such as [24, 30, 35, 52], can be alternatives as the post-processing.

4 EXPERIMENTS

4.1 Implementation Details

We implement our method based on PyTorch [29] and train the framework on a NVIDIA GeForce RTX 3090 GPU. The standard Adam optimizer [12] is adopted to train the entire network, with the batch size set to 8. The initial learning rate is set to 4×10^{-4} and gradually decayed according to the scheme of Cosine annealing [7] with restart set to 10^{-7} .

Regarding the loss function, since previous LUT-based methods [44, 45, 50] have proven the effectiveness of the smooth regularization and monotonicity regularization, we add 4D smooth regularization and 4D monotonicity regularization adapted to the IA-LUT into the loss function. If we add the additional denoising module, a pairwise loss between the denoised result and ground truth will be included in the loss function. As a result, the total loss function is defined as:

$$l_{total} = \begin{cases} l_{r0} + l_{r1} + \alpha_s l_s + \alpha_m l_m, & \text{if denoising,} \\ l_{r0} + \alpha_s l_s + \alpha_m l_m, & \text{otherwise,} \end{cases} \quad (9)$$

where l_{r0} denotes the reconstruction loss between the transformed normal-light video and ground truth, and l_{r1} denotes the denoising loss between the final denoised result and ground truth. Both of them use Charbonnier Loss [13, 14]. The 4D smooth regularization loss l_s consists of two parts which correspond to the video-adaptive IA-LUT and the video-dependent weights, respectively. It prevents artifacts caused by extreme color changes in LUT, while the 4D monotonicity regularization loss l_m preserves the robustness during the enhancement process. The detailed formulations of l_s and l_m can be found in appendix.

As for hyper-parameters, in the loss function, we set α_s and α_m to 0.0001 and 10, respectively. In terms of L and T , although

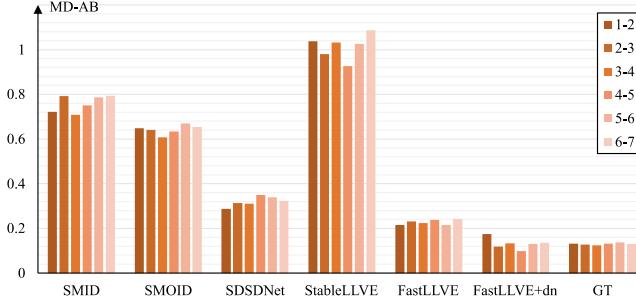


Figure 5: The Mean Differences of Average Brightness (MD-AB) between adjacent frames of SDSD test dataset. [Key: 1-2: difference between the 1st frame and the 2nd frame, 2-3 to 6-7 are indicated similarly]

higher values contribute to the precision of color transformation modeled by LUT, they can significantly increase the parameters of all LUTs used in the entire framework. Therefore, we follow the most widely-used setting [44, 45, 50] of the two numbers, which is proposed to balance the accuracy and the size of parameters. Thus, the number of sampling grid points on each dimension is set to 33 and the number of basis IA-LUTs is set to 3.

4.2 Experiment Setup

We present a comprehensive comparison of our proposed method with eight SOTA low-light enhancement methods, including both image-based and video-based approaches. The image-based methods we evaluate are MBLLEN [23], Cheby [28] and LEDNet [53], while SMID [2], SMOID [10], StableLLVE [51], SALVE [1] and SDSDNet [38] are completely video-based methods. We use their released codes and follow the same training strategies to train these networks on two real-world low-light video datasets, namely SDSD [38] dataset and SMID [2] dataset. The SDSD dataset is split into the SDSD training and test datasets with 23,542 and 750 video frames respectively, while the SMID training and test datasets contain 18,278 and 1,470 video frames. Then, we evaluate the performance of FastLLVE and the compared methods on the SDSD and SMID test datasets, to demonstrate the superiority of our method. It's worth noting that the SMID dataset used in our experiments has been pre-processed to transform the RAW data into sRGB data via its own pre-processing method proposed in SMID.

We use the common evaluation metrics of Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) to assess the quality of the enhanced videos. In addition, drawing on previous works [10, 23, 51], we consider the variance of Average Brightness (AB (Var)) and Mean Absolute Brightness Difference (MABD) to evaluate the ability to maintain inter-frame brightness consistency, where lower values stand for better consistency. Furthermore, we also record the average processing time of a 1,080p low-light frame on a NVIDIA GeForce RTX 3090 GPU for each method.

