APSIPA Transactions on Signal and Information Processing, 2023, 12, e102

Original Paper

SALVE: Self-Supervised Adaptive Low-Light Video Enhancement

Zohreh Azizi* and C.-C. Jay Kuo

University of Southern California, Los Angeles, California, USA

ABSTRACT

A self-supervised adaptive low-light video enhancement method, called SALVE, is proposed in this work. SALVE first enhances a few keyframes of an input low-light video using a retinex-based low-light image enhancement technique. For each keyframe, it learns a mapping from low-light image patches to enhanced ones via ridge regression. These mappings are then used to enhance the remaining frames in the low-light video. The combination of traditional retinex-based image enhancement and learning-based ridge regression leads to a robust, adaptive and computationally inexpensive solution to enhance low-light videos. Our extensive experiments along with a user study show that 87% of participants prefer SALVE over prior work. Our codes are available at: https://github.com/zohrehazizi/SALVE.

Keywords: Low light image enhancement, Low light video enhancement, Retinex model.

1 Introduction

Videos captured under low light conditions are often noisy and of poor visibility. Low-light video enhancement aims to improve viewers' experience by increasing

*Corresponding author: Zohreh Azizi, zazizi@usc.edu.

Received 21 December 2022; Revised 26 April 2023
ISSN 2048-7703; DOI 10.1561/116.00000085

brightness, suppressing noise, and amplifying detailed texture. The performance of computer vision tasks such as object tracking and face recognition can be severely affected under low-light noisy environments. Hence, low-light video enhancement is needed to ensure the robustness of computer vision systems. Besides, the technology is highly demanded in consumer electronics such as video capturing by smart phones.

While mature methods for low-light *image* enhancement have been developed in recent years, low light *video* enhancement is still a standing challenge and open for further improvement. A trivial solution to low light video enhancement is to enhance each frame with an image enhancement method independently. However, since this solution disregards temporal consistency, it tends to result in flickering videos [19]. Also, frame-by-frame low light video processing can be too computationally expensive for practical applications.

Several methods utilized deep learning (DL) to preserve the temporal consistency of video frames. For instance, 3D CNNs are trained to process a number of frames simultaneously in order to take temporal consistency into account [14, 27]. Some papers enforce similarity between pairs of frames with a temporal loss function or loss function regularization in training [6, 9]. Other works extract the motion information and leverage redundancy among frames to ensure the temporal consistency of enhanced videos [19, 38].

On one hand, the efforts mentioned above lead to high-performance models with a range of acceptable to excellent quality results. On the other hand, their performance is dependent on the training dataset. Differences between the training and testing environments can degrade the performance of low light video enhancement severely. In other words, when deployed in the real world, the DL-based models cannot be trusted and utilized without fine-tuning. Considering the fact that paired low-light/normal-light video datasets are very scarce, fine-tuning these models can be challenging.

In this paper, we propose an alternative low-light video enhancement method to address the above-mentioned challenges. Our proposed method is called the self-supervised adaptive low-light video enhancement (SALVE) method. By self-supervision, we mean that SALVE directly learns to enhance an arbitrary input video without requiring to be trained on other training videos.

SALVE offers a robust solution that is highly adaptive to new real-world conditions. SALVE selects a couple of keyframes from the input video and enhances them using an effective retinex-based image enhancement method called NATLE [2]. Given NATLE-enhanced keyframes of the input video, SALVE learns a mapping from low-light frames to enhanced ones via ridge regression. Finally, SALVE uses this mapping to enhance the remaining frames. SALVE does not need low- and normal-light paired videos in training. Therefore, it can be an attractive choice for non-public environments such as warehouses and diversified environments captured by phone cameras.

SALVE is a hybrid method that combines components from a retinex-based image enhancement method and a learning-based method. The former component leads to a robust solution which is highly adaptive to new real-world environments. The latter component offers a fast, computationally inexpensive and temporally consistent solution. We conduct extensive experiments to show the superior performance of SALVE. Our user study shows that 87% of participants prefer SALVE over prior work.

The rest of this paper is organized as follows. Related work is discussed in Section 2. The low light image enhancement method, NATLE, is reviewed and then the proposed low light video enhancement method, SALVE, is explained in Section 3. Experimental results are presented in Section 4. Finally, concluding remarks are given in Section 5.

2 Related Work

2.1 Low Light Image Enhancement

There are two categories of traditional low-light image enhancement methods: histogram equalization and retinex decomposition. Histogram equalization stretches the color histogram to increase the image contrast. Although it is simple and fast, it often yields unnatural colors, amplifies noise, and under/over-exposes areas inside an image. To address these artifacts, more complex priors are adopted for histogram-based image enhancement [1, 5, 12, 22, 28]. Specific penalty terms were designed and used to control the level of contrast enhancement, noise, and mean brightness in [1]. Inter-pixel contextual information was used for non-linear data mapping in [5]. To preserve the mean brightness, histogram equalization was applied to different dynamic ranges of a smoothed image in [12]. The gray-level differences between adjacent pixels were amplified to enhance image contrast based on layered difference representation of 2D histograms in [22]. Differential gray-level histogram equalization was proposed in [28] based on the concept of differential histograms.

