1.5em 0pt

Universiteit Leiden

Master Computer Science

Real-Time Low-Light Image and Video Enhancement with Lightweight Retinex-based Network

Name:	Yağmur Doğan
Student ID:	s3910024
Date:	05/08/2025
Specialisation:	Data Science
1st supervisor:	Dr. Hazel R. Doughty
2nd supervisor:	Dr. Rita Pucci

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Acknowledgements

I would like to express my deepest gratitude to my first supervisor, Dr. Hazel R. Doughty, for her invaluable guidance, support, patience, and encouragement throughout every stage of this research. Her feedback and advice have helped me stay on track and made this process so much more manageable. I am especially thankful for how approachable and supportive she's been whenever I had questions or doubts. Her presence as my supervisor truly made a difference. Her mentorship not only guided the technical aspects of this thesis but also provided me with the confidence and motivation to overcome challenges and stay focused on my academic goals.

I also sincerely thank Dr. Rita Pucci for her role as my second supervisor. Although our interactions were limited, I am grateful for her presence on my supervision team and for her support in helping to complete my thesis. Her involvement is genuinely appreciated.

I also want to thank my family and friends for always being there for me, cheering me up, listening to my complaints and keeping me sane during all the stressful times of this thesis process. Your support means more than I can put into words, and I couldn't have done this without you.

Additionally, I would like to acknowledge the support of the Republic of Türkiye, which has contributed to my education and research by providing the necessary resources and opportunities during my studies abroad. This research has been carried out with the backing of the Republic of Türkiye.

This work was performed using the compute resources from the Academic Leiden Interdisciplinary Cluster Environment (ALICE) provided by Leiden University.

Abstract

Low-light conditions pose significant challenges for computer vision systems, particularly in real-time applications such as autonomous vehicles and visual monitoring systems. While existing deep learning approaches for low-light enhancement demonstrate high-quality results, they typically require substantial computational resources that make real-time processing impractical.

This thesis presents a lightweight, real-time solution for low-light image and video enhancement based on Retinex theory that addresses the fundamental trade-off between enhancement quality and processing speed. Building upon the KinD++ [51] architecture, we add MobileNet-style [16] optimizations, including depthwise separable convolutions and channel width multipliers, to significantly reduce computational complexity while preserving visual quality. Our approach features a dual-branch network structure for reflectance and illumination processing, using a width multiplier of 0.5, which results in a lightweight model size with less than 35K trainable parameters and with its compact size, the model achieves real-time performance of over 80 FPS on GPU hardware, making it suitable for real-time low-light enhancement tasks.

We perform comprehensive evaluations using both quantitative metrics (PSNR, SSIM, LPIPS, DeltaE) and qualitative visual analysis. Our experimental results demonstrate that models trained without GAN components, particularly using Mutual Input (MI) Loss, outperform GAN-based variants across most metrics. The proposed method achieves promising results.

Comparative analysis with state-of-the-art methods shows that our approach achieves competitive or superior performance on quantitative metrics while offering significant advantages in computational efficiency and practical deployment. The solution supports various input sources including images, video files, and RTSP streams, with a multi-threaded processing and efficient memory management for continuous video processing.

This work demonstrates that high-quality low-light video enhancement can be achieved in real-time without relying on computationally expensive architectures, making advanced enhancement capabilities more accessible for resource-constrained applications where real-time performance is critical.

Table of Contents

1 Introduction

Computer vision is a field of artificial intelligence that uses machine learning, deep learning techniques and neural networks to teach computers to see, observe and get meaningful information from visual inputs like images and videos. Computer vision systems have become an inseparable part of the modern technology with the applications in many fields. To make accurate decisions and provide reliable outputs, the quality of the visual input is crucial.

