This is the Pre-Published Version.

The following publication L. -W. Wang, W. -C. Siu, Z. -S. Liu, C. -T. Li and D. P. -K. Lun, "Video Lightening with Dedicated CNN Architecture," 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 6447-6454 is available at https://dx.doi.org/10.1109/ICPR48806.2021.9413235.

Video Lightening with Dedicated CNN Architecture

Li-Wen Wang, Wan-Chi Siu, *Life-FIEEE,* Zhi-Song Liu, Chu-Tak Li and Daniel Pak-Kong Lun, *SrMIEEE*
The Hong Kong Polytechnic University, Hong Kong, China

Abstract— **Darkness brings us uncertainty, worry and low confidence. This is a problem not only applicable to us walking in a dark evening but also for drivers driving a car on the road with very dim or even without lighting condition. To address this problem, we propose a new CNN structure named as Video Lightening Network (VLN) that regards the low-light enhancement as a residual learning task, which is useful as reference to indirectly lightening the environment, or for vision-based application systems, such as driving assistant systems. The VLN consists of several Lightening Back-Projection (LBP) and Temporal Aggregation (TA) blocks. Each LBP block enhances the low-light frame by domain transfer learning that iteratively maps the frame between the low- and normal-light domains. A TA block handles the motion among neighboring frames by investigating the spatial and temporal relationships. Several TAs work in a multi-scale way, which compensates the motions at different levels. The proposed architecture has a consistent enhancement for different levels of illuminations, which significantly increases the visual quality even in the extremely dark environment. Extensive experimental results show that the proposed approach outperforms other methods under both objective and subjective metrics.**

Keywords—Low-light video enhancement, deep learning

I. Introduction

In order to resolve the problem of walking or driving in the dark environment, we may make use of visual aid (including small devices such as mobile phones), which is more easily or commonly available as compared with dedicated approaches such as infrared detection. Since each video composes of a sequence of images frames. Let us start with the problem of the image with bad lightening conditions. Images captured with insufficient illumination conditions usually have bad visual quality, such as low contrast, dim color, etc. Information among the low-light images faces substantial degradation that reduces its utility value. There could be possible solutions for taking photos in the low-light conditions, such as, using flash, increasing the sensitivity of the camera sensor (ISO) and taking photos with longer exposure time. However, these solutions have significant limitations: Flash may not be allowed in some public place, like cinema, museum, exhibition, etc. Higher camera sensitivity often brings noticeable noise in the dark regions. Longer exposure time is impractical for video capturing. Burst processing takes multiple low-light images under different exposures at a short time, and then combines them to obtain a large dynamic range. However, it cannot be generalized for enhancing low-light videos. Hence, it may be inevitable to obtain low-visibility images in low-light conditions. Enhancing such low-light images not only gives us a better visual quality but also gives benefit to vision-based systems, like autonomous driving, vision-based place recognition, etc.

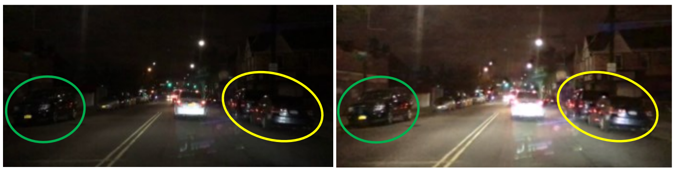

Figure 1. Effect of our proposed method
(the green and yellow circles show the significant improvement)

Researchers have proposed several methods to enhance the visible quality of the low-light images. Histogram Equalization (HE) [2, 7] method can rearrange the frequency of pixel intensity to obey uniform distribution, which significantly increases the dynamic range of the low-light images. However, changing directly the pixel values may cause color shift problems. Retinex-based methods [10] decompose the low-light image into two elements: reflectance and illumination map. The reflectance is the inherent attribute of the scene that is stable under different illumination conditions, while light information in the illumination map is different for low- and normal-light images. By adjusting the illumination map, the low- and normal-light images can be converted into each other. Other methods [11, 12] adopt dehazing theory which regards one image as a combination of scene information, ambient light, and refraction. Through enhancing the refraction map, it can predict the normal-light estimation for the low-light image.

Learning-based approaches have attracted enormous attention. Especially, the Convolutional Neural Networks (CNNs) have achieved impressive results in various vision-based tasks, e.g. classification [14], segmentation [1], super-resolution [15], etc. Benefited from the back-propagation theory and powerful computational ability of GPUs, the CNN-based methods have excellent learning ability. They can automatically learn and refine the feature representations of huge training datasets. The learning-based feature shows more discriminative power than those hand-craft features of the conventional approaches. Some CNN-based methods have been proposed for low-light image enhancement. Methods like Retinex-Net [9] and LightenNet [6] are based on the Retinex theory. They firstly use a CNN model to decompose the low-light image into illumination and reflectance. By using another CNN model to refine the illumination map, the low-light image can be enhanced for better visual quality. However, it is difficult to define the ground-truth maps for the decomposition process, which limits the performance of the enhancement. Other approaches, like EnlightenGAN [13], adopt the Generative Adversarial Network (GAN) [16] structure. Each GAN consists of a generator network and a discriminator network. The generator is responsible for predicting a fake normal-light image (estimation) for the input low-light image, while the discriminator needs to distinguish the fake normal-light images from a set of real normal-light images. During the training stage, the two networks will beat against each other, which constrains the visual quality of the predicted normal-

© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

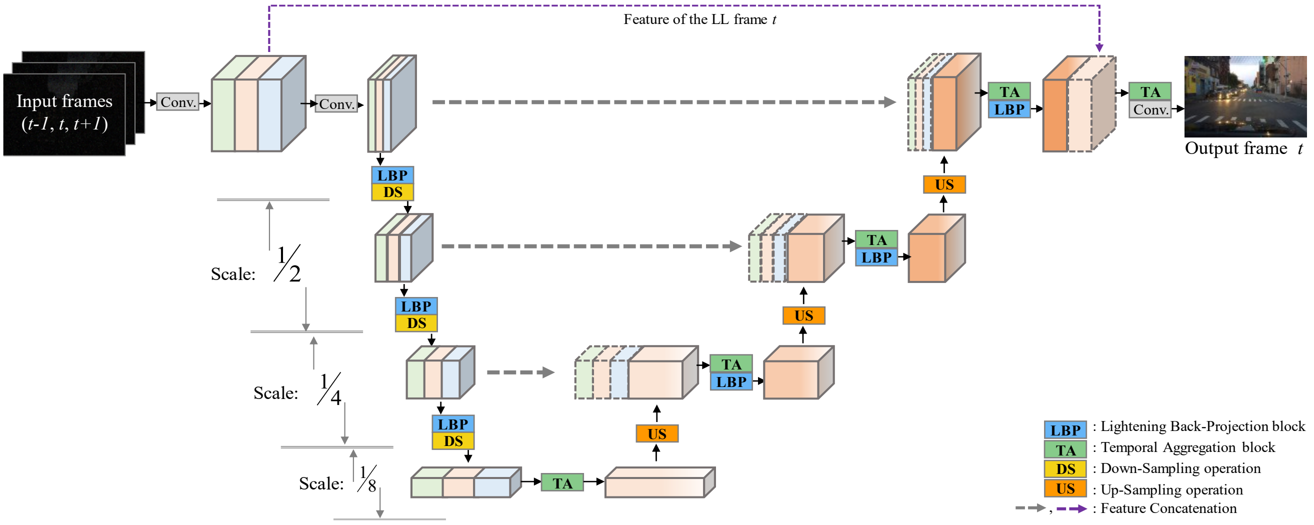
Feature of the LL frame t
Input frames
(t-1, t, t+1)
Conv.
Conv.
LBP
DS
TA
LBP
TA
Conv.
Output frame t
US
TA
LBP
Scale: 1/2
LBP
DS
US
TA
LBP
Scale: 1/4
LBP
DS
US
TA
Scale: 1/8
LBP : Lightening Back-Projection block
TA : Temporal Aggregation block
DS : Down-Sampling operation
US : Up-Sampling operation
: Feature Concatenation

Figure 2. Architecture of the proposed Video Lightening Network (VLN). The rectangles denote operations and the cubes denote the feature maps. LBP denotes the Lightening Back-Projection block (see Section II-C for details). TA represents the Temporal Aggregation block (see Section II-D for details)

light images. In other words, the generator will learn the mapping from low- to normal-light conditions. However, the two networks have completely different objectives, which makes the training process unstable. For low-light video enhancement, there are only few works on it. MBLLEN [17] hierarchically used 3-D convolution to extract features from the low-light frames and enhances them by an encoder-decoder structure. [18] investigates the temporal information of static scenes. It assumes that there is only a slight motion between video frames, which limits the generalization ability in real scenes. ViDeNN [19] proposes a realistic noise model for low-light video denoising, but lacks illumination enhancement. Although recent methods have achieved certain progress, the usage of temporal information is still in its fancy stage.

In this paper, we focus on video enhancement in low-light conditions. Based on the idea of enhancing the video frame iteratively and handling the motion in a multi-scale way, we propose a new CNN structure, i.e., the Video Lightening Network (VLN). It achieves remarkable enhancement for the low-light videos, as shown in Fig. 1. The novelty of the proposed method is listed below:

- **Video Lightening Network (VLN)**: The proposed method is based on our residual model to enhance a low-light frame with the support of neighboring video frames. It contains several lightening and temporal aggregation blocks that enhance the low-light frame accumulatively. Our VLN is compared with several state-of-the-art approaches with comprehensive experiments under both subjective and objective measures.

- **Back-Projection theory for video enhancement**: Based on the idea of enhancing video frames iteratively, we propose a Lightening Back-Projection (LBP) block that iteratively finds the mapping relation between low- and normal-light domains. It is the first work that successfully introduces back-projection theory for low-light video enhancement.

- **Temporal Aggregation**: We propose a Temporal Aggregation (TA) block that investigates the spatial and temporal relationships among adjacent frames. To handle multi-level motions, several TAs work in a multi-scale way which progressively compensates the motion.

The rest of the paper is organized as follows: Section II gives our model of the low-light video enhancement firstly, and then presents our proposed method (the VLN). Section III shows our experimental results, and Section IV concludes the paper.

II. Methodology

A. Overview

Low-light video enhancement is a fundamental video processing task that aims to reconstruct the normal-light (NL) videos from low-light (LL) inputs. Frames from LL videos usually have lower pixel values and more noise compared with those in the NL condition. For video processing, adjacent frames are strongly correlated. Using more video frames can bring more information that benefits noisy control and detail reconstruction. Instead of enhancing a LL video frame $\mathbf{X}_t$ individually, the VLN takes advantage of the information from $2N$ neighboring frames (i.e., from $\mathbf{X}_{t-N}$ to $\mathbf{X}_{t+N}$). We consider the relationship between the LL frame $\mathbf{X}_t$ and its corresponding NL frame $\mathbf{Y}_t$ as:

$$\mathbf{Y}_t = \mathbf{X}_t + E(\mathbf{X}_t) - n(\mathbf{X}_t) \tag{1}$$

where $E(\cdot)$ is an enhancing operator that estimates the lightening residual. $n(\cdot)$ is the additive noise of the LL frame. CNN is a powerful machine-learning tool that can be used to approximate the mapping function $F(\cdot)$ from the LL domain $\mathbf{X}_t$ to NL domain $\mathbf{Y}_t$. The optimization can be formulated as:

$$F = \mathrm{argmin}_F(\|\mathbf{Y}_t - F(\mathbf{X}_t)\|_2 + \Omega(F)) \tag{2}$$

where $\|\cdot\|_2$ represents the L2-norm distance between the estimation $F(\mathbf{X}_t)$ and the ground-truth NL frame $\mathbf{Y}_t$. $\Omega(\cdot)$ denotes the regularization term.

B. Video Ligthening Network (VLN)

Fig. 2 illustrates the architecture of the proposed Video Lightening Network (VLN). It takes three LL frames ($\mathbf{X}_{t-1}$, $\mathbf{X}_t$ and $\mathbf{X}_{t+1}$) as input, and predicts the NL estimation for the middle frame ($\hat{\mathbf{Y}}_t$). The VLN consists of six Lightening Back-Projection (LBP) blocks and five Temporal Aggregation (TA) blocks (we will present the details later).

As shown in Fig. 2, three LL frames firstly go through a convolutional process to obtain shallow feature representations (for the first Conv. on the left-hand side, each frame is processed by 36 filters, with filter size of 3×3×3, stride of 1, and the padding of 1). Subsequently, the channels are squeezed to select key features (the second Conv. on the left-hand side, each frame is processed by 12 filters, with the filter size of 3×3×36, the stride of 1, and the padding of 1). The frames then go through a U-shape architecture that consists of six LBP processes (see Section II-C for details) and four TA operations (see Section II-D for details). The left-hand side path acts in a contracting way, where there is a down-sampling process ("DS" in Fig. 2) after each LBP. We double the channels of the feature maps at each down-sampling operation through doubling the numbers of convolutional filters. The down-sampling operation is done via a convolution operation where each frame (e.g., the feature size is W×H×C, where W, H and C denote the size of width, height and channels separately) is processed by 2C filters, with filter size of 3×3×C, stride 2, and padding 1. The right-hand side path works in an expansive way that up-samples the feature map with several up-sampling ("US" in Fig. 2) processes. We reduce the number of the feature maps by half at each up-sampling process. The up-sampling process is done via a deconvolution layer, where the input feature map (e.g., the size is W×H×2C) is processed by C filters, with the filter size of 4×4×2C, stride 2, and padding 2. For the left- and right-side paths, the LBPs work in different ways. The LBPs at the left-hand-side path process each frame individually. In other words, there is no sharing of information among the adjacent frames on the left-side path. After the aggregation process of the TA blocks, information among different frames is fused and investigated. Then, the LBPs at the right-hand side path focus on the lightening process of the targeted middle frame. For low-light enhancement, both local and global information are useful, where global information can benefit the problem of illumination change, and local information can enhance details. Through these down/up-sampling processes, the LBPs (to be discussed in Section II-C) work in a multi-scale way that benefits the whole enhancing process. Besides, we add extra concatenations (the gray dotted arrows in Fig. 2) to migrate information from the former LBPs to the latter ones.

After the processing of the U-shape architecture, we can obtain the features for enhancing the LL frame $\mathbf{X}_t$. We regard the enhancement as a residual learning task and adds a short connection (purple dotted arrow in Fig. 2) to migrate the shallow features of $\mathbf{X}_t$. The setting of residual learning simplifies the goal of the U-shape network which makes it learn the residual between $\mathbf{Y}_t$ and $\mathbf{X}_t$, i.e., $E(\mathbf{X}_t) - n(\mathbf{X}_t)$ of Eqn. 1. Finally, the VLN combines (the last TA block on the right-hand side) the results and predicts (the last Conv. block on the right-hand side) the NL estimation for the middle frame, i.e., $\hat{\mathbf{Y}}_t$.

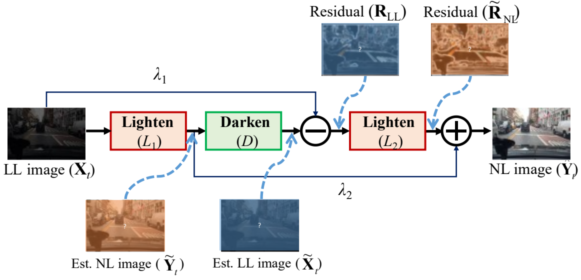
Residual ($\mathbf{R}_{\mathrm{LL}}$)
Residual ($\tilde{\mathbf{R}}_{\mathrm{NL}}$)
λ_1
Lighten (L_1)
Darken (D)
Lighten (L_2)
LL image ($\mathbf{X}_t$)
NL image ($\hat{\mathbf{Y}}_t$)
λ_2
Est. NL image ($\tilde{\mathbf{Y}}_t$)
Est. LL image ($\tilde{\mathbf{X}}_t$)

Figure.3. Lighten Back-Projection (LBP) Block

C. Lightening Back-Projection (LBP) Block

Our previous work on image (Deep Lightening Network (DLN) [3, 30]) has presented the advantages of LBP block in single low-light image enhancement. The LBP block is a basic processing module in our VLN network. Let us briefly explain the principle and structure of it. For low-light enhancement, the objective is to find the mapping function from LL domain to the NL domain. Similarly, it is obvious that there is an inverse mapping (darkening) from NL domain to the LL domain. Fig. 3 shows the structure of our LBP block which iteratively lightens and darkens the input LL frames. The procedure can be described as: a lightening operator L_1 firstly takes the LL frame $\mathbf{X}_t$ as input, then predicts its NL estimation $\tilde{\mathbf{Y}}_t$. Homogeneously, a darkening operator D can map the NL estimation $\tilde{\mathbf{Y}}$ back to the LL domain (i.e., $\tilde{\mathbf{X}}_t$). The estimated LL frame $\tilde{\mathbf{X}}_t$ should be close to the LL input $\mathbf{X}_t$. We measure the differences ($\mathbf{R}_{\mathrm{LL}}$) between them, which can be lightened (L_2) in the NL domain ($\tilde{\mathbf{R}}_{\mathrm{NL}}$). Based on the previous lightening result $\tilde{\mathbf{Y}}_t$, it can refine the NL estimation by $\hat{\mathbf{Y}}_t = \tilde{\mathbf{Y}}_t + \tilde{\mathbf{R}}_{\mathrm{NL}}$. Besides learning the direct mapping from the low-light domain to the normal-light domain ($\tilde{\mathbf{Y}}_t$), we use the proposed LBP block to lighten and darken the frame iteratively. It learns an additional residual term ($\tilde{\mathbf{R}}_{\mathrm{NL}}$) in the NL domain that can refine the NL estimation ($\hat{\mathbf{Y}}_t = \tilde{\mathbf{Y}}_t + \tilde{\mathbf{R}}_{\mathrm{NL}}$). The iterative structure divides the LL enhancement into lightening and refining operations, which decreases the burden of the lightening operation and increases the training efficiency of the whole architecture. The procedure can be formulated as the following equation:

$$\hat{\mathbf{Y}}_t = \lambda_2 \cdot L_1(\mathbf{X}_t) + L_2(D(L_1(\mathbf{X}_t)) - \lambda_1 \cdot \mathbf{X}_t) \qquad (3)$$

where λ_1 and λ_2 are two balanced coefficients that are acted by a 1-by-1 convolutional layer. Each of the lightening (L_1, L_2) and darkening (D) operations is implemented by a convolutional layer (the convolutional layer has C filters, where C is the number of channels of $\mathbf{X}_t$. Each filter has the size of 3×3×C, the stride of 1, and the padding of 1,).

The training process of LBP is guided by a global loss function at the end (see Section II-E), which constrains the output of the LBP in the NL domain. Considering the addition is a simple compensate operation, we can infer that its two inputs are in the same NL domain. Similarly, the input of LBP is in the LL domain, and the residual term after the subtraction should be in the same LL domain. Then, the convolutional layers are trained to convert the frame between the LL and NL domains (i.e., lightening or darkening).

D. Temporal Aggregation (TA) Block

Although the adjacent frames are strongly correlated, there are inevitable differences among them which is caused by moving objects, the motion of the camera, noise, etc. To utilize the information among the neighboring frames, we propose a novel TA block which aims to work for spatial-temporal feature aggregation through a multi-scale way.

To take the bottom TA block of Fig.2 as an example, it takes the result of the left-bottom LBP block as the input. As we mentioned before, the left-hand side path processes three

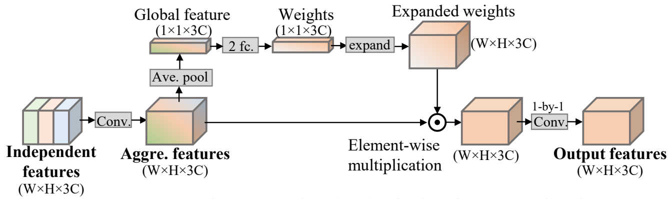

Figure.4. Temporal Aggregation (TA) Block. The rectangles denote operations and the cubes denote the feature maps.

frames ($\mathbf{X}_{t\text{-}1}$, $\mathbf{X}_{t}$ and $\mathbf{X}_{t+1}$) individually. For each frame, it obtains a feature map with the size of W×H×C. As shown in Fig. 4, the input features (independent features in Fig. 4) of the bottom TA block have the size of W×H×3C. It is firstly processed by a convolution layer that has 3C filters, with the filter size of 3×3×3C, the stride of 1, and the padding of 1. For the first two dimensions of the filters (3×3 at the spatial plane), the filters work on 3×3 regions to investigate the spatial relation of the feature map. Note that, due to the previous down-sampling process, each value at the feature map corresponds to the effect of a region of the three original frames, and the 3×3 filter can capture the temporal motion of these consecutive frames. The input (independent features at Fig. 4) consists features of 3C channels, where every "C channels" is from one input frame (frames *t-1*, *t* and *t+1*). Parameters of the third dimension (3C, the channel dimension) give the effect of the temporal information, which estimates the importance and correlations of the features of the three adjacent frames. Through the spatial and temporal fusion of the first convolutional layer, information of adjacent frames is fused that forms our aggregated features, as shown in Fig. 4.

The VLN aims to reconstruct NL estimations $\hat{\mathbf{Y}}_t$ from its LL observation $\mathbf{X}_t$ and two neighboring frames $\mathbf{X}_{t\text{-}1}$ and $\mathbf{X}_{t+1}$. The two neighboring frames provide additional information that benefits the reconstruction. However, most of the additional information is redundant, and even some information will degrade the result as noise and distorting motion. After the previous spatial-temporal aggregation, it is essential to extract and digest valuable information from the aggregated raw features. Different channels store the information from different feature descriptors. By investigating the channel-wise dependency, it can estimate the importance (weight) for different channels [31]. As shown in Fig. 4, the aggregated map (with the size of W×H×3C) is squeezed at the spatial plane through an average process (the "Ave. pool" in Fig. 4) to extract global representations (1×1×3C). Then, two fully-connected layers are used to estimate the weight of each channel (the "2 fc." in Fig. 4, where the first fully-connected layer has C/16 neurons and the second fully-connected layer has C neurons). To bring the weight into effect, it expands the weight (size: 1×1×3C) by repeating the values at the spatial plane (the size becomes W×H×3C) before multiplying to the aggregated features. Finally, a 1×1 convolutional layer (filter size is 1×1×3C, with the stride of 1, padding of 1) prepares features for the following process.

Multi-scale Motion Compensation: The TA block can handle the motion of a small 3×3 region of the feature map. However, the motions are at different levels, especially in the driving scene. There are four TA blocks in the U-shape structure, that work on different scales. The feature size of the left-bottom TA block is eight times (after three down-sample modules, $1/8=(1/2)^3$) smaller than the original frame size, while the size is sequentially doubled after the up-sampling processes in the right-side path. The multi-scale settings can benefit the capacity of handling motion at different levels.

E. Loss function

Given three LL video frames ($\mathbf{X}_{t\text{-}1}$, $\mathbf{X}_t$, and $\mathbf{X}_{t+1}$), the VLN predicts the NL estimation for the middle frame ($\hat{\mathbf{Y}}_t$). We regard the low-light video enhancement as a supervised learning task where the input LL frames ($\mathbf{X}_{t\text{-}1}$, $\mathbf{X}_t$, and $\mathbf{X}_{t+1}$) and NL target ($\mathbf{Y}_t$) are known during the training process. The loss function is defined to measure the difference between the estimation $\hat{\mathbf{Y}}_t$ and its ground truth target $\mathbf{Y}_t$. Considering that using the L1-norm loss may cause the blur problem, we include a refinement term, "$\lambda\cdot\|cp(\hat{\mathbf{Y}}_t)\text{-}cp(\mathbf{Y}_t)\|_2$", as shown in Eqn.4, which makes uses of the content perceptron loss $cp(\cdot)$. The whole loss function is shown in the following equation:

$$Loss(\hat{\mathbf{Y}}_t, \mathbf{Y}_t) = \|\hat{\mathbf{Y}}_t \text{-} \mathbf{Y}_t\|_1 + \lambda\cdot\|cp(\hat{\mathbf{Y}}_t) - cp(\mathbf{Y}_t)\|_2 \tag{4}$$

where $cp(\cdot)$ is defined as the feature maps from relu3_3 layer of the VGG-16 [20]. In other words, the estimation $\hat{\mathbf{Y}}_t$ and the ground truth $\mathbf{Y}_t$ are processed by the VGG-16 network. We then measure the L2-norm distance between their features from the layer relu3_3. The VGG-16 network was trained at ImageNet for image classification, where the extracted feature has discrimination power for recognizing different contents (objects). Some research works were done to verify its perceptual power through experiments [21]. λ is a balanced coefficient (it was set to 0.01 in our experiment).

III. Experimental Results

A. Implementation Details

Cameras cannot capture two videos simultaneously under different illuminations, which makes it impossible to obtain the LL-NL video pairs. However, a CNN model consists a large number of trainable parameters which require a huge dataset for the training process. Note that because videos captured with normal lighting (NL) condition contain nearly all scene information of the corresponding LL ones, we can simulate the LL videos from the NL ones by a reasonable synthesis method.

1) Simulation of Low-light Videos

Based on our low-light enhancement model (Eqn. 1), there are two main differences between LL($\mathbf{X}$) and NL($\mathbf{Y}$) videos: $E(\mathbf{X})$ and $n(\mathbf{X})$, where $E(\mathbf{X})$ is regarded as the content degradation and $n(\mathbf{X})$ is the additional noise. After comparing a set of LL and NL videos subjectively, we found that content degradation mainly due to saturation and contrast. The noise looks like Gaussian noise. Then, the LL video simulation pipeline can be designed as follows. The code of this LL-simulation pipeline will be released.

Saturation and contrast degradation: Insufficient illumination increases the difficulty of distinguishing different colors by the sensor. The low-contrast detail is easily degraded by the darkness. Therefore, videos in LL conditions always have dim color and weak contrast. We reduce the saturation and contrast to simulate such degradation. This method is similar to the controls on a color TV (by the Pillow [22] package). We degraded the saturation by interpolating between the input NL frame $\mathbf{Y}_t$ and its grayscale version $\mathbf{Y}_t^{(\text{grayscale})}$, using a control factor a (as shown in Eqn. 5). Similarly, the contrast is degraded by interpolating between the input $\mathbf{Y}_t^{(\text{-clr})}$ and a gray image $\mathbf{G}_t^{(\text{gray})}$ (as shown in Eqn. 6), where the gray

TABLE II. COMPARISON OF DIFFERENT METHODS ON DIFFERENT ILLUMINATIONS
(**RED**: BEST; **BLUE**: THE 2ND BEST, **GREEN**: THE 3RD BEST)

	Highway						*Cityscape*						Average	
	Slight dark		Middle dark		Extreme dark		Slight dark		Middle dark		Extreme dark			
Methods	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
AHE [2]	13.578	0.812	9.332	0.482	6.581	0.161	17.465	0.800	11.089	0.398	8.184	0.111	11.038	0.461
BIMEF [4]	17.414	**0.920**	13.119	0.722	8.732	0.343	**21.869**	**0.964**	14.784	0.737	9.911	0.224	14.305	0.652
LIME [5]	**21.438**	**0.921**	17.102	**0.861**	12.599	0.628	18.391	0.862	**19.611**	**0.815**	**14.343**	0.446	**17.247**	**0.756**
LightenNet [6]	16.154	0.867	13.742	0.823	13.868	**0.759**	15.966	0.814	17.160	0.760	13.024	0.383	14.986	0.734
LLNet [8]	**20.451**	0.896	**21.932**	**0.860**	**15.409**	0.697	**19.380**	**0.869**	17.967	0.723	13.221	**0.505**	**18.060**	**0.758**
Retinex-Net [9]	15.147	0.810	17.237	0.851	**14.911**	**0.726**	12.996	0.652	18.387	0.788	**14.482**	**0.468**	15.527	0.716
EnlightenGAN [13]	17.888	0.847	**19.493**	0.850	12.205	0.599	17.840	0.814	**20.030**	**0.855**	12.083	0.417	16.590	0.730
VLN(proposed)	**29.750**	**0.974**	**29.822**	**0.964**	**27.120**	**0.917**	**23.759**	**0.922**	**23.816**	**0.907**	**25.759**	**0.863**	**26.671**	**0.924**
Average	18.977	0.881	17.722	0.802	13.928	0.604	18.458	0.837	17.855	0.748	13.876	0.427	16.803	0.716

value is determined by the mean of the original image's grayscale version.

$$\mathbf{Y}_t^{(\text{-clr})} = a\mathbf{Y}_t + (1\text{-}a)\mathbf{Y}_t^{(\text{grayscale})} \quad (5)$$

$$\mathbf{Y}_t^{*} = b\mathbf{Y}_t^{(\text{-clr})} + (1\text{-}b)\mathbf{G}_t^{(\text{gray})} \quad (6)$$

where $\mathbf{Y}_t^{(\text{-clr})}$ denotes the frame after color degradation. $\mathbf{Y}_t^{*}$ represents the frame after both color and contrast degradation. a and b are two control factors that are randomly selected from 0.3 to 1.

Brightness degradation: Another significant difference between the LL and NL videos is the brightness. A LL video usually have lower pixel values and narrow dynamic ranges, which can be simulated by the linearized Gamma transformation [23]. After the saturation and contrast degradation, we decrease the brightness of the previous result $\mathbf{Y}_t^{*}$ that can be formulated as:

$$\mathbf{X}_t^{*(i)} = \beta\times(\alpha\times\mathbf{Y}_t^{*(i)})^{\gamma} \quad (7)$$

where $\mathbf{X}_t^{*}$ and $\mathbf{Y}_t^{*}$ denote the t^{th} degraded frame and the result after the saturation and contrast degradation. The pixel value is compressed to [0, 1]. $i \in\{R,G,B\}$ is the RGB channels of the image. $\alpha \sim U\,(0.9, 1)$, $\beta \sim U\,(0.5, 1)$ and $\gamma \sim U\,(1.5, 5)$ are the factors that control the brightness of the simulated frames (all settings are the same as [23]).

Additive Gaussian Noise: Gaussian Noise is a widely-used noise model that shows great generalization ability in image and video denoising field. It needs to add the noise to degrade videos through $\overline{\mathbf{X}}_t = \mathbf{X}_t^{*} + \mathbf{n}$, where $\overline{\mathbf{X}}_t$ denotes the simulated LL video frame. $\mathbf{n}\sim N(0, \delta)$, and δ controls the noise level (we randomly selected δ from 0 to 0.001 in our experimental work).

2) Experiment Settings

Berkeley Deep Drive (BDD) [24] dataset is a widely used dataset that contains 100,000 HD video sequences of driving experience. We selected seven video sequences (five sequences for training and two sequences for testing) in ideal NL conditions, where each sequence lasts for 40 seconds (about 1,200 frames, 30 fps). To build a LL-NL frame pair, we randomly selected three consecutive NL frames, and simulated the corresponding LL frames through the same LL-simulation procedure as the above. Let us refer to the architecture of our VLN network. We randomly initialized the weights with normal distribution and biases equal to zero as in [25]. We adopted the Adam [26] optimization method with momentum of 0.9, weight decay of 0.0001. The learning rate was set as 0.0001. We randomly cropped 256*256 patches from the LL and NL frames as the training pair. For each iteration, the mini-batch size was set to 20, and the model was trained for 500 epochs. All experiments were conducted through a PC with two NVIDIA GTX2080Ti GPUs.

TABLE I. SIMULATION PARAMETERS OF DIFFERENT ILLUMINATION LEVELS

	Slight dark	Middle dark	Extreme dark
β	0.9	0.8	0.6
α	0.98	0.95	0.93
γ	2	3	4
Satur.	0.8	0.6	0.4
Contrast	0.8	0.6	0.4

There is no objective evaluation method in the field of low-light video enhancement, which makes it difficult to compare with other approaches. We believe that the predicted NL estimations should be close to the ground-truth NL frames. So we can evaluate it by measuring the difference between estimation and ground truth ones. Peak Signal-to-Noise Ratio (PSNR) and Structure SIMimarity (SSIM) are two widely used evaluation measures in the image restoration field [15, 27, 28]. We adopt them as the objective measurement. Subjectively, we will also make a visualization comparison for videos originally taken in a dark environment. We will compare the proposed method with many approaches, including conventional approaches (Adaptive Histogram Equalization [2] (AHE), BIMEF [4], LIME [5]) and CNN-based approaches (LightenNet [6], LLNet [8], Retinex-Net [9], EnlightenGAN [13]). In order to make the comparison as fair as possible, we have made use of their desired settings for the possible best performance.

B. Evaluation on the Synthesis Dataset

Different scenes may under different illuminations, which could produce inconsistent brightness of video sequences. Therefore, the LL video enhancement method should be robust

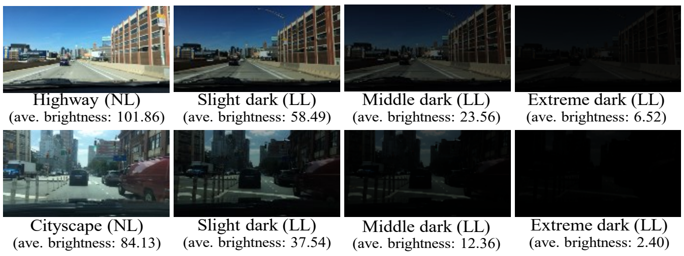
Highway (NL) (ave. brightness: 101.86)
Slight dark (LL) (ave. brightness: 58.49)
Middle dark (LL) (ave. brightness: 23.56)
Extreme dark (LL) (ave. brightness: 6.52)
Cityscape (NL) (ave. brightness: 84.13)
Slight dark (LL) (ave. brightness: 37.54)
Middle dark (LL) (ave. brightness: 12.36)
Extreme dark (LL) (ave. brightness: 2.40)

Figure 5. Simulated LL Videos under different illuminations (frame: 10), where the maximum brightness is 255.

to the illumination changes. To evaluate the robustness, two testing NL videos (one was captured at the highway, and the other was the cityscape) were synthesized for different illuminations: slight, middle and extreme dark. The settings are listed in TABLE I. Fig. 5 shows some examples of simulated testing videos. The video frames of highway (the average brightness is 101.86) are brighter than the cityscape's (the average brightness is 84.13), as there are more buildings in the street. By using the three settings, we obtained the LL videos in different illuminations. In the slight-dark case, the brightness is reduced by a half. The frames become slightly dark but the contents are still visible. In the middle-dark condition, the frames are darker than the slight ones, where only the salient contents are visible. We make use of the extreme dark condition to simulate the night, where nearly all contents are invisible and the frames are almost black (the average brightness is only 6.52 for *highway* and 2.40 for *cityscape*). We then tested different methods with the testing dataset.

TABLE II shows the performance of different methods. For the evaluation indices (i.e. PSNR and SSIM), a larger value means the estimation is closer to the ground truth. Video frames with darker scene usually faces more substantial content degradation. The performance in extreme-dark cases is worse than the slight-dark ones (e.g., the average SSIM is 0.604 at the extreme-dark *highway*, but it is 0.881 at the slight-dark *highway*). It can be seen from the table that AHE and BIMEF are more sensitive to the illumination changes. For example, the BIMEF achieves 0.964 SSIM in slight-dark cityscape, but the performance reduces significantly that it obtains 0.224 SSIM in the extreme-dark case. Learning-based approaches (LightenNet and Retinex-Net) were usually trained with huge datasets that contain scenes under different illuminations. It significantly improves the robustness of the methods. It can be seen from the table that the learning-based methods give a better estimation for the extreme cases compared with the conventional approaches (LightenNet achieves 0.759 SSIM at the extreme-dark *highway*). Our method is more robust and obtains consistent results in different illumination conditions.

Ablation Study: To evaluate the effectiveness of the proposed structure, we made a comparison among different structures on a small validation dataset. It contains five short sequences (each contains ten frames), and we used the same simulation methods to simulate the NL frames to middle-dark illuminations. It can be seen from Table III that using image-based DLN [3], which was our previous work, achieves 17.3 dB PSNR at the validation dataset. The Plain-Net consists of ten stacked convolutional layers that achieve less PSNR (12.23 dB) with more trainable parameters (Plain-Net: 1.04M, DLN: 0.70M). The Residual [29] structure solves the gradient vanishing problem that only increases training efficiency. We composed a shallow "Residual-Net" that consists of seven residual blocks. It achieves similar performance as the PlainNet with slightly fewer parameters (0.89M). The U-Net [1] is initially designed for medical image segmentation that contains about 13.30M parameters. It achieves better performance (0.81 SSIM) compared with PlainNet and Residual-Net. The DLN achieves similar SSIM (0.79) and required the fewest parameters (0.70M) among all approaches. The VLN which contains six LBP blocks achieves the highest (20.98 dB) PSNR with 4.96M parameters. The U-Net achieves 19.22 dB PSNR but requires a large number of parameters (13.3M) in the image-based experiments. It suggests the effectiveness of the proposed LBP structure. Let us also compare the U-Net with our proposed VLN for video-based enhancement. It is clear that adjacent frames benefit the LL enhancement, where the video-based U-Net increases the PSNR with 2.48dB (=21.70-19.22). Our proposed VLN network contains four Temporal Aggregation (TA) blocks working at the features of different scales, which can investigate the temporal relationship among the adjacent frames. Compared with the image-based version, the video-based VLN improves the PSNR (from 20.98 dB to 26.40 dB) and SSIM (from 0.82 to 0.95) to a large extent.

TABLE III. COMPARISON OF DIFFERENT STRUCTURES

Method	Input Structure*	PSNR (dB)	SSIM	Para. (M)
PlainNet	SImage	12.23	0.57	1.04
Residual-Net	SImage	12.03	0.54	0.89
U-Net [1]	SImage	19.22	0.81	13.30
DLN [3]	SImage	17.30	0.78	0.70
VLN	SImage	20.98	0.82	4.96
U-Net [1]	Video	21.70	0.93	13.30
VLN (proposed)	**Video**	**26.40**	**0.95**	**4.96**

* the type "SImage" denotes the method using single-image enhancement (the input is a single LL image), "video" means the enhancement is based on a set of adjacent frames (all using three LL frames).

Fig. 6 gives a visual comparison of different methods. It is clear that the visual quality of the LL frames can be improved after processing the LL enhancement. For example, the sign of the slight-dark *cityscape* (picture at the first row, sixth column) is difficult to identify originally. After the enhancement, the contrast is increased that makes it easy to recognize the objects (see the remaining pictures of the sixth column). However, when the darkness level is increased (e.g., extreme-dark cases), the AHE and BIMEF give the estimations with narrow dynamic range (see the cars of the extreme-dark *highway*). Retinex-Net tends to predict brighter results but the estimations look like an oil painting. LIME and EnlightenGAN give better predictions for the slight- and middle-dark scenes, but the estimations for extreme-dark cases are with bad visual quality (see the signs of the extreme-dark *cityscape*). The proposed DLN has a consistent enhancement for different levels of illuminations (see the cars and signs of different illuminations), which suggests the robustness of the method.

C. Evaluation on the Real Datasets

An application of the low-light enhancement is able to assist drivers during the nighttime. To evaluate the generalization ability of the proposed method, we apply it for the real scenes. Our testing videos are from BDD dataset [24] that was captured by the driving recorder at night. Fig. 7 shows the comparisons among different methods. It can be seen from Fig. 7 (a) that the LLNet produces many stains that decrease the visual quality. Retinex-Net [9] distorts the color information whose result looks like an artwork. The outputs of LIME and EnlightenGAN have satisfactory saturation and contrast. However, the enhancements amplify the noise which makes the frame mess (see the red rectangle area of Fig. 7 (a)). As we mentioned before, the proposed method is robust to different illuminations. Based on the spatial and temporal information of adjacent neighboring frames, the noise is suppressed that the visual quality of the reconstructed NL frames are improved. Our method produces a clearer result (see the proposed method in Fig. 7 (a)). It contains less noise and

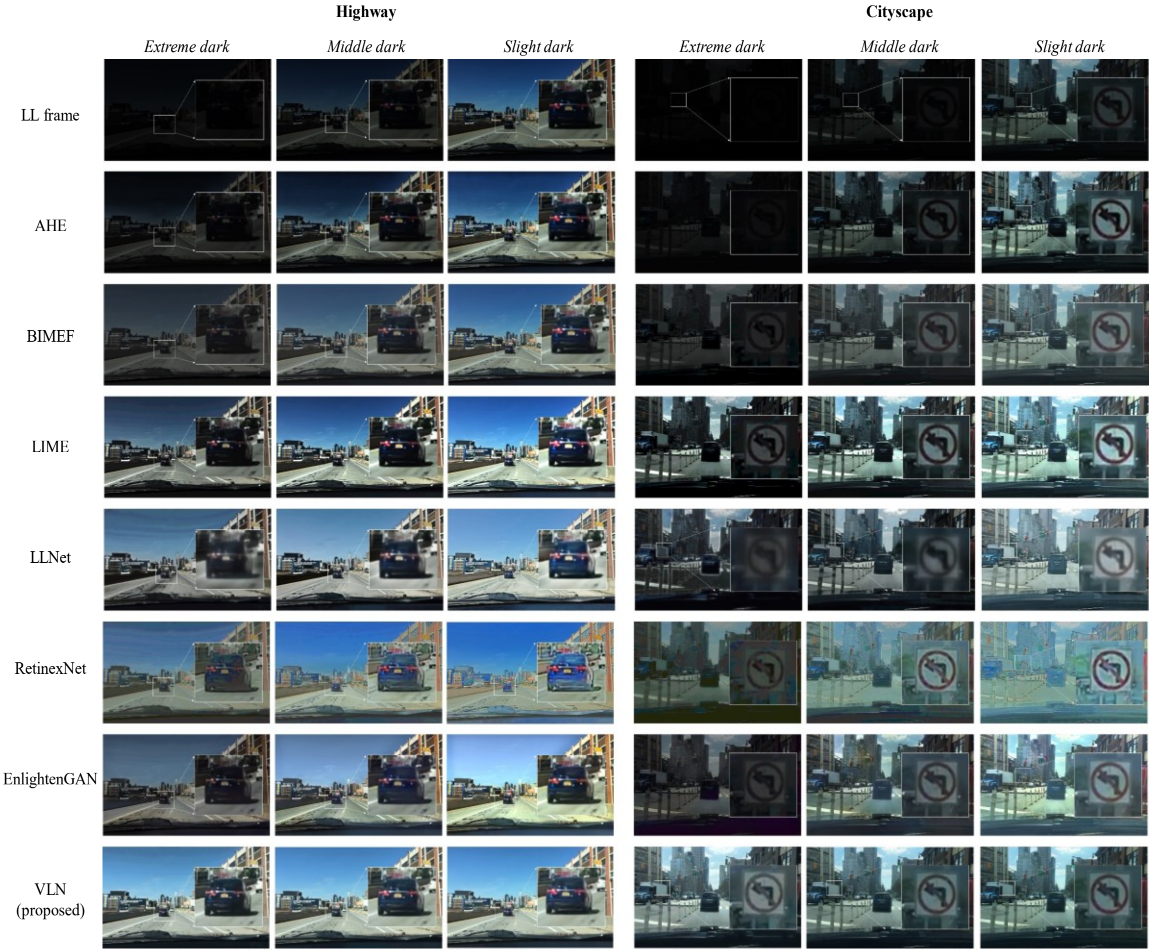

Figure 6. Visual comparison of different methods for different illuminations. The first row is the LL pitures under different illumnations. The first three pitures are the 10th frame of the tesing video *highway* that are under extre-, slight-, and slight darknesses seperately. The following three pitures are the 10th frame of the tesing video *cityscape*. Different rows contain the estimations of different methods. Zoom in for a better view.

reasonable brightness, which provides a better view for driving. The video in Fig. 7 (b) is captured in a narrow alley. The scene is extremely dark and it is challenging to recognize cars on both sides. All LL enhancement methods can improve the brightness that makes the cars more visible. However, the result of the LLNet contains black-and-white stains. LIME [5] and EnlightenGAN [13] produce visible noise (the green rectangle area of Fig. 7 (b)). Our proposed method gives a clear result with good visual quality. Based on the temporal information, our method achieves better temporal consistency that is more stable than others.

IV. Conclusion

In this paper, we have introduced our proposed Video Lightening Network (VLN) for low-light video enhancement. We have proposed our Lightening Back-Projection (LBP) as a basic enhancing module that iteratively learns the mappings between low- and normal-light domain. To utilize the temporal information among adjacent frames, we have also proposed a novel Temporal Aggregation (TA) block, which investigates the spatial and temporal relations of a small region. Based on the hierarchically multi-scale features, the TAs can handle the motion of different levels. We have used both objective and subjective metrics to compare the proposed method with others. Extensive experimental results show that the proposed method outperforms others (both conventional and learning-based) in quantitative and qualitative aspects. The proposed method provides a novel solution to resolve the problem of dark environment, which can be used in a wide range of applications. A possible application is to design an intelligent front windshield such that it can automatically be activated when driving in the dark environment.

Ack.: This work is supported from the HKPolyU Consultancy Project P19-0319 for HKPC.

References

[1] O Ronneberger, P Fischer *et al.*, "U-net: Convolutional networks for biomedical image segmentation," *Proceedings, Int. Conf. on medical image computing and computer-assisted intervention*, pp. 234-241, 2015, Munich, Germany.

[2] M Abdullah-Al-Wadud, MH Kabir *et al.*, "A dynamic histogram equalization for image contrast enhancement," *IEEE trans. on consumer electronics,* vol. 53, no. 2, pp. 593-600, 2007.

[3] Li-Wen Wang, Zhi-Song Liu, Wan-Chi Siu and Daniel Pak-Kong Lun, "Deep Lightening Network for Low-light Image Enhancement," *Proceedings, The IEEE ISCAS, Paper Accepted*, 2020, Seville Spain.

[4] Z Ying, G Li *et al.*, "A bio-inspired multi-exposure fusion framework for low-light image enhancement," *arXiv preprint arXiv:1711.00591*, 2017.

[5] X Guo, Y Li *et al.*, "LIME: Low-light image enhancement via illumination map estimation," *IEEE transactions on image processing (TIP),* vol. 26, no. 2, pp. 982-993, 2016.

[6] C Li, J Guo *et al.*, "Lightennet: A convolutional neural network for weakly illuminated image enhancement," *Pattern recognition letters,* vol. 104, pp. 15-22, 2018.

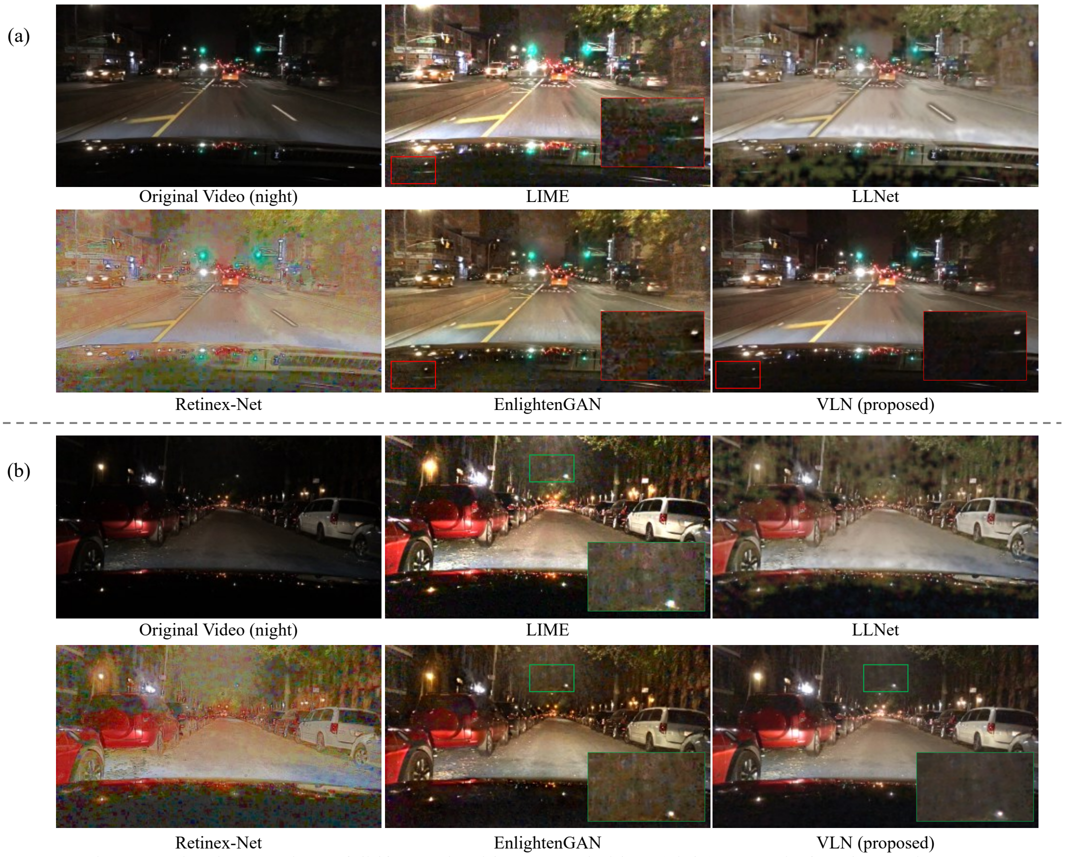

Figure 7. Visual comparison of didferent algorithms on real videos (night). Zoom in for a better view.

[7] ED Pisano, S Zong *et al.*, “Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms,” *Journal of digital imaging (JDI),* vol. 11, no. 4, pp. 193, 1998.

[8] KG Lore, A Akintayo *et al.*, “LLNet: A deep autoencoder approach to natural low-light image enhancement,” *Pattern Recognition,* vol. 61, pp. 650-662, 2017.

[9] C Wei, W Wang *et al.*, “Deep retinex decomposition for low-light enhancement,” *Proceedings, BMVC*, 2018, Newcastle, UK.

[10] Z-u Rahman, DJ Jobson *et al.*, “Retinex processing for automatic image enhancement,” *Journal of electronic imaging,* vol. 13, no. 1, pp. 100-111, 2004.

[11] J-H Kim, J-Y Sim *et al.*, “Single image dehazing based on contrast enhancement,” *Proceedings, IEEE ICASSP*, pp. 1273-1276, 2011, Prague, Czech Republic.

[12] L Li, R Wang *et al.*, “A low-light image enhancement method for both denoising and contrast enlarging,” *Proceedings, the IEEE ICIP*, pp. 3730-3734, 2015, Québec, Canada.

[13] Y Jiang, X Gong *et al.*, “EnlightenGAN: Deep Light Enhancement without Paired Supervision,” *arXiv preprint arXiv:1906.06972*, 2019.

[14] A Krizhevsky, I Sutskever *et al.*, “Imagenet classification with deep convolutional neural networks,” *Proceedings, Advances in neural information processing systems*, pp. 1097-1105, 2012.

[15] Zhi-Song Liu, Li-Wen Wang, Chu-Tak Li and Wan-Chi Siu, “Hierarchical Back Projection Network for Image Super-Resolution,” *Proc., the IEEE CVPR Workshop*, 2019, California.

[16] I Goodfellow, J Pouget-Abadie *et al.*, “Generative adversarial nets,” *Proceedings, Advances in neural information processing systems*, pp. 2672-2680, 2014.

[17] F Lv, F Lu *et al.*, “MBLLEN: Low-Light Image/Video Enhancement Using CNNs,” *Proc., BMVC*, pp. 220-233, 2018.

[18] C Chen, Q Chen *et al.*, “Seeing motion in the dark,” *Proceedings, the IEEE International Conference on Computer Vision*, pp. 3185-3194, 2019, South Korea.

[19] M Claus and J van Gemert, “ViDeNN: Deep Blind Video Denoising,” *Proc., the IEEE CVPR Workshop*, 2019, California.

[20] K Simonyan and A Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[21] R Zhang, P Isola *et al.*, “The unreasonable effectiveness of deep features as a perceptual metric,” *Proc, the IEEE CVPR*, pp. 586-595, 2018, Utah, United States.

[22] ACa Contributors. "Pillow: a python imaging library," Access date: 24 Sep, 2019; Retrieved from https://python-pillow.org.

[23] F Lv and F Lu, “Attention-guided Low-light Image Enhancement,” *arXiv preprint arXiv:1908.00682*, 2019.

[24] F Yu, W Xian *et al.*, “Bdd100k: A diverse driving video database with scalable annotation tooling,” *arXiv preprint arXiv:1805.04687*, 2018.

[25] K He, X Zhang *et al.*, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *Proc., the IEEE ICCV*, pp. 1026-1034, 2015, Las Condes, Chile.

[26] DP Kingma and J Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[27] Jun-Jie Huang, Wan-Chi Siu and Tian-Rui Liu, “Fast image interpolation via random forests,” *IEEE Trans. on Image Processing,* vol. 24, no. 10, pp. 3232-3245, 2015.

[28] Jun-Jie Huang and Wan-Chi Siu, “Learning hierarchical decision trees for single-image super-resolution,” *IEEE Trans. on CSVT,* vol. 27, no. 5, pp. 937-950, 2017.

[29] K He, X Zhang *et al.*, “Deep residual learning for image recognition,” *Proc., IEEE CVPR*, pp. 770-778, 2016, Las Vegas.

[30] L. Wang, Z. Liu, W. Siu and D. P. K. Lun, "Lightening Network for Low-Light Image Enhancement," *IEEE transactions on image processing (TIP)*, vol. 29, pp. 7984-7996, 2020.

[31] Hu, Jie, Li Shen, and Gang Sun. "Squeeze-and-excitation networks." *Proc., IEEE CVPR*,, pp. 7132-7141. 2018, Utah.