Inspired by the human vision system (HVS), it is assumed in the retinex theory [20] that each image can be decomposed into two components: a reflectance (R) term containing inherent properties and an illumination (L) term containing the lightness condition. Along this line, another approach for low light image enhancement is to decompose an input image into R and L terms and adjust the L term to the normal-light condition. Earlier work focused on R and L decomposition and attempted to acquire R and L more accurately [16, 17] using a Single-Scale Retinex (SSR) representation. Later, SSR was extended to a MultiScale Retinex (MSR) representation, which can be used for color image restoration. An adaptive MSR was proposed in [21], which computes the weights of an SSR according to the content of the input

image. More recently, optimization functions were carefully designed in [2, 30] to determine the R and L terms. They attempted to find a balance in suppressing noise and preserving texture through the optimization functions.

Recently, the deep-learning (DL) paradigm has been proposed for low-light image enhancement based on retinex theory [34, 37, 40]. A decomposition network and an illumination network were trained to perform retinex decomposition and enhancement, respectively, in [37]. The work in [40] added another network, called reflectance restoration, to mitigate color distortion and noise. A generative adversarial network (GAN) [10] was employed in [34] to generate decomposed and enhanced images. Another GAN work [15] was trained without paired data. The application of auto-encoders to image enhancement was investigated in [25]. A multi-branch network was proposed in [27] to extract rich features in different levels for enhancement via multiple subnets. An end-to-end network for raw camera image enhancement was proposed in [7].

2.2 Low Light Video Enhancement

While low-light image enhancement is a well-studied topic, low-light video enhancement is still an ongoing and challenging research topic. Applying image-based algorithms to each frame of a video yields flickering artifacts due to inconsistent enhancement results along time [19]. It is essential to take both temporal and spatial information into account in video processing. One approach is to extend 2D convolutional neural networks (CNNs) to 3D CNNs [27], which includes the 2D spatial domain and the 1D temporal domain. A 3D U-net [31] was proposed in [14] to enhance raw camera images. However, these 3D DL-based methods have huge model sizes and extremely high computational costs.

Another approach is to exploit self-consistency [6, 9]. The resulting methods operate on single frames of video but impose the similarity constraint on image pairs to improve the performance and stability of their models. A new static video dataset was proposed in [6], containing short- and long-exposure images of the same scene. They took two random frames from the same sequence in training and utilized the self-consistency temporal loss to make the network robust against noise and small changes in the scene. Different motion types were accounted for by imposing temporal stability using a regularized cost function in [9]. Another family of self-consistency-based methods [19, 38] used the optical flow to estimate the motion information in a sequence. They utilized the FlowNet [13] to predict the optical flow between two frames, and warped the frames based on the predicted flow to avoid inconsistent frame processing. An image segmentation network was exploited to detect the moving object regions before optical flow prediction in [38]. A model to reduce noise and estimate illumination was proposed in [35] based on the retinex theory.

It took each frame along with two past and two future frames as input to enhance the middle frame.

All existing methods on low-light video enhancement employ deep neural networks (DNNs) as their backbone. In this work, we propose an effective and high performance method called SALVE to achieve the same goal without the use of DNNs. Our method contributes to green video processing with a lower carbon footprint [3, 18, 32]. Additionally, SALVE does not need a training dataset; it is a self-supervised approach which utilizes the frames of the test video and adapts its enhancement strategy accordingly. As such, our approach does not rely on massive training datasets and is robust against environmental changes.

3 Proposed Method

Figure 1 presents an overview of our proposed method. The top row shows the steps taken to enhance an input frame, which we discuss in Section 3.1. The bottom row shows the extension to videos, i.e. it shows how we treat different frames of the video. We discuss this process in Section 3.2.

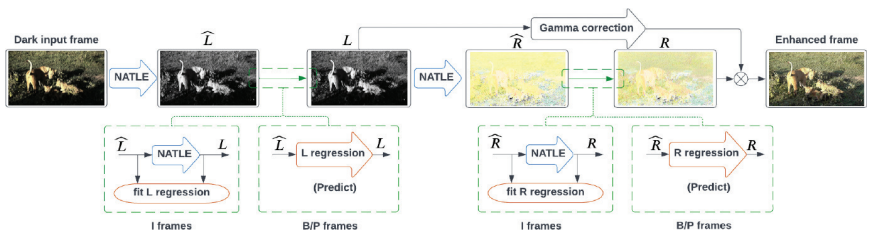


Figure 1: An overview of the proposed SALVE method. For intra-coded frames (I frames), it estimates the illumination ($\widehat{L}$) component and the reflectance ($\widehat{R}$) components using the NATLE method. For inter-coded frames (P/B frames), it predicts these components using a ridge regression learned from the last low-light and enhanced I frame pairs.

3.1 NATLE

In order to propose our method in Section 3.2, we need to first review NATLE [2], which is an effective method for low light image enhancement. NATLE is a retinex-based low light image enhancement method. A classic retinex model decomposes an input image S into the element-wise multiplication of two components; a reflectance map (R) and an illumination (L) map:

$$S = R \circ L, \tag{1}$$

where R represents the inherent features within an image which remain the same in different lightness conditions. L shows the lightning condition. Ideally,

R contains all the texture and details within the image and L is a piece-wise smooth map with significant edges. NATLE presents a methodology to find solutions for R and L. It then enhances L to a normal light condition, and follows the retinex model to combine the enhanced L with R and obtain the enhanced image. In what follows, we explain the steps NATLE takes to find solutions for R and L.