One of the most significant challenges that computer vision systems face is low-light conditions, where natural or artificial illumination is not sufficient, particularly in applications that require real-time analysis such as autonomous vehicles and medical imaging. Underexposed frames often suffer from reduced contrast, high noise levels, and loss of structural detail, which can severely decrease the performance of computer vision applications such as object detection and scene understanding. While low-light image enhancement has been widely studied, video enhancement in real-time introduces additional complexities, including temporal consistency and computational efficiency.

In low-light environments, several factors affect and decrease image quality. The reduced photon count leads to an increased signal-to-noise ratio (SNR), making it difficult for sensors to capture weak light signals accurately. This results in significant noise and artifacts in the captured images. Additionally, color information becomes distorted in dark regions, leading to inaccurate color reproduction and a reduced color range. The dynamic range of the captured images is also making it challenging to preserve details in both bright and dark regions simultaneously.

While low-light image enhancement has been extensively studied, video enhancement presents additional complexities that make it a more challenging problem. The temporal nature of video requires maintaining visual consistency across frames while handling motion between them. This becomes particularly difficult in low-light conditions where longer exposure times are often necessary, leading to motion blur and the loss of sharpness in moving objects. Furthermore, the need to process multiple frames per second in real-time adds another layer of complexity to the enhancement process.

Real-time low-light video enhancement plays a critical role in modern applications. For instance, in automotive systems, the ability to enhance low-light video feeds in real-time is crucial for night-time driving assistance, pedestrian detection, and lane detection. Similarly, in security applications, the capability to process and enhance low-light video streams without delay is essential for

24/7 monitoring, threat detection, and real-time alert systems. Medical imaging applications also benefit from real-time enhancement capabilities, particularly in surgical guidance, endoscopy, and patient monitoring in low-light conditions.

Current solutions in the field face significant limitations that obstruct their practical application. Deep learning models, while capable of producing high-quality enhancements, often require substantial computational resources and memory, making them unsuitable for real-time processing on standard hardware. There exists a fundamental trade-off between enhancement quality and processing speed, where high-quality enhancement methods are typically too slow for real-time processing, while faster methods may compromise the quality of the enhancement. Additionally, hardware constraints in deployment environments, including limited GPU resources, power consumption considerations, and cost constraints, further complicate the implementation of real-time solutions.

This thesis proposes a lightweight, real-time solution for low-light image and video enhancement based on Retinex theory [23]. Building on the baseline of KinD++ [51], our approach introduces MobileNet-style optimizations [16], specifically incorporating depthwise separable convolutions to significantly reduce computational complexity while maintaining visual quality. Unlike existing methods, our model is designed with real-time performance in mind, including a processing pipeline that supports multi-threaded execution for efficient resource utilization and batch processing for improved GPU throughput. The solution includes an advanced processing pipeline that utilizes multi-threaded processing for efficient resource utilization and batch processing for improved GPU utilization. The system is designed to support various input sources, including images, video files and even RTSP streams while providing real-time performance monitoring and error handling.

The proposed solution focuses on enhancing detail preservation in dark regions, improving color accuracy, and reducing noise and artifacts through efficient frame-by-frame processing. By optimizing the network design for real-time performance with under 35K parameters and implementing efficient memory management for continuous video processing, our approach aims to bridge the gap between high-quality enhancement and real-time processing capabilities by ensuring high speed which we reached more than 80 FPS. This work aims to make low-light image and video enhancement more practical and accessible for real-world applications, particularly in scenarios where real-time processing speed is crucial for decision making and system response.

1.1 Research Objectives & Questions

Low-light video enhancement presents a major challenge in computer vision applications such as robotics and autonomous systems, where clarity and visibility are critical for tasks like object detection, tracking, and navigation. In these scenarios, underexposed frames often suffer from reduced contrast, color distortion, high noise levels, and loss of structural details that severely degrade downstream performance.

While deep learning-based approaches have achieved impressive results in low-light image enhancement, many rely on large models or expensive computations, making them impractical for real-time video processing. Video enhancement also requires maintaining temporal consistency between frames, which adds further complexity. Existing methods that offer strong visual results in offline settings often fall short in real-time applications, especially on resource-constrained platforms such as embedded systems, mobile devices, or live video pipelines.