4.3 Comparisons of Brightness Consistency

As presented in Table 1, the AB (Var) and MABD are employed to assess the maintenance of inter-frame brightness consistency. FastLLVE+dn outperforms the compared methods on the two test

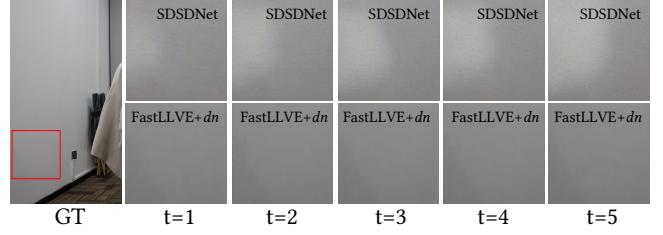


Figure 6: The local inter-frame brightness inconsistency on SDSD test dataset.

datasets, including the SDSDNet that is the current SOTA in terms of brightness consistency. It is worth noting that SDSDNet also contains a denoising module based on the Retinex theory [16]. Moreover, we also compute the Mean Differences of Average Brightness (MD-AB) between adjacent frames of SDSD test dataset shown in Figure 5. It is evident that the FastLLVE+dn achieves the best performance in brightness consistency and behaves most similar to ground truth.

In addition to quantitative comparisons, we also provide the qualitative comparisons in Figure 6. SDSDNet generates an incorrect light spot that varies among adjacent frames, which manifests its limitation when dealing with local brightness inconsistencies. In comparison, FastLLVE+dn has better ability to suppress the inter-frame brightness inconsistency in local areas, benefiting from the elaborate design of IA-LUT.

4.4 Comparisons of Enhanced Performance

In Table 1, we present the performance comparisons in terms of PSNR and SSIM on both SDSD and SMID test datasets. Compared with the StableLLVE, our FastLLVE achieves almost 3× inference speed and outperforms it in terms of PSNR. Meanwhile, the FastLLVE+dn achieves the SOTA performance in PSNR and SSIM on both datasets, and outperforms the StableLLVE by a large margin. Notably, the generally higher values of PSNR on SDSD test dataset indicate the difficulty of color restoration from the color space with an extremely low dynamic range since most videos in SDSD test dataset are taken in extremely low-light scenarios.

We further conduct qualitative comparisons on the two datasets in Figure 7-8. In Figure 7, we present the results of extremely low-light video in SDSD test dataset. Except for SMOID and SDSDNet, previous methods suffer from incorrect color restoration evidently. SMOID produces blurry results and lack of texture details. SDSDNet produces certain degree of noise owing to the deviation of noise map estimation. Besides, the darker area and the colorful border of the bright area appear in the results of other methods, which indicates the incorrect enhancement in brightness. Compared with these methods, our FastLLVE+dn solves the difficulties of color restoration, enhancement in brightness, as well as noise suppression, achieving visually great performance. In Figure 8, it is challenging to restore low-light videos with severe color biases. For instance, the enhanced low-light videos from SDSDNet present much darker than the ground truth, and the results of StableLLVE suffer from color distortions. The reason behind might be the inaccurate estimation of color transformation. In comparison, both FastLLVE and FastLLVE+dn can enhance the low-light videos well

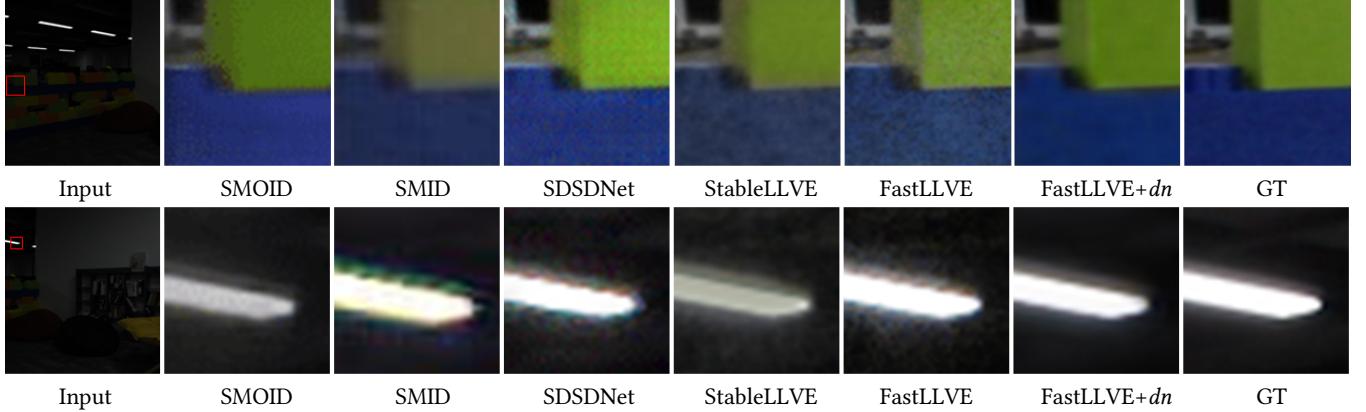


Figure 7: Qualitative comparisons on SDSD test dataset. Our method achieves correct enhancement and the best performance.