Step 1. Calculate an initial estimation of L, namely $\widehat{L}$, which is a weighted average of RGB color channels:

$$\widehat{L} = 0.299R + 0.587G + 0.114B. \tag{2}$$

Step 2. Form an optimization function to find a piece-wise smooth solution for L as

$$\arg\min_{L} \|L - \widehat{L}\|_F^2 + \alpha\|\nabla L\|_1, \tag{3}$$

where α is set to 0.015 and $\|\nabla L\|_1$ is approximated with

$$\lim_{\epsilon \to 0^+} \sum_{x} \sum_{d \in \{h,v\}} \frac{(\nabla_d L(x))^2}{\mid \nabla_d \widehat{L}(x) \mid + \epsilon} = \|\nabla L\|_1, \tag{4}$$

and where d is the gradient direction and v and h indicate the vertical and horizontal directions, respectively. Eq. (3) can be rewritten as

$$\arg\min_{L} \|L - \widehat{L}\|_F^2 + \sum_{x} \sum_{d \in \{h,v\}} A_d(x)(\nabla_d L(x))^2, \tag{5}$$

where

$$A_d(x) = \frac{\alpha}{\mid \nabla_d \widehat{L}(x) \mid + \epsilon}. \tag{6}$$

Finally, Eq. (5) is solved by differentiating with respect to L and setting the derivative to zero. The final solution is derived in closed form as

$$l = (I + \sum_{d \in \{h,v\}} D_d^T \mathrm{D}iag(a_d) D_d)^{-1} \widehat{l}, \tag{7}$$

where $\mathrm{Diag}(a_d)$ is a matrix with a_d on its diagonal, D_d is a discrete differential operator matrix that plays the role of ∇ and I is the identity matrix. Once vector l is determined, it is reshaped to matrix L.

Step 3. Calculate an estimate of R in form of

$$\widehat{R} = S \oslash (L + \varepsilon) - N, \tag{8}$$

where S is the input image, L is the estimated illumination obtained in Step 2, $-N$ shows a median filter denoising followed by a bilateral filter denoising,

$\oslash$ denotes element-wise division and ε is a small value to prevent division by zero.

Step 4. Form an optimization function to find R via

$$\arg\min_{R} \|R - \widehat{R}\|_F^2 + \beta\|\nabla R - G\|_F^2, \tag{9}$$

where β is set to 3 in our experiments. The first term in Eq. (9) ensures that R is noise-free and consistent with the retinex model. The second term has a noise-removal and texture-preserving dual role. Then, we get

$$G = \begin{cases} 0, & \nabla S < \epsilon_g \\ \lambda \nabla S, & \end{cases} \tag{10}$$

where ϵ_g is the threshold to filter out small gradients, which are viewed as noise, and λ controls the degree of texture amplification. The values of ϵ_g and λ are set to 0.05 and 1.1, respectively. Finally, the optimization problem in Eq. (9) is solved by differentiating with respect to R and setting the derivative to zero. The final solution can be derived in closed form as

$$r = (I + \beta \sum_{d\in\{h,v\}} D_d^2)^{-1}(\widehat{r} + \beta \sum_{d\in\{h,v\}}, D_d^T g_d), \tag{11}$$

where D_d, ∇ and I are defined the same as those in Eq. (7). Once vector r is determined, it is reformed to matrix R.

Step 5. Apply gamma correction to L for illumination adjustment. The ultimate enhanced image is computed as

$$S^{'} = R \circ L^{\frac{1}{\gamma}}, \tag{12}$$

where γ is set to 2.2 in the experiment.

3.2 Video Enhancement

We showed the performance of NATLE in low light image enhancement in [2]. NATLE suppresses noise and preserves texture while enhancing low-light images. In order to extend the application of NATLE to videos, the trivial idea would be to apply NATLE separately on all the frames within a video. However, a series of consecutive frames within a video usually have significant correlations in structure, color, and light. We may leverage this temporal similarity in order to lower the costs of the video enhancement from a series of repetitive image enhancements.

Here, we propose a self-supervised method for low light video enhancement based on learning from NATLE. By applying NATLE on selected frames within a video, we acquire pairs of low-light and enhanced frames, from which

we learn a mapping from low-light to enhanced frames. We then apply the learnt mapping to the rest of the frames to accomplish the low light video enhancement. In particular, we approximate Eqs. (7) and (11) in NATLE, which take the major portion of NATLEś runtime. Thus, our proposed video enhancement method is significantly faster and computationally less expensive than applying frame-by-frame NATLE.

In order to decide the frame on which we apply NATLE, we use the FFMPEG compression technique. In FFMPEG, there are three types of frames, namely identity (I), bidirectional (B), and predicted (P) frames. The I frames are the keyframes which indicate a significant spatial or temporal change within the video. More precisely, an I frame is placed where one of the following conditions is met:

- The frame remarkably differs from the previous frame.
- One second has passed from the previous I frame.

We explain the steps to obtain enhanced video frames using SALVE below.

Step 1. Apply NATLE to an I frame:

$$I_{enhanced}, \widehat{L}_I, L_I, \widehat{R}_I, R_I = NATLE(I), \tag{13}$$

where I and $I_{enhanced}$ are the low-light and enhanced keyframes, respectively. The rest of the parameters, i.e., $\widehat{L}_I$, L_I, $\widehat{R}_I$ and R_I, are the results of intermediate steps in NATLE as described in Section 3.1.

Step 2. Learn two ridge regressions mapping $\widehat{L}_I$ and $\widehat{R}_I$ to L_I and R_I, respectively. To be more precise, we look for W_L and W_R to solve the following optimization problems:

$$\min_{W_l} ||l_I - \widehat{l}_I W_l||_2^2 + \alpha ||W_l||_2^2, \tag{14}$$

$$\min_{W_r} ||r_I - \widehat{r}_I W_r||_2^2 + \alpha ||W_r||_2^2, \tag{15}$$

where $l_I \in \mathbb{R}^{n\times 1}$ is the vectorized form of L_I with n pixels. $\widehat{l}_I \in \mathbb{R}^{n\times 25}$ denotes 5×5 neighborhoods of each pixel in $\widehat{L}_I$. The solution, $W_l \in \mathbb{R}^{25\times 1}$, maps each 5×5 patch in $\widehat{L}_I$ to the corresponding center pixel in L_I. The same process and notation is used for Eq. (15).

Step 3. Compute $\widehat{L}_P$ of B/P frames using Eq. (2). Obtaining $\widehat{L}$ by NATLE is computationally inexpensive. Hence, we keep it the same while enhancing B/P frames.

Step 4. Compute L_P using the ridge regressor W_l learned in Step 2:

$$l_P = \widehat{l}_P W_l, \tag{16}$$

where $\widehat{l}_P \in \mathbb{R}^{n\times 25}$ denotes 5×5 neighborhoods of each pixel in $\widehat{L}_P$. $l_P \in \mathbb{R}^{n\times 1}$ is the vectorized form of L_P with n pixels. We reshape l_P vector to obtain L_P matrix.

Step 5. Compute $\widehat{R}_P$ for B/P frames using Eq. (8).

Step 6. Compute R_P using the ridge regressor W_r learned in Step 2; namely,

$$r_P = \widehat{r}_P W_r, \tag{17}$$

where $\widehat{r}_P \in \mathbb{R}^{n\times 25}$ denotes a neighborhood window of size 5×5 of each pixel in $\widehat{R}_P$. $r_P \in \mathbb{R}^{n\times 1}$ is the vectorized form of R_P with n pixels. We reshape r_P to obtain R_P.

Step 7. Apply gamma correction to L_P for illumination adjustment. The final enhanced B/P frame is computed using Eq. (12).

We perform Steps 1 and 2 on the I frames and Steps 3 to 7 on the subsequent B/P frames. Once a new I frame is encountered, we repeat Steps 1 and 2 and continue. This setting ensures that the self-supervised learning from NATLE is being updated frequently enough to keep up with any significant temporal changes.

4 Experiments

4.1 Experimental Setup

We conduct extensive qualitative and quantitative experiments to evaluate our method and show its effectiveness. In our experiments, we use the DAVIS video dataset [4, 29] as the ground truth. DAVIS offers two resolutions, 480P and full resolution. We use all full resolution videos from 2017 and 2019 challenges. Following [26], we synthesize dark videos by darkening the normal-light frames of DAVIS dataset with gamma correction and linear scaling:

$$x = \mathcal{B} \times (\mathcal{A} \times y)^{\gamma}, \tag{18}$$

where y is the ground-truth (normal light) frame, x is the darkened frame, $\mathcal{A}$ and $\mathcal{B}$ are linear scaling factors and sampled from uniform distributions $U(0.9, 1)$ and $U(0.5, 1)$, respectively, and γ is the gamma correction factor which is sampled from $U(2, 3.5)$.

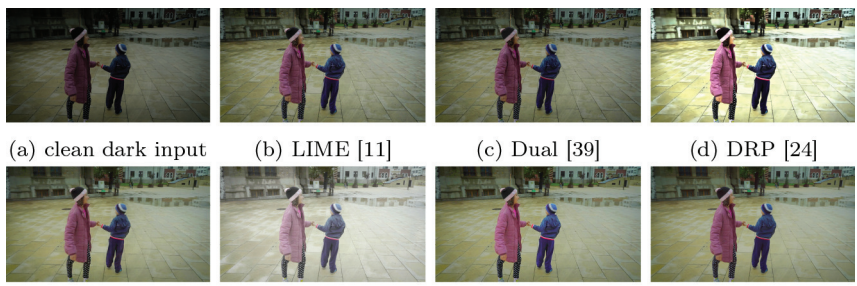
We also synthesize the noisy version of the dark frames. Following [38], we add both Poisson and Gaussian noise to the low-light frames:

$$n = \mathcal{P}(\sigma_p) + \mathcal{N}(\sigma_g), \tag{19}$$

where σ_p and σ_g are parameters of Poisson noise and Gaussian noise, respectively. They are both sampled from a uniform distribution $U(0.01, 0.04)$. We acquire two sets of videos, namely clean-dark and noisy-dark videos. Our goal is to enhance these videos and assess them qualitatively and quantitatively.