This thesis aims to address these limitations by proposing a lightweight, real-time low-light enhancement model tailored for continuous video streams. The approach builds upon Retinex theory [23], a perceptually grounded model of human vision, and uses KinD++ [51] as a baseline enhancement method. To make the architecture suitable for real-time applications, we cooperate with MobileNet-style optimizations [16], including depthwise separable convolutions and channel width multipliers, which significantly reduce the number of parameters and computational cost. The model is further supported by system-level improvements such as multithreaded frame handling, batch-based inference, and runtime memory optimization to meet real-time constraints.

The proposed solution is evaluated using both standard image quality metrics (PSNR, SSIM, LPIPS) and perceptual comparisons on a benchmark low-light dataset. Additionally, runtime performance, including speed, memory footprint, and GPU utilization, is measured to validate its suitability for real-world deployment.

This research aims to address fundamental challenges in real time low-light enhancement by investigating the following research questions:

1. **Main Research Question**
   How can we design computationally efficient network architectures specifically optimized for low-light video enhancement that maintain acceptable visual quality while achieving real time performance?

2. **Research Question 2**
   How does the inclusion or exclusion of GAN based training affect the performance and visual quality of low light video enhancement models?

3. **Research Question 3**
   How does our proposed approach compare with existing state-of-the-art methods in terms of quantitative metrics assessment?

To address these research questions, this study presents a combined method that brings together Retinex-based image decomposition with lightweight deep learning techniques. The approach uses the loss functions from KinD++ [51], which are effective for low-light enhancement, and applies MobileNet style [16] depthwise separable convolutions to reduce model size and computation. Our primary objective is to create an efficient network that performs well in real-time settings while maintaining good visual enhancement quality. Then with an ablation study that we made where we compare the usage of different loss functions and inclusion and exclusion of GAN based training we observed the effects.

1.2 Thesis Organization

This thesis is organized into eight chapters that systematically address the research problem of efficient real time low-light video enhancement. The structure follows a logical progression from theoretical foundations to practical implementation and evaluation.

Chapter 2: Related Work introduces existing literature that we have reviewed, especially the ones that are relevant to our research, providing critical analysis of current state-of-the-art methods in real time low-light enhancement and identifying gaps that motivate the proposed approach.

Chapter 3: Background provides the technical background to low-light enhancement, especially the methods that we used in our thesis. The chapter begins with an examination of some methodological approaches. Additionally, the chapter covers fundamental deep learning components essential to understand the proposed approach, including convolutional operations, efficient network architectures, training optimizations, and architectural design patterns. The chapter also introduces loss functions, training strategies, and data augmentation techniques that form the technical background for the methodology.

Chapter 4: Methodology presents the core technical contribution of this thesis. The chapter begins with the theoretical framework, covering Retinex theory

[23] and RetinexNet [47] foundations, the role of convolutional neural networks in image and video enhancement, and the principles of depthwise separable convolutions. It then details the proposed architecture and network structure design. Implementation details are thoroughly covered, including Sata Preprocessing, Optimization Configuration, Learning rate scheduling and mixed precision training. We also cover depthwise separable convolution implementation and width multiplier optimization techniques.

Chapter 5: Data and Preprocessing describes the datasets used in the study and the preprocessing pipeline developed to prepare data for training and evaluation. This chapter ensures reproducibility by providing detailed information about data preparation procedures.

Chapter 6: Experiments & Results presents the experimental evaluation of the proposed method. This chapter includes both quantitative performance analysis using standard image and video quality metrics and qualitative visual assessment. The results demonstrate the effectiveness of the proposed approach in achieving the balance between computational efficiency and enhancement quality.

Chapter 7: Conclusion synthesizes the key findings of the research, evaluates the extent to which the research objectives have been achieved, answers of the research questions and it also discusses the broader implications of the work for real time video processing applications.