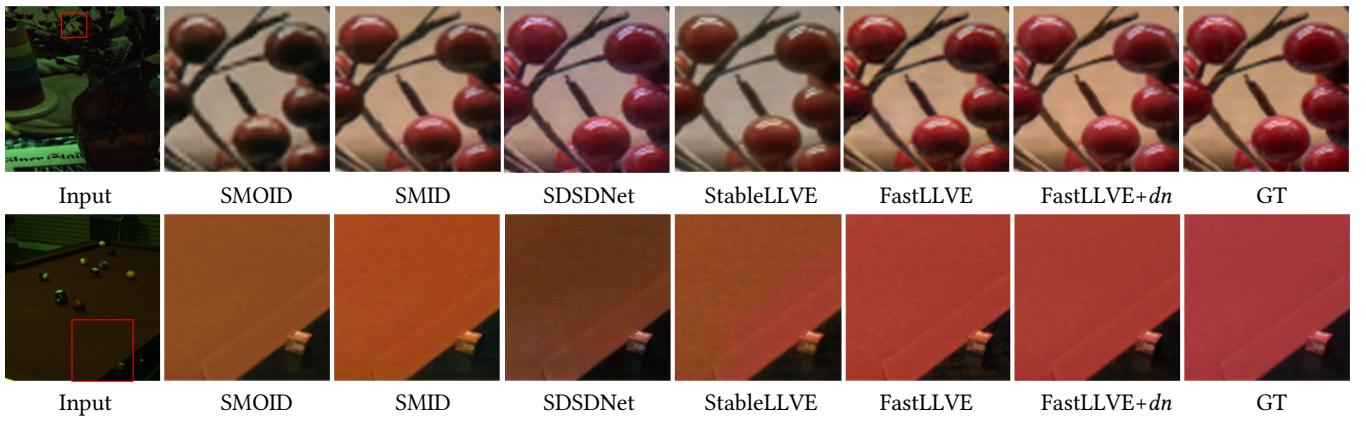


Figure 8: Qualitative comparisons on SMID test dataset. Compared to other methods, only our method restores color correctly.

with a similar color space to ground truth, even though the color biases exist.

4.5 Ablation Study

In order to evaluate the effectiveness of the IA-LUT and explore better use of the denoising module, we perform the ablation study and show the results in Figure 9 and Table 5.

Structure of LUT: We change IA-LUT into the common 3D LUT and remove the decoder, so as to construct the low-light video enhancement network based on 3D LUT in comparison to the FastLLVE with IA-LUT. As shown in Table 5, FastLLVE completely outperforms the 3D LUT-based network, which demonstrates that the additional dimension of enhancement intensity greatly improves the ability of common 3D LUT to model color transformation under low-light conditions.

In addition to quantitative comparisons, we also visualize the common 3D LUT and our IA-LUT in Figure 9. Since 4-dimension IA-LUT is difficult to illustrate succinctly, we select out three representative enhancement intensities e and fix them to draw the 3D remainder in IA-LUT. From the visualization, it can be seen that the remainder of IA-LUT models the color transformation more smoothly and correctly than 3D LUT. More importantly, with e increasing, the 3D remainder of IA-LUT that stores the mapping relationships of colors is inclined to be brighter. This observation validates the influence of enhancement intensities on determining

the relatively optimal color transformation, which is in line with our overall design.

Denoising: As we pointed out, LUT is susceptible to noise. However, the real-world videos captured in low-light conditions are often degraded by noises. To this end, it is expected to improve the network performance by adding a denoising module to suppress the noises. In the comparisons between FastLLVE and FastLLVE+dn, a simple denoising module is able to complement the LUT in performance with the evident improvement on all metrics. Note that the denoising module also benefits the evaluation of brightness consistency. One possible reason is that the AB (Var) and MABD are sensitive to noise as well.

Locations of denoising module: Except for setting a denoising module as the post-processing, it is also feasible to construct a denoising module at the beginning of the whole framework as the pre-processing. Neglect to the possibility of enhancing noise, the pre-processing denoising module seems to be more reasonable. However, it is worth noting that there is no feasible way to collect clean low-light videos in real world, which leads to the lack of ground truths for the supervised training of the denoising module as the pre-processing. As a consequence, the denoising module without supervision can affect the video-adaptive IA-LUT to learn incorrect mapping through the end-to-end training of the entire framework. In addition, even though the training principles [17, 25] are specifically designed for supervised training with unpaired

Table 2: The results of ablation study. The best is in red. [Key: 3D LUT: baseline based on 3D LUT, FastLLVE: our framework without denoising module, dn +FastLLVE: our framework with denoising module as the pre-processing, FastLLVE+ dn : our framework with denoising module as the post-processing]

Method	SDSD				SMID				Runtime(s)
	PSNR	SSIM	AB (Var)↓	MABD↓	PSNR	SSIM	AB (Var)↓	MABD↓	
3D LUT	22.76	0.69	0.107	0.265	25.08	0.73	1.024	1.227	0.007
FastLLVE	27.06	0.78	0.038	0.091	26.45	0.75	0.476	0.748	0.013
dn +FastLLVE	24.02	0.81	0.146	0.368	23.85	0.72	1.657	2.872	0.080
FastLLVE+ dn	27.55	0.86	0.033	0.040	27.62	0.80	0.065	0.050	0.080

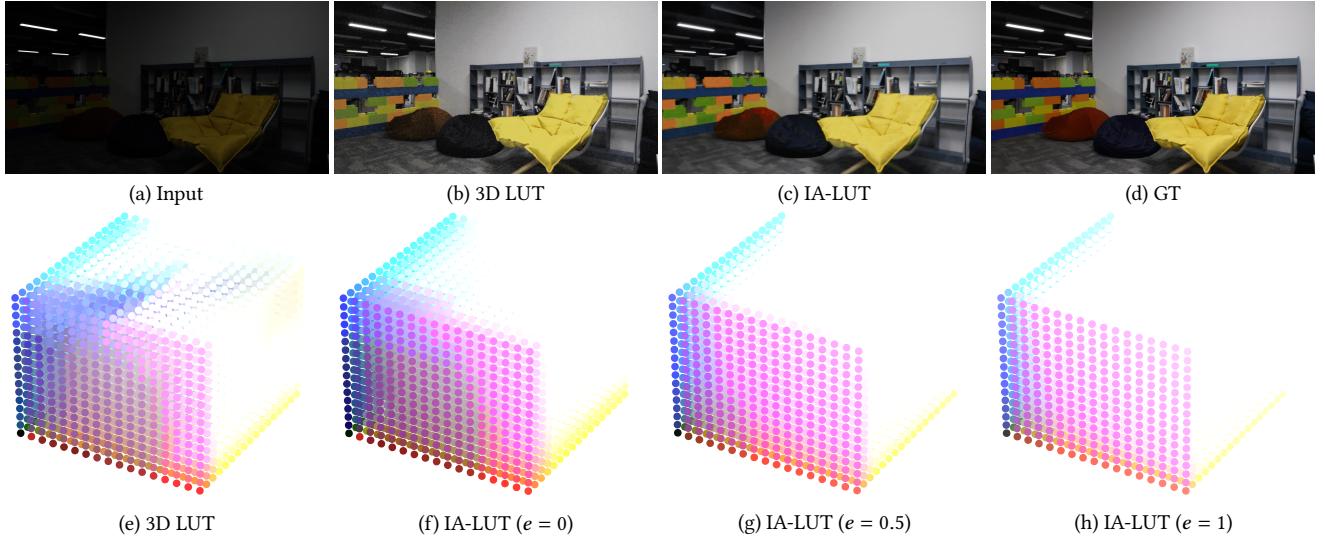


Figure 9: Visualization of the common 3D LUT and the remainder of IA-LUT while the enhancement intensities e are fixed.

noisy data, they may still not be able to completely solve this problem since it is too difficult to find a reliable noise model that enables the simulation in various low-light conditions.

For comparison, we follow the training principle Noiser2Noise [25] to self-supervise the denoising module as the pre-processing, with the low-light noise model from [22] simulating the noise in low-light conditions. As shown in Table 5, FastLLVE+ dn performs much better than dn +FastLLVE on all metrics, which supports the denoising module as the post-processing. Otherwise, due to the unreliable low-light noise model, the training of dn +FastLLVE based on Noiser2Noise is unstable and easy to fail as expected. Therefore, we use the denoising module as the post-processing in our method.

5 LIMITATIONS AND BROADER IMPACT

In this paper, we propose a Intensity-Aware LUT for LLVE and validate its advantages of high efficiency and natural maintenance of inter-frame brightness consistency. However, since the LUT-based methods is commonly susceptible to noise, a from-the-shelf denoising module is leveraged to further improve the visual quality of enhanced normal-light videos, affecting the efficiency of the whole framework. Therefore, a novel denoising strategy suitable for LUT is promising, and we leave it as future work.

Real-time LLVE has potential to bring a significant impact in many ways. On the one hand, it can be applied on camera monitor

system to improve the public safety by increasing the visibility of critical areas, such as streets, parking lots, and transportation, particularly during night-time hours. On the other hand, it can enhance the quality of visual media produced in low-light conditions, such as documentary footage and home videos. This would improve the overall quality of such visual media, making them more engaging and informative.

6 CONCLUSION

We firstly introduce the LUT in LLVE tasks, and propose a novel LUT-based framework, named FastLLVE, for real-time low-light video enhancement. In order to bring the LUT to LLVE, we design the Intensity-Aware LUT (IA-LUT) with a new dimension of enhancement intensity to solve the one-to-many mapping problem. In terms of the flickering effect in the enhanced video, we point out that IA-LUT can naturally maintain the brightness consistency among video frames. Extensive experiments have validated the superiority of the proposed method as compared to the LLVE SOTAs. We envision that this work could facilitate the development of LLVE in practical applications.

ACKNOWLEDGEMENT

This work was supported in part by the Shanghai Pujiang Program under Grant 22PJ1406800.

REFERENCES

- [1] Zohreh Azizi and C-C Jay Kuo. 2022. SALVE: Self-supervised Adaptive Low-light Video Enhancement. *arXiv preprint arXiv:2212.11484* (2022).
- [2] Chen Chen, Qifeng Chen, Minh N Do, and Vladlen Koltun. 2019. Seeing motion in the dark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3185–3194.
- [3] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. 2019. Real-world image denoising with deep boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 12 (2019), 3071–3087.
- [4] Gong Cheng, Peicheng Zhou, and Junwei Han. 2016. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing* 54, 12 (2016), 7405–7415.
- [5] Xueyang Fu, Yinghao Liao, Delu Zeng, Yue Huang, Xiao-Ping Zhang, and Xinghao Ding. 2015. A probabilistic method for image enhancement with simultaneous illumination and reflectance estimation. *IEEE Transactions on Image Processing* 24, 12 (2015), 4965–4977.
- [6] Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping Zhang, and Xinghao Ding. 2016. A weighted variational model for simultaneous reflectance and illumination estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2782–2790.
- [7] Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. *arXiv preprint arXiv:1810.13243* (2018).
- [8] Shan Huang, Xiaohong Liu, Tao Tan, Menghan Hu, Xiaoer Wei, Tingli Chen, and Bin Sheng. 2023. TransMRSR: Transformer-based Self-Distilled Generative Prior for Brain MRI Super-Resolution. *arXiv preprint arXiv:2306.06669* (2023).
- [9] Haidi Ibrahim and Nicholas Sia Pik Kong. 2007. Brightness preserving dynamic histogram equalization for image contrast enhancement. *IEEE Transactions on Consumer Electronics* 53, 4 (2007), 1752–1758.
- [10] Haiyang Jiang and Yinqiang Zheng. 2019. Learning to see moving objects in the dark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7324–7333.
- [11] Hakki Can Karaimer and Michael S Brown. 2016. A software platform for manipulating the camera imaging pipeline. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I* 14. Springer, 429–444.
- [12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [13] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. 2017. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 624–632.
- [14] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. 2018. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence* 41, 11 (2018), 2599–2613.
- [15] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. 2018. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*. 170–185.
- [16] Edwin H Land. 1977. The retinex theory of color vision. *Scientific american* 237, 6 (1977), 108–129.
- [17] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. 2018. Noise2Noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189* (2018).
- [18] Guofu Li, Yifan Yang, Xingda Qu, Dongpu Cao, and Keqiang Li. 2021. A deep learning based image enhancement approach for autonomous driving at night. *Knowledge-Based Systems* 213 (2021), 106617.
- [19] Xiaohong Liu, Lei Chen, Wenyi Wang, and Jiyang Zhao. 2018. Robust multi-frame super-resolution based on spatially weighted half-quadratic estimation and adaptive BTV regularization. *IEEE Transactions on Image Processing* 27, 10 (2018), 4971–4986.
- [20] Xiaohong Liu, Lingshi Kong, Yang Zhou, Jiyang Zhao, and Jun Chen. 2020. End-to-end trainable video super-resolution based on a new mechanism for implicit motion estimation and compensation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2416–2425.
- [21] Xiaohong Liu, Kangdi Shi, Zhe Wang, and Jun Chen. 2021. Exploit camera raw data for video super-resolution via hidden Markov model inference. *IEEE Transactions on Image Processing* 30 (2021), 2127–2140.
- [22] Feifan Lv, Yu Li, and Feng Lu. 2021. Attention guided low-light image enhancement with a large scale low-light simulation dataset. *International Journal of Computer Vision* 129, 7 (2021), 2175–2193.
- [23] Feifan Lv, Feng Lu, Jianhua Wu, and Chongsoon Lim. 2018. MBLLEN: Low-Light Image/Video Enhancement Using CNNs.. In *BMVC*, Vol. 220. 4.
- [24] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. 2016. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Advances in neural information processing systems* 29 (2016).
- [25] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. 2020. Noisier2noise: Learning to denoise from unpaired noisy data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12064–12072.
- [26] Sean Moran, Pierre Marza, Steven McDonagh, Sarah Parisot, and Gregory Slabaugh. 2020. DeepLpf: Deep local parametric filters for image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12826–12835.
- [27] Keita Nakai, Yoshikatsu Hoshi, and Akira Taguchi. 2013. Color image contrast enhancement method based on differential intensity/saturation gray-levels histograms. In *2013 International Symposium on Intelligent Signal Processing and Communication Systems*. IEEE, 445–449.
- [28] Jinwang Pan, Deming Zhai, Yunchao Bai, Junjun Jiang, Debin Zhao, and Xianming Liu. 2022. ChebyLighter: Optimal Curve Estimation for Low-light Image Enhancement. In *Proceedings of the 30th ACM International Conference on Multimedia* 1358–1366.
- [29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
- [30] Haoyu Ren, Mostafa El-Khamy, and Jungwon Lee. 2019. Dn-resnet: Efficient deep residual network for image denoising. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part V* 14. Springer, 215–230.
- [31] Sourav Samanta, Amartya Mukherjee, Amira S Ashour, Nilanjan Dey, João Manuel RS Tavares, Wahiba Ben Abdessalem Karâa, Redha Taïar, Ahmad Taher Azar, and Aboul Ella Hassanien. 2018. Log transform based optimal image enhancement using firefly algorithm for autonomous mini unmanned aerial vehicle: An application of aerial photography. *International Journal of Image and Graphics* 18, 04 (2018), 1850019.
- [32] Zhihao Shi, Xiaohong Liu, Chengqi Li, Linhui Dai, Jun Chen, Timothy N Davidson, and Jiyang Zhao. 2021. Learning for unconstrained space-time video super-resolution. *IEEE Transactions on Broadcasting* 68, 2 (2021), 345–358.
- [33] Zhihao Shi, Xiaohong Liu, Kangdi Shi, Linhui Dai, and Jun Chen. 2021. Video frame interpolation via generalized deformable convolution. *IEEE transactions on multimedia* 24 (2021), 426–439.
- [34] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. 2022. Video frame interpolation transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17482–17491.
- [35] Matias Tassano, Julie Delon, and Thomas Veit. 2020. Fastvdnet: Towards real-time deep video denoising without flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1354–1363.
- [36] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022* (2016).
- [37] Qing Wang and Rabab K Ward. 2007. Fast image/video contrast enhancement based on weighted thresholded histogram equalization. *IEEE transactions on Consumer Electronics* 53, 2 (2007), 757–764.
- [38] Ruixing Wang, Xiaogang Xu, Chi-Wing Fu, Jiangbo Lu, Bei Yu, and Jiaya Jia. 2021. Seeing dynamic scene in the dark: A high-quality video dataset with mechatronic alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9700–9709.
- [39] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. 2019. Underexposed photo enhancement using deep illumination estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6849–6857.
- [40] Shuhang Wang, Jin Zheng, Hai-Miao Hu, and Bo Li. 2013. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE transactions on image processing* 22, 9 (2013), 3538–3548.
- [41] Tao Wang, Yong Li, Jingyang Peng, Yipeng Ma, Xian Wang, Fenglong Song, and Youliang Yan. 2021. Real-time image enhancer via learnable spatial-aware 3d lookup tables. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2471–2480.
- [42] Guangyang Wu, Xiaohong Liu, Kunming Luo, Xi Liu, Qingqing Zheng, Shuaicheng Liu, Xinyang Jiang, Guangtao Zhai, and Wenyi Wang. 2023. AccFlow: Backward Accumulation for Long-Range Optical Flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.
- [43] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853* (2015).
- [44] Canqian Yang, Meiguang Jin, Xu Jia, Yi Xu, and Ying Chen. 2022. AdaInt: learning adaptive intervals for 3D lookup tables on real-time image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17522–17531.
- [45] Canqian Yang, Meiguang Jin, Yi Xu, Rui Zhang, Ying Chen, and Huaida Liu. 2022. SepLUT: Separable Image-Adaptive Lookup Tables for Real-Time Image Enhancement. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII*. Springer, 201–217.

- [46] Meifang Yang, Xin Nie, and Ryan Wen Liu. 2019. Coarse-to-fine luminance estimation for low-light image enhancement in maritime video surveillance. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 299–304.
- [47] Wenhan Yang, Shiqi Wang, Yuming Fang, Yue Wang, and Jiaying Liu. 2020. From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3063–3072.
- [48] Guanghao Yin, Zefan Qu, Xinyang Jiang, Shan Jiang, Zhenhua Han, Ningxin Zheng, Xiaohong Liu, Huan Yang, Yaqing Yang, Dongsheng Li, and Lili Qiu. 2023. Online Video Streaming Super-Resolution with Adaptive Look-Up Table Fusion. *arXiv preprint arXiv:2303.00334* (2023).
- [49] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I* 13. Springer, 818–833.
- [50] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. 2020. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 4 (2020), 2058–2073.
- [51] Fan Zhang, Yu Li, Shaodi You, and Ying Fu. 2021. Learning temporal consistency for low light video enhancement from single images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4967–4976.
- [52] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. 2017. Beyond a gaussian denoiser: Residual learning of deep cnns for image denoising. *IEEE transactions on image processing* 26, 7 (2017), 3142–3155.
- [53] Shangchen Zhou, Chongyi Li, and Chen Change Loy. 2022. Lednet: Joint low-light enhancement and deblurring in the dark. In *European Conference on Computer Vision*. Springer, 573–589.

A IMPLEMENTATION DETAILS

Network Structure. As illustrated in Table 3, the lightweight encoder is comprised of five convolutional blocks that perform down-sampling on each frame of the input video, thereby reducing the resolution to 1/32. Each convolutional block consists of a 3D convolution layer, a leaky ReLU [43] activation function, and an instance normalization [36] layer. Besides, the corresponding lightweight decoder consists of five deconvolutional blocks that restore each frame of the encoder output to the original resolution through up-sampling. The deconvolutional block has the same structure as the convolutional block, with the exception that the convolution layer is replaced by a deconvolution [49] layer. It is worth noting that all instance normalization layers used in the network have learnable parameters, and the last deconvolutional block that directly outputs the intensity map is devoid of the leaky ReLU activation function and instance normalization layer.