4.2 Visual Comparison

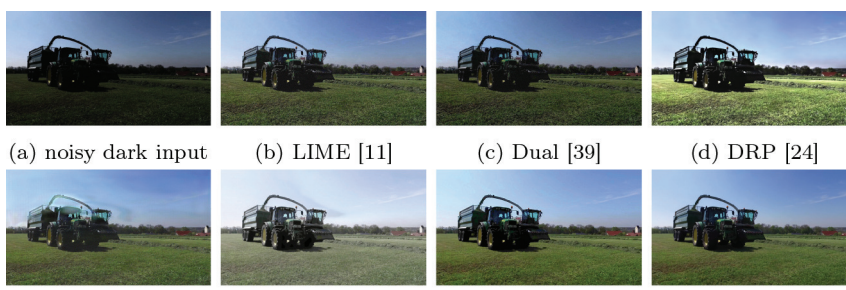
We first provide visual analysis on an exemplary video frame from the clean-dark and noisy-dark datasets in Figures 2 and 3, respectively. We observe that methods not based on deep learning (LIME and DUAL) do not add artifacts to the frame, but the resulting enhanced frame still lacks lightness. Among prior methods based on deep learning, DRP [24] is a self-supervised method that gives nice enhancement results. While DRP adds colorful textures to the enhanced images, the results tend to be slighly different with the ground truth and have artifacts in some regions. SDSD [35] is a supervised method which is fine-tuned to the DAVIS dataset. SDSD tends to add artifacts to enhanced images. This issue is more noticable in Figure 3. StableLLVE [38] is a supervised method trained on the DAVIS dataset. The enhancement results by StableLLVE have a pale color. Our method achieves enhanced frames that are fairly close to the ground-truth and avoids adding artifacts or changing the coloring of the image.



(a) clean dark input (b) LIME [11] (c) Dual [39] (d) DRP [24]

(e) SDSD [35] (f) StableLLVE [38] (g) Ours (h) Ground-truth

Figure 2: Visual comparison between our method and prior work on clean-darkened video frames from DAVIS dataset.



(a) noisy dark input (b) LIME [11] (c) Dual [39] (d) DRP [24]

(e) SDSD [35] (f) StableLLVE [38] (g) Ours (h) Ground-truth

Figure 3: Visual comparison between our method and prior work on noisy-darkened video frames from DAVIS dataset.

The exemplary input dark frames in Figures 2 and 3 were created synthetically. Next, we examine our framework on a real-world video randomly selected from the LoLi-Phone dataset [23]. Note that there is no ground-truth video in this case. We present the enhanced frames corresponding to our work and related work in Figure 4. We have a similar observation to that of the synthetic dataset. Our method is capable of achieving an image with a natural lightness while keeping the coloring and visual content intact.

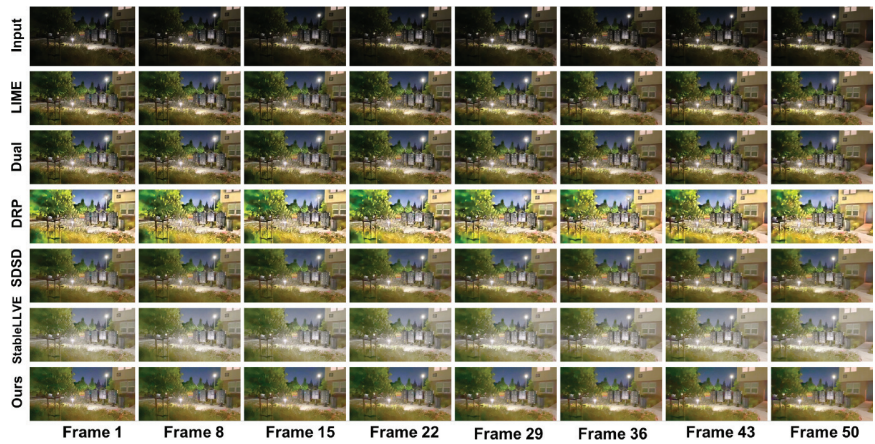


Figure 4: Visualization of video frames from a real-world dark video from LoLi-Phone dataset.

4.3 Quantitative Evaluation

We use four quantitative metrics to evaluate the performance of our method and compare it with prior work. First two metrics are Peak-Signal-to-Noise Ratio (PSNR) and Structural-SIMilarity (SSIM) [36], which we apply on all frames of the videos. We also use AB(Var) from [27] and Mean Absolute Brightness Difference (MABD) from [14] to assess temporal stability and its consistency with those of the ground truth. Table 1 and Table 2 show comparisons between our method and prior work for the clean and noisy datasets, respectively. In Table 1 and 2, we take the scores of all prior work except for Dual [39], DRP [24] and SDSD [35] from [38]. For DUAL (traditional method) and DRP (self-supervised method), we use their public codes to enhance the images. For SDSD, we fine-tune their pre-trained model on the DAVIS dataset. We then use the fine-tuned SDSD model to enhance the images. We then calculate the scores of these three methods and report them in Tables 1 and 2. Low scores of the DRP method is caused by its generative flavor, which makes

Table 1: Quantitative comparison for enhancing clean dark videos. The best scores are indicated in **bold**.

Method	PSNR↑	SSIM↑	AB(Var)↓	MABD↓
LIME [11]	17.36	0.7386	9.65	0.37
Dual [39]	18.12	0.8283	2.13	0.07
MBLLEN [27]	18.41	0.8100	77.24	1.95
RetinexNet [37]	19.78	0.8353	1.32	0.09
SID [7]	22.95	0.9428	4.93	0.43
DRP [24]	6.89	0.3160	6.73	0.52
NATLE [2]	26.70	0.9127	2.04	0.03
MBLLVEN [27]	24.50	0.9482	1.79	0.80
SMOID [14]	24.85	0.9472	1.30	0.17
SFR [9]	23.81	0.9413	2.14	0.11
BLIND [19]	22.87	0.9344	8.66	0.43
StableLLVE [38]	24.07	**0.9483**	1.96	0.05
SDSD [35]	25.09	0.8783	**0.98**	0.01
SALVE (Ours)	**28.85**	0.9225	1.47	**0.006**

Table 2: Quantitative comparison for enhancing noisy dark videos. The best scores are indicated in **bold**.

Method	PSNR↑	SSIM↑	AB(Var)↓	MABD↓
LIME [11]	16.43	0.4567	8.29	0.33
Dual [39]	18.38	0.6073	2.14	0.22
MBLLEN [27]	18.38	0.7982	78.76	1.93
RetinexNet [37]	19.56	0.7475	1.45	0.09
SID [7]	22.93	0.9253	4.03	0.39
DRP [24]	5.55	0.4107	15.37	0.34
NATLE [2]	25.55	0.8237	1.48	0.06
MBLLVEN [27]	23.08	0.8839	2.81	1.02
SMOID [14]	23.42	0.9212	**0.82**	0.17
SFR [9]	22.82	0.9299	2.29	0.12
BLIND [19]	22.94	0.9174	7.86	0.33
StableLLVE [38]	24.01	**0.9305**	3.00	0.10
SDSD [35]	22.27	0.8051	1.35	0.03
SALVE (Ours)	**27.06**	0.8202	1.18	**0.01**

the results different from the ground truth. Also, their public codes are only conducted for low-resolution video content. To tailor their implementations to high-resolution video content demands special efforts.

4.4 Computational Complexity

In this section, we calculate the runtime and FLOPs (FLoating Point Operations) of SALVE and prior work to offer a comparison on the computational complexity.

Table 3 shows the runtime comparison between different methods on CPU and GPU resources. We measure the average runtime of different methods for enhancing an RGB frame of size 530×942 on the CPU resource of *Intel Xeon 6130* and the GPU resource of *NVIDIA Tesla V100*. Since LIME and Dual only have CPU implementations, their GPU runtimes are not mentioned in the table. Table 3 shows that SALVE's runtime is better than LIME, Dual, NATLE and DRP. Specifically, SALVE is more than 2500 times faster than the self-supervised DRP method.

Table 4 compares the numbers of FLOPs per pixel among StableLLVE, SDSD, NATLE and SALVE. We use a FLOP counter tool[1] [33] for PyTorch to calculate FLOPs of StableLLVE and SDSD. We did not find a tool to measure the number of FLOPs of (non-PyTorch) LIME, Dual, and DRP methods. For NATLE and SALVE, we calculate their FLOP numbers manually [8].

Table 4 shows that SALVE has a significantly lower number of FLOPs than StableLLVE and SDSD. Our explanation for the lower runtime of these two methods in Table 3 is that their implementations in PyTorch are very efficient. In contrast, SALVE uses several libraries including SciPy in most of

Table 3: Average runtime (in seconds) comparison per RGB frame of size 530×942 pixels.

Method	CPU	GPU
LIME [11]	6.60	N/A
Dual [39]	13.20	N/A
DRP [24]	2760	2728
StableLLVE [38]	0.063	0.057
SDSD [35]	0.307	0.261
NATLE [2]	3.14	2.90
SALVE (Ours)	0.980	0.322

Table 4: FLOPs comparison per pixel.

Method	FLOPs	Ratio
StableLLVE [38]	51.19 K	7.20×
SDSD [35]	233.45 K	32.83×
NATLE [2]	10.85 K	1.52×
SALVE (Ours)	7.11 K	1×

[1] https://github.com/sovrasov/flops-counter.pytorch

the calculations. The latter is not as efficient as PyTorch. As a result, SALVE has a slightly longer runtime despite its lower number of FLOPs.

4.5 User Study

To further demonstrate the effectiveness of our method, we conduct a user study with 31 participants. In this study, we have 10 blind A/B tests between our method and prior works. At each time, only 2 videos are shown to the user. The 10 videos are randomly selected for this study. Each of the five prior work appears two times in the study. We show the results of this study in Figure 5. As shown in the figure, depending on the comparison baseline, between 87% to 100% of users prefer our method over the benchmarking method.

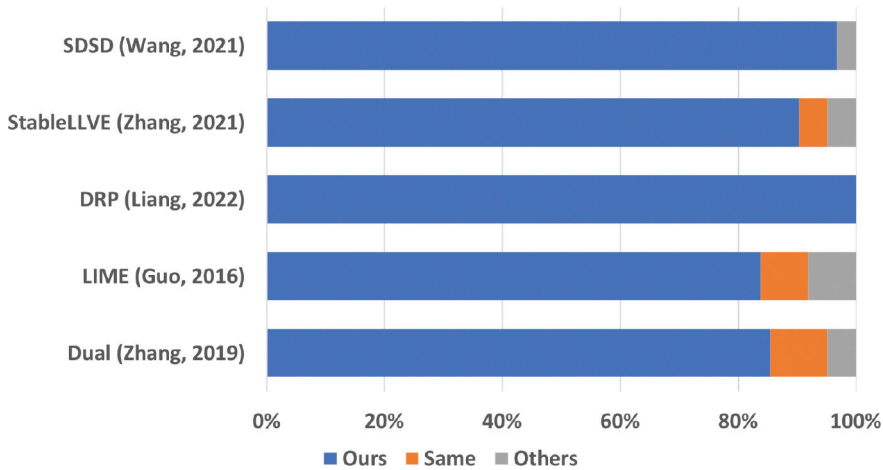


Figure 5: User study results, where we show user's preference in pair-wise comparison between our method and five benchmarking methods.

4.6 Ablation Study

An ablation study was conducted in [2] to show the effectiveness of α, β and denoising of $\widehat{R}$ on the final enhanced image. Here, we study the effect of these three parameters on the future enhanced frames in Figure 6. We see from the figure that cancelling parameter α and/or disabling the denoising operation results in noisy textures. Setting parameter $\beta = 0$ makes the edges of objects blurry and degrades texture preservance quality of the method.

As mentioned in Eq. (17), the window size of ridge regression is 5×5. Here, we analyze the effect of the window size on enhanced frames. Figure 7 shows this analysis. A small window size results in artifacts in the enhanced

(a) $\alpha = 0.0$, $\beta = 3$, denoising enabled. (b) $\alpha = 0.015$, $\beta = 0$, denoising enabled. (c) $\alpha = 0.015$, $\beta = 3$, denoising disabled. (d) $\alpha = 0.015$, $\beta = 3$, denoising enabled.

Figure 6: Effect of parameters α and β as well as the denoising operation on the enhanced frame's quality.

(a) 1×1 window (b) 3×3 window (c) 5×5 window (d) 7×7 window

Figure 7: Effect of the regressor's window size on the quality of enhanced frames.

frame (e.g., some pixels on the street light become red instead of maintaining the black color). Noisy patterns can be reduced as the window size increases. Overall, we strike a good balance between visual quality and the cost of ridge regression setting the window size to 5×5. A close look reveals noise patterns, blurry textures or artifacts in Figures 6 and 7.[2]

5 Conclusion

A new method for low-light video enhancement, called SALVE, was proposed in this work. The new self-supervised learning method is fully adaptive to the test video. SALVE enhances a few keyframes of the test video, learns a mapping from low-light to enhanced keyframes, and finally uses the mapping to enhance the rest of the frames. This approach enables SALVE to work without requiring (paired) training data. Furthermore, we conducted a user study and observed that participants preferred our enhanced videos in at least 87% of the tests. Finally, we performed an ablation study to demonstrate the contribution of each component of SALVE.

[2]High-resolution versions of Figures 6 and 7 are available on bit.ly/3TXGxRh.

Acknowledgment

This research was supported by a gift grant from Mediatek. The authors acknowledge the Center for Advanced Research Computing (CARC) at the University of Southern California for providing computing resources that have contributed to the research results reported within this publication. URL: https://carc.usc.edu.

References

[1] T. Arici, S. Dikbas, and Y. Altunbasak, "A histogram modification framework and its application for image contrast enhancement," *IEEE Transactions on image processing*, 18(9), 2009, 1921–35.

[2] Z. Azizi, X. Lei, and C.-C. J. Kuo, "Noise-aware texture-preserving low-light enhancement," in *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, IEEE, 2020, 443–6.

[3] Z. Azizi and C.-C. J. Kuo, "PAGER: Progressive Attribute-Guided Extendable Robust Image Generation," *arXiv preprint arXiv:2206.00162*, 2022.

[4] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis, and L. Van Gool, "The 2019 davis challenge on vos: Unsupervised multi-object segmentation," *arXiv preprint arXiv:1905.00737*, 2019.

[5] T. Celik and T. Tjahjadi, "Contextual and variational contrast enhancement," *IEEE Transactions on Image Processing*, 20(12), 2011, 3431–41.

[6] C. Chen, Q. Chen, M. N. Do, and V. Koltun, "Seeing motion in the dark," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 3185–94.

[7] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, 3291–300.

[8] S. Cools and W. Vanroose, "The communication-hiding pipelined BiCGstab method for the parallel solution of large unsymmetric linear systems," *Parallel Computing*, 65, 2017, 1–20.

[9] G. Eilertsen, R. K. Mantiuk, and J. Unger, "Single-frame regularization for temporally stable cnns," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 11176–85.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, 27, 2014.