Chapter 8: Future Work identifies potential extensions and improvements to the current approach, outlining promising directions for continued research in efficient real time low-light enhancement and related areas.

2 Related Work

In this section, we discuss the methods from our literature review on low-light enhancement of both images and videos. We evaluate them in terms of methodologies, supervision types, network architectures, contributions, limitations, and evaluation metrics.

2.1 Methods Used for Low-Light Enhancement

A wide range of methods, both classical and learning based methods, have been developed to address low-light conditions in images and videos. Below is an overview of the foundational approaches that form the basis of most state-of-the-art methods today.

2.1.1 Histogram Equalization (HE)

Histogram Equalization is one of the simplest and earliest contrast enhancement techniques. It works by spreading out the most frequent intensity values in an image, thereby increasing global contrast. This technique assumes that a uniform distribution of pixel values is visually better, which is often true for natural scenes. However, it does not consider local context and as a result, it may introduce noise or distort textures in low-light regions. Even though histogram equalization was introduced in the early stages of image processing, it is thoroughly explained in detail in the book by Gonzalez and Woods [10].

2.1.2 Gamma Correction (GC)

Gamma Correction is a nonlinear operation that brightens or darkens an image based on a specific gamma value. It is particularly useful for adjusting mid-range brightness levels and is widely used in image processing pipelines, including display calibration. However, gamma correction requires careful tuning and is typically applied globally, which may not be ideal for scenes with varying lighting. Although the concept of gamma like transformations dates back to earlier developments in photographic science, a detailed explanation of the gamma correction method in the context of digital imaging is provided in the book by Poynton [36].

2.1.3 Retinex Theory-Based Methods

Retinex theory [23], which is short for Retina and Cortex, is based on the idea that an observed image can be separated into reflectance and illumination components. The reflectance contains the true color and texture information, while the illumination varies with lighting conditions. Enhancement is performed by

estimating and adjusting the illumination while preserving reflectance. This approach enables selective brightening of dark regions without overexposing bright ones. It forms the conceptual basis for many modern algorithms, including those that use learning-based decomposition.

Several methods are built on Retinex theory. For example, Beyond Brightening (KinD++) [51] separates illumination and reflectance for more targeted enhancement and denoising. Meanwhile, Zero-Reference Physical Quadruple Priors [46] use physical priors and pretrained diffusion models to enable enhancement without paired data.

2.1.4 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks have become a cornerstone in image processing tasks due to their ability to learn spatial hierarchies of features [24]. In the context of low-light enhancement, CNNs are trained to map dark input images to their well-lit counterparts [30]. They automatically learn filters that enhance texture, reduce noise, and correct colors, all in one unified framework. CNN-based models are highly effective and can operate in real time with the right optimization, making them suitable for both image and video enhancement.

2.1.5 Generative Adversarial Networks (GANs)

Generative Adversarial Networks involve two networks which are a generator and a discriminator competing in a minimax game [11]. In low-light enhancement, the generator learns to produce enhanced images that appear natural, while the discriminator evaluates whether an image looks realistic or not. This setup encourages the model to produce visually pleasing outputs with natural textures and lighting [3]. GANs are especially useful when training data lacks exact ground truth, as they can learn enhancement patterns from unpaired data [19].

Some low-light image enhancement methods utilize Generative Adversarial Networks (GANs). EnlightenGAN [19] is the first to apply GANs in an unsupervised setting, i.e., without paired low-light and normal-light images. However, the model is large and not suitable for real-time use.

The ImCam framework by Dai et al. [5] combines the Retinex model with GANs to enhance surveillance images in the wild. It first applies illumination correction, then uses a GAN for enhancement, improving downstream classification accuracy on public surveillance datasets. Despite strong results, its generalizability beyond surveillance and real time applicability remain uncertain.

2.1.6 Temporal Models

For video enhancement, maintaining consistency across frames is essential. 3D CