Report of The First Stage

1. Background:

In real world, images are often captured under sub-optimal lighting conditions due to unavoidable environmental factors and technological limitations, resulting in low-light and overexposure images with poor visual quality. For a vehicle platform, such low-quality images are not only challenging to handle but may also pose risks. To address the aforementioned issue and enhance those images, low-light enhancement and overexposure correction have emerged, aiming to respectively transform low-light images and overexposed images into normal-light images.

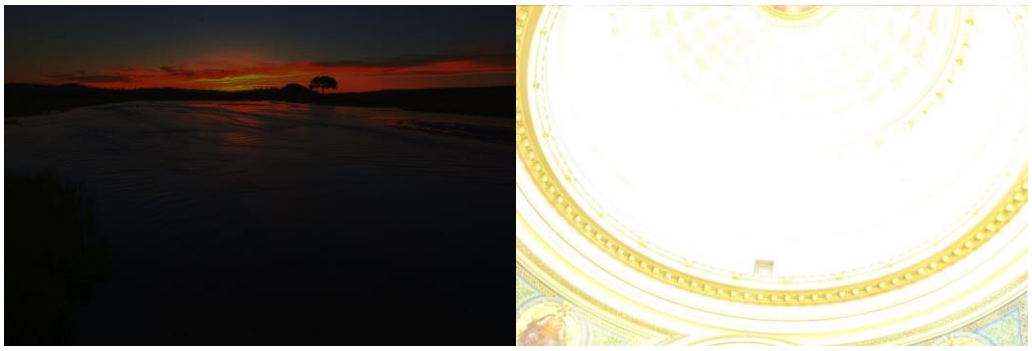

Figure 1 Low-light image and overexposure image.

Low-light enhancement refers to improving the visibility and contrast of images under low-light conditions while minimizing noise and color distortion as much as possible. Initially, low-light enhancement was primarily based on analog methods and simple digital techniques like histogram equalization and gamma correction. These methods aimed to improve the visibility of images captured in low-light conditions by adjusting brightness and contrast. In the late 20th century, the Retinex theory became influential. It led to methods that separated the illumination and reflectance components of an image, aiming to enhance detail while preserving natural colors. With the rise of digital image processing, more sophisticated techniques were developed. These included adaptive histogram equalization and local tone mapping, which offered better results by considering local image content. The most significant leap came with the introduction of deep learning. Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) have been applied to learn the mapping from low-light to well-lit images, resulting in remarkable improvements in image quality. Today, state-of-the-art methods in low-light enhancement leverage deep neural networks to address various challenges, such as noise reduction, detail preservation, and color fidelity.

Overexposure correction involves addressing the issue of areas in an image being too bright and color-faded due to overly long exposure times. In the early days of photography, overexposure was addressed during the development process in darkrooms. Photographers used techniques like dodging and burning to correct exposure on specific areas of the print. With the advent of digital cameras and image editing software, overexposure correction became more accessible. Tools like Photoshop introduced features to adjust brightness, contrast, and recover details from overexposed

areas. High Dynamic Range (HDR) imaging and tone mapping techniques emerged, allowing for the combination of multiple exposures to create a single image with a balanced exposure throughout. Recently, deep learning has revolutionized overexposure correction. Neural networks can now learn to recover details from overexposed regions by analyzing large datasets of images. These methods aim to generate alternative contents for saturated regions while maintaining global contrast and naturalness.

Despite the dominance of deep learning methods represented by CNNs due to their superior effectiveness, generality, and performance, they struggle to achieve high efficiency, which has become a significant barrier to their widespread application on vehicle platforms. On the contrary, Look-Up Tables (LUTs), widely utilized in Image Signal Processing (ISP) system and image editing software as an effective pixel adjustment tool, inherently possess high efficiency since LUTs employs lookup to replace complex calculations.

In low-light image processing, where images often suffer from color distortion and blurred details due to dim lighting conditions, applying a 3D LUT allows us to correct color shifts in the image, resulting in more accurate and natural colors. Additionally, a 3D LUT can adjust the contrast and brightness of the image, making it clearer and more visible under low-light conditions. Through precise mapping and adjustment of pixel values, a 3D LUT effectively enhances the image's details and color expression, thereby improving its quality and visual appeal.