In addition, before the two fully-connected layers, which achieve the mapping h from the compact feature vector γ to the video-adaptive IA-LUT, the latent features extracted from the encoder are first subjected to an average pooling operation, followed by a dropout operation with the dropout rate set to 0.5. Subsequently, the features are reshaped to obtain the feature vector γ . In terms of these operations as the pre-processing before weight predictor, the average pooling operation serves to reduce the parameters of the first fully-connected layer, and the dropout operation aims at enhancing training data, instead of just avoiding over-fitting that is the common role of dropout operation.

As for the denoising module, in this paper, we follow the practice in [3] to design the additional denoising module. Compared with the original DDFN approach, we decrease the number of feature integration blocks to one, thereby reducing the size of the denoising module. Moreover, in order to effectively process the enhanced videos, we leverage 3D convolution layers to implement the denoising module. However, notably, it is also feasible to process

each frame recursively with a denoising method of single image processing.

Quadrilinear Interpolation. Although we have presented a detailed definition of IA-LUT and its corresponding quadrilinear interpolation in this paper, the calculation formula for the offset O_x is omitted due to the space limitation. Therefore, we supplement the calculation formula here, which can be expressed as:

$$O_x = O_r \cdot O_g \cdot O_b \cdot O_e, \quad (10)$$

where we have the four terms

$$\begin{aligned} O_r &= \left\{ \frac{r - r_i}{r_{i+1} - r_i}, \frac{r_{i+1} - r}{r_{i+1} - r_i} \right\}, \quad O_g = \left\{ \frac{g - g_i}{g_{i+1} - g_i}, \frac{g_{i+1} - g}{g_{i+1} - g_i} \right\}, \\ O_b &= \left\{ \frac{b - b_i}{b_{i+1} - b_i}, \frac{b_{i+1} - b}{b_{i+1} - b_i} \right\}, \quad O_e = \left\{ \frac{e - e_i}{e_{i+1} - e_i}, \frac{e_{i+1} - e}{e_{i+1} - e_i} \right\}, \end{aligned} \quad (11)$$

where (r, g, b, e) denotes the input index, and $C_x = [r_i, g_i, b_i, e_m]$ stands for the index of grid point $x = [i, j, k, m]$. All the terms appeared above have been defined in the paper. By utilizing this calculation formula, we are able to compute the 16 offsets required during the quadrilinear interpolation, respectively.

Loss Function. In addition to the loss between the output of our method and the ground truth, we also introduce 4D smooth regularization and 4D monotonicity regularization adapted to the IA-LUT into the loss function. The 4D smooth regularization is designed to ensure the stability of the conversion from the input space to the mapped color space, which helps avoid artifacts caused by extreme color changes in the IA-LUT. It consists of two parts which correspond to the video-adaptive IA-LUT and the video-dependent weights, respectively. Firstly, we have the 4D smooth regularization

$$l_s = l_{lut} + l_w, \quad (12)$$

where l_{lut} denotes the part related to the video-adaptive IA-LUT, and l_w indicates the other part related to the video-dependent weights. For brevity, we here let

$$\begin{aligned} x_i &= [i + 1, j, k, m], \quad x_j = [i, j + 1, k, m], \\ x_k &= [i, j, k + 1, m], \quad x_m = [i, j, k, m + 1], \end{aligned} \quad (13)$$

as the four grid points obtained by increasing the index of grid point $x = [i, j, k, m]$ by one unit length along each of the four dimensions. Then, we define the function \mathcal{F} as:

$$\begin{aligned} \mathcal{F}(i, j, k, m) &= \left\| C'_{x_i} - C'_x \right\|^2 + \left\| C'_{x_j} - C'_x \right\|^2 \\ &\quad + \left\| C'_{x_k} - C'_x \right\|^2 + \left\| C'_{x_m} - C'_x \right\|^2, \end{aligned} \quad (14)$$

where $C'_x = [r_x, g_x, b_x]$ denotes the stored values in IA-LUT for grid point x . Thus, we have

$$l_s = \sum_{i,j,k,m} \mathcal{F}(i, j, k, m) + \sum_{n=1}^T \|w_n\|^2, \quad (15)$$

where $w_n, n \in [1, 2, \dots, T]$ denotes the T video-dependent weights.

Table 3: Architecture of the encoder-decoder network, where "nf" is a hyper-parameter that serves as a channel multiplier controlling the width of each convolution layer. In this paper, the "nf" is set to 8.

Id	Encoder		Decoder	
	Layer	Output Shape	Layer	Output Shape
0	Conv $3 \times 3 \times 3$, Leaky ReLU	$nf \times H/2 \times W/2$	Deconv $3 \times 3 \times 3$, Leaky ReLU	$8nf \times H/32 \times W/32$
1	InstanceNorm	$nf \times H/2 \times W/2$	InstanceNorm	$8nf \times H/32 \times W/32$
2	Conv $3 \times 3 \times 3$, Leaky ReLU	$2nf \times H/4 \times W/4$	Deconv $3 \times 3 \times 3$, Leaky ReLU	$4nf \times H/8 \times W/8$
3	InstanceNorm	$2nf \times H/4 \times W/4$	InstanceNorm	$4nf \times H/8 \times W/8$
4	Conv $3 \times 3 \times 3$, Leaky ReLU	$4nf \times H/8 \times W/8$	Deconv $3 \times 3 \times 3$, Leaky ReLU	$2nf \times H/4 \times W/4$
5	InstanceNorm	$4nf \times H/8 \times W/8$	InstanceNorm	$2nf \times H/4 \times W/4$
6	Conv $3 \times 3 \times 3$, Leaky ReLU	$8nf \times H/16 \times W/16$	Deconv $3 \times 3 \times 3$, Leaky ReLU	$nf \times H/2 \times W/2$
7	InstanceNorm	$8nf \times H/16 \times W/16$	InstanceNorm	$nf \times H/2 \times W/2$
8	Conv $3 \times 3 \times 3$, Leaky ReLU	$8nf \times H/32 \times W/32$	Deconv $3 \times 3 \times 3$, Leaky ReLU	$1 \times H \times W$
9	InstanceNorm	$8nf \times H/32 \times W/32$	\	\

Table 4: Quantitative results of FastLLVE with different numbers of L on SDSD test dataset. The best is in red.

L	9	17	33	64
PSNR	25.10	26.42	27.06	24.52
SSIM	0.7717	0.7736	0.7769	0.7358

As for the other regularization loss l_m , we let $\mathcal{M}(a) = \max(a, 0)$ and define the function \mathcal{G} as:

$$\begin{aligned} \mathcal{G}(i, j, k, m) &= \mathcal{M}(r_{x_i} - r_x) + \mathcal{M}(g_{x_i} - g_x) + \mathcal{M}(b_{x_i} - b_x) \\ &+ \mathcal{M}(r_{x_j} - r_x) + \mathcal{M}(g_{x_j} - g_x) + \mathcal{M}(b_{x_j} - b_x) \\ &+ \mathcal{M}(r_{x_k} - r_x) + \mathcal{M}(g_{x_k} - g_x) + \mathcal{M}(b_{x_k} - b_x) \\ &+ \mathcal{M}(r_{x_m} - r_x) + \mathcal{M}(g_{x_m} - g_x) + \mathcal{M}(b_{x_m} - b_x). \end{aligned} \quad (16)$$

So we can express the calculation of l_m as:

$$l_m = \sum_{i,j,k,m} \mathcal{G}(i, j, k, m). \quad (17)$$

B EXPLORATION OF HYPER-PARAMETERS

In terms of the number of grid points and the number of basis IA-LUTs, which determine the precision of the color transformation modeled by the generated LUT, their values cannot be increased indiscriminately due to the size limitation of IA-LUT. Although in this paper, we follow the most widely-used setting [44, 45, 50] of the two hyper-parameters L and T , further experiments are needed to confirm the validity of this selected setting.

To explore the influence of the number of grid points on the enhancement effect, as shown in Table 4, we divide the IA-LUT into different numbers of unit lattices (i.e., different numbers of grid points). As expected, increasing L from 9 to 33 improves the performance of our method. However, note that degraded performance is observed with the value of L more than 33. One possible reason is that too high precision may result in over-fitting. Additionally, a large number of grid points also leads to heavy memory burden and great training difficulty which we should avoid. Therefore, with the consideration of avoiding over-fitting and reducing the size of

Table 5: Quantitative results of FastLLVE with different numbers of T on SDSD test dataset. The best is in red.

T	1	2	3	4
PSNR	26.40	26.58	27.06	25.23
SSIM	0.7539	0.7648	0.7769	0.7641

Table 6: Quantitative results of FastLLVE+dn with different numbers of N on SDSD test dataset. The best is in red.

T	5	7	9
PSNR	26.51	27.55	26.12
SSIM	0.851	0.855	0.849
AB (Var)↓	0.032	0.033	0.052
MABD↓	0.049	0.040	0.055

IA-LUT, setting the number of grid points to 33 is indeed the one of the optimal choices.

Similarly, to explore the influence of the number of basis IA-LUTs on the enhancement effect, as shown in Table 5, we use different numbers of basis IA-LUTs to fuse the video-adaptive IA-LUTs. It can be observed that the performance of our method is positively correlated with the value of T . Then, similar to the above experiment, degraded performance appears when $T = 4$, possibly due to the over-fitting. Besides, there is also a need for reducing the parameters of the fully connected layers. As a consequence, it may not be a good choice to further increase the number of basis IA-LUTs, compared with setting the value of T to 3.

Finally, we consider the influence of the number of input frames on the enhancement effect. Therefore, we conduct an experiment, using different numbers of input frames, to investigate its effect on performance of the model. According to the Table 6, our method achieves its optimal performance when N is set to 7. Notably, increasing N beyond this value leads to performance degradation, possibly attributable to challenges encountered by the model in achieving convergence. As a result, the experimental result suggests 7 input frames.