[11] X. Guo, Y. Li, and H. Ling, "LIME: Low-light image enhancement via illumination map estimation," *IEEE Transactions on image processing*, 26(2), 2016, 982–93.

[12] H. Ibrahim and N. S. P. Kong, "Brightness preserving dynamic histogram equalization for image contrast enhancement," *IEEE Transactions on Consumer Electronics*, 53(4), 2007, 1752–8.

[13] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, 2462–70.

[14] H. Jiang and Y. Zheng, "Learning to see moving objects in the dark," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 7324–33.

[15] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, and Z. Wang, "Enlightengan: Deep light enhancement without paired supervision," *IEEE Transactions on Image Processing*, 30, 2021, 2340–9.

[16] D. J. Jobson, Z.-u. Rahman, and G. A. Woodell, "A multiscale retinex for bridging the gap between color images and the human observation of scenes," *IEEE Transactions on Image processing*, 6(7), 1997, 965–76.

[17] D. J. Jobson, Z.-u. Rahman, and G. A. Woodell, "Properties and performance of a center/surround retinex," *IEEE transactions on image processing*, 6(3), 1997, 451–62.

[18] C.-C. J. Kuo and A. M. Madni, "Green learning: Introduction, examples and outlook," *Journal of Visual Communication and Image Representation*, 90, 2023, 103685.

[19] W.-S. Lai, J.-B. Huang, O. Wang, E. Shechtman, E. Yumer, and M.-H. Yang, "Learning blind video temporal consistency," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, 170–85.

[20] E. H. Land, "The retinex theory of color vision," *Scientific american*, 237(6), 1977, 108–29.

[21] C.-H. Lee, J.-L. Shih, C.-C. Lien, and C.-C. Han, "Adaptive multiscale retinex for image contrast enhancement," in *2013 International Conference on Signal-Image Technology & Internet-Based Systems*, IEEE, 2013, 43–50.

[22] C. Lee, C. Lee, and C.-S. Kim, "Contrast enhancement based on layered difference representation of 2D histograms," *IEEE transactions on image processing*, 22(12), 2013, 5372–84.

[23] C. Li, C. Guo, L.-H. Han, J. Jiang, M.-M. Cheng, J. Gu, and C. C. Loy, "Low-light image and video enhancement using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[24] J. Liang, Y. Xu, Y. Quan, B. Shi, and H. Ji, "Self-Supervised Low-Light Image Enhancement Using Discrepant Untrained Network Priors," *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11), 2022, 7332–45.

[25] K. G. Lore, A. Akintayo, and S. Sarkar, "LLNet: A deep autoencoder approach to natural low-light image enhancement," *Pattern Recognition*, 61, 2017, 650–62.

[26] F. Lv, Y. Li, and F. Lu, "Attention guided low-light image enhancement with a large scale low-light simulation dataset," *International Journal of Computer Vision*, 129(7), 2021, 2175–93.

[27] F. Lv, F. Lu, J. Wu, and C. Lim, "MBLLEN: Low-Light Image/Video Enhancement Using CNNs.," in *BMVC*, Vol. 220, No. 1, 2018, 4.

[28] K. Nakai, Y. Hoshi, and A. Taguchi, "Color image contrast enhacement method based on differential intensity/saturation gray-levels histograms," in *2013 International Symposium on Intelligent Signal Processing and Communication Systems*, IEEE, 2013, 445–9.

[29] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017.

[30] X. Ren, W. Yang, W.-H. Cheng, and J. Liu, "LR3M: Robust low-light enhancement via low-rank regularized retinex model," *IEEE Transactions on Image Processing*, 29, 2020, 5862–76.

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, 234–41.

[32] M. Rouhsedaghat, M. Monajatipoor, Z. Azizi, and C.-C. J. Kuo, "Successive subspace learning: An overview," *arXiv preprint arXiv:2103.00121*, 2021.

[33] V. Sovrasov, *ptflops: a flops counting tool for neural networks in pytorch framework*, https://github.com/sovrasov/flops-counter.pytorch.

[34] J. Wang, W. Tan, X. Niu, and B. Yan, "RDGAN: Retinex decomposition based adversarial learning for low-light enhancement," in *2019 IEEE international conference on multimedia and expo (ICME)*, IEEE, 2019, 1186–91.

[35] R. Wang, X. Xu, C.-W. Fu, J. Lu, B. Yu, and J. Jia, "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*, 2021, 9700–9.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, 13(4), 2004, 600–12.

[37] C. Wei, W. Wang, W. Yang, and J. Liu, "Deep retinex decomposition for low-light enhancement," *arXiv preprint arXiv:1808.04560*, 2018.

[38] F. Zhang, Y. Li, S. You, and Y. Fu, "Learning temporal consistency for low light video enhancement from single images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 4967–76.

[39] Q. Zhang, Y. Nie, and W.-S. Zheng, "Dual illumination estimation for robust exposure correction," in *Computer Graphics Forum*, Vol. 38, No. 7, Wiley Online Library, 2019, 243–52.

[40] Y. Zhang, J. Zhang, and X. Guo, "Kindling the darkness: A practical low-light image enhancer," in *Proceedings of the 27th ACM international conference on multimedia*, 2019, 1632–40.