Similarly, in overexposed image processing, where images may lose detail in certain areas or exhibit excessive color saturation due to strong light or improper exposure, a 3D LUT can be utilized to adjust the image's color and brightness, resulting in a more balanced and natural exposure across the image. By accurately mapping and adjusting the pixel values of the image, a 3D LUT can effectively correct color shifts and brightness imbalances in overexposed images, enhancing their visual appeal and overall quality.

However, traditional LUTs are entirely designed and calibrated manually, which results in higher costs and significantly less flexibility compared to deep learning methods. As a result, the combination of LUTs and deep learning is proposed to automatically learn more powerful LUTs via data-driven deep learning approaches and simultaneously achieve high efficiency. Following this idea, we propose an efficient LUT-based low-light video enhancement method with high performance in our paper titled "FastLLVE: Real-Time Low-Light Video Enhancement with Intensity-Aware Lookup Table", further demonstrating the feasibility of AI-based LUTs.

Compared to pure deep learning methods or LUTs, AI-based LUTs that balance performance and efficiency perfectly meet the requirements of vehicle platforms, becoming a new direction for developing image enhancement methods for vehicles. Therefore, this project, Integrated Image Enhancement Method for Vehicles Based on Lookup Tables, came into being, aiming at developing AI-based LUT algorithm based on our research to compensate for the shortcomings of hardware ISP in processing images on vehicle platforms.

2. Algorithm:

According to the requirements of integrated image enhancement on vehicle, we adjuste the proposed FastLLVE as shown in Figure 2 to an image version, further simplifying the model to improve efficiency while ensuring performance. Specifically, we adopt the classical 3D LUT instead of LUT with more complex structure, add down-sampling operations, and remove unnecessary

optional modules. The adjusted image enhancement model is shown in Figure 2. In terms of operation sequence, the entire model can be divided into three stages: feature extraction, LUT generation, and LUT transformation.

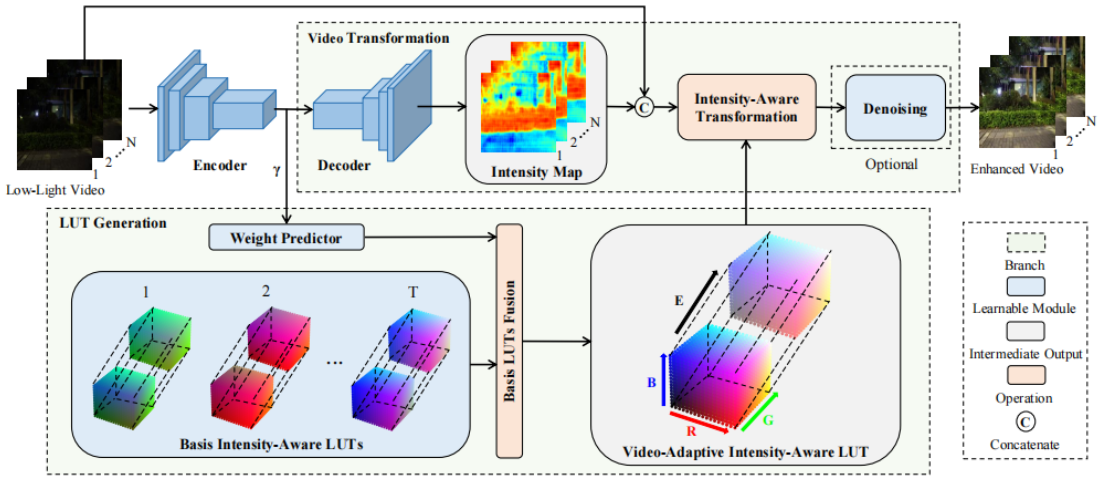

Figure 2 The architecture of FastLLVE.

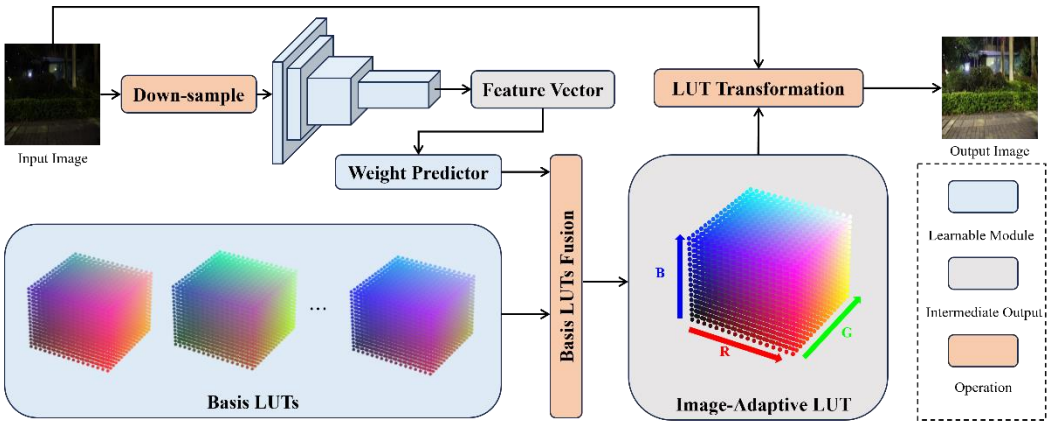

Figure 3 The architecture of adjusted model.

2.1 Feature Extraction:

Since existing research has shown that features extracted from a low-resolution image are sufficient to generate a LUT, the input image is first down-sampled to a size of 256×256 to improve efficiency. Then, an encoder implemented by a CNN is used to extract features from the processed image, and finally generate a feature vector for LUT Generation. It is worth noting that this CNN-based encoder is specifically designed to be lightweight. Due to this lightweight design, coupled with the down-sampling of input images as mentioned earlier, the impact of CNN on algorithm efficiency is minimized.

2.2 LUT Generation:

Utilizing the information transmitted through the feature vector, a weight predictor implemented by a fully-connected layer outputs weights one-to-one corresponding to basis LUTs, enabling the generated LUT used for image enhancement to be adaptive to the input image. Thanks to this automation advantage, LUT-based deep learning methods differ fundamentally from

traditional LUT tools, effectively eliminating the drawbacks of inflexibility and high calibration costs associated with traditional LUTs.

After obtaining the image-dependent weights, another fully-connected layer is employed to map these weights to all elements of the image-adaptive LUT. In addition, the learnable parameters of this fully-connected layer are all encoded basis LUTs. Therefore, the second fully-connected layer actually implement the weighted fusion of basis LUTs. In a word, two fully-connected layers map feature vectors to image-adaptive LUTs.

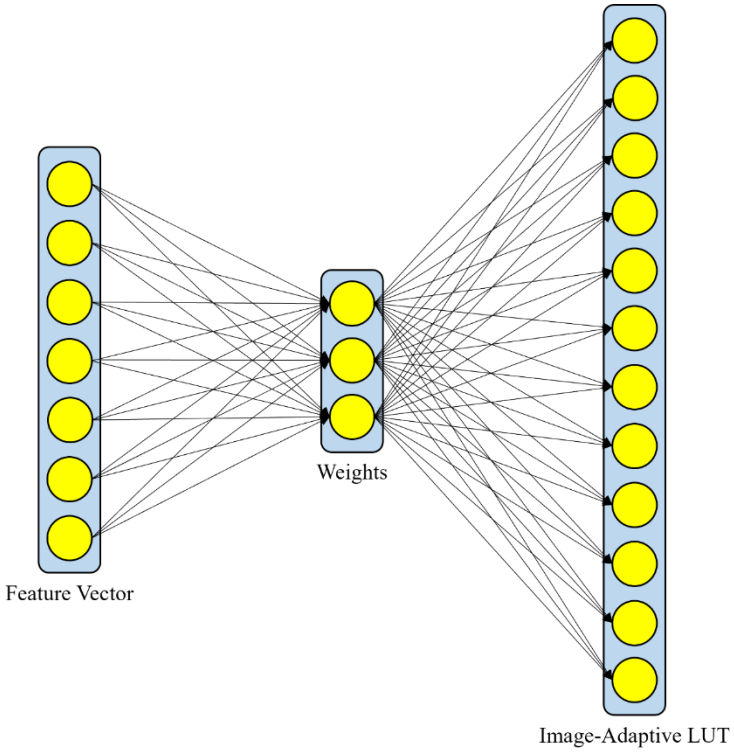

Figure 4 Two fully-connected layers map feature vectors to image-adaptive LUTs.

2.3 LUT Transformation:

For classical 3D LUT, it is primarily used for precise color correction and adjustment within a color space. It maps the pixel values of an input image to corresponding output values through a pre-calculated color mapping table, thereby achieving accurate control over the image's color. Since we already have the input image and its corresponding LUT, the operations in the LUT transformation stage are similar to traditional LUT tools, as they both involve looking up and performing linear interpolation based on a precomputed LUT to transform images.

Specifically, in the LUT transformation, for a pixel in the input image with RGB values of (r, g, b), we first transform the RGB values into the related index of the 3D LUT. The index is denoted as (x, y, z), which can be calculated as follows:

$$x = \frac{r}{255} \cdot N, y = \frac{g}{255} \cdot N, z = \frac{b}{255} \cdot N,$$

where N indicates the number of defined sampling points on each dimension of the 3D LUT. Then, as shown in Figure 5, the looking up operation is adopted to find the nearest 8 adjacent sampling points. We use $(\{i, i+1\}, \{j, j+1\}, \{k, k+1\})$ to denote the locations of the defined sampling points, which can be computed as follows:

$$i = \lfloor x \rfloor, j = \lfloor y \rfloor, k = \lfloor z \rfloor,$$

where $\lfloor \cdot \rfloor$ represents the floor function. As a results, the LUT transformation completely becomes trilinear interpolation, and the formula of the trilinear interpolation can be directly applied.

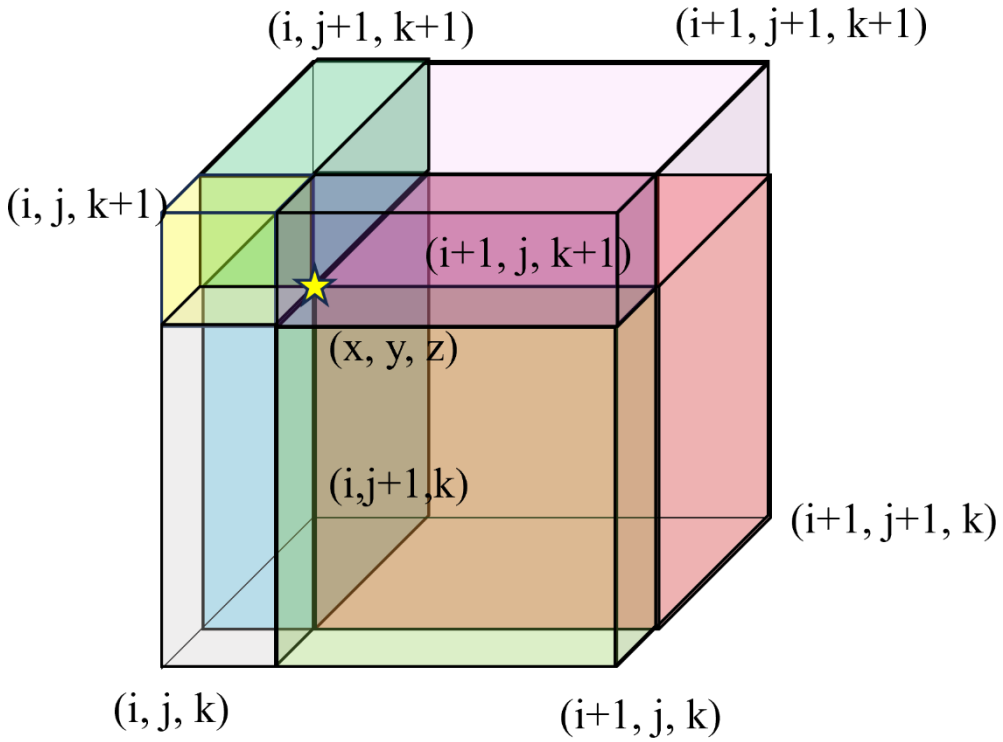

Figure 5 Illustration of the trilinear interpolation.

3. Project Progress:

We use generic low-light image datasets and overexposure image datasets to train the designed integrated image enhancement model based on 3D LUT. For low-light enhancement, we utilize the LOL-v2 dataset which consists of two versions synthetic dataset and real dataset. Besides, for overexposure correction, we adopt the overexposure portion of the MSEC dataset. On each dataset, we have trained two versions of the model, one version with a single basis LUT and another version with three basis LUTs.

Extensive quantitative and qualitative experiments are conducted based on those trained models. We compare our method with existing state-of-the-art methods using PSNR and SSIM metrics. As shown below, our method achieves superior performance compared to most existing methods while maintaining high efficiency, and it does not exhibit a significant gap compared to the state-of-the-art methods.

Methods	Dong	LIME	MF	SRIE	BIMEF	DRD
PSNR	17.26	15.24	18.73	17.34	17.85	15.47
SSIM	0.527	0.470	0.559	0.686	0.653	0.567
Methods	RPM	SID	DeepUPE	KIND	DeepLPF	FIDE
PSNR	17.34	13.24	13.27	14.74	14.10	16.85
SSIM	0.686	0.442	0.452	0.641	0.480	0.678

Methods	LPNet	MIR-Net	RF	Band	EG	Retinex
PSNR	17.80	20.02	14.05	20.29	18.23	18.37
SSIM	0.792	0.820	0.458	**0.831**	0.617	0.723
Methods	Sparse	IPT	Uformer	SNR	3DLUT-1	3DLUT-3
PSNR	20.06	19.80	18.82	**21.48**	**21.88**	21.55
SSIM	0.816	0.813	0.771	**0.849**	0.806	0.802

Table 1 Quantitative comparison on LOL-v2-real.

Methods	Dong	LIME	MF	SRIE	BIMEF	DRD
PSNR	16.90	16.88	17.50	14.50	17.20	17.13
SSIM	0.749	0.776	0.751	0.616	0.713	0.798
Methods	RPM	SID	DeepUPE	KIND	DeepLPF	FIDE
PSNR	17.15	15.04	15.08	13.29	16.02	15.20
SSIM	0.727	0.610	0.623	0.578	0.587	0.612
Methods	LPNet	MIR-Net	RF	Band	EG	Retinex
PSNR	19.51	21.94	15.97	**23.22**	16.57	1655
SSIM	0.846	0.876	0.632	**0.927**	0.734	0.652
Methods	Sparse	IPT	Uformer	SNR	3DLUT-1	3DLUT-3
PSNR	22.05	18.30	19.66	**24.14**	21.40	23.16
SSIM	0.905	0.811	0.871	**0.928**	0.858	0.900

Table 2 Quantitative comparison on LOL-v2- synthetic.

Methods	HE	ZERO-DCE	SID	MSEC	DRBN	3DLUT-1	3DLUT-3
PSNR	16.53	10.40	18.83	19.79	19.37	**19.94**	**20.21**
SSIM	0.699	0.514	0.806	0.816	**0.832**	0.828	**0.831**

Table 3 Quantitative comparison on MSEC.

In addition to performance, we also compare the processing time of a single 600×400 image on a RTX 4090 GPU with existing high-performance methods such as MIR-Net and SNR (Band and Sparse are not open-source and the algorithms themselves are complex, hence not considered). As shown below, our method significantly outperforms existing state-of-the-art methods in terms of efficiency, and even achieves real-time processing speed. Additional experiment on processing single high-resolution images of 1920×1080 pixels on CPU further demonstrates that our method can process high-resolution images in real-time without relying on GPU.

Methods	MIR-Net	SNR	3DLUT-1	3DLUT-3
Runtime (s)	0.303	0.018	**0.002**	**0.002**

Table 4 Quantitative comparison of runtime on GPU.

Methods	3DLUT-1	3DLUT-3
Runtime (s)	0.008	0.010

Table 5 Quantitative experiment of runtime on CPU.

Input 3DLUT-1 3DLUT-2

Figure 6 Some qualitative results.

In terms of hardware environment required for subsequent development work, according to Mr. Du's guidance and the R-CarV4H_V4M_V3H_V3M_SDK_StartupGuide documentation, we have installed the development environment (R-CarSDK) on our computer, and ran the sample code on it - showing two images (the original image and the enhanced image in the later work) on the left and right sides of the monitor. In terms of connection, we used a wired method to connect the EVA-board to the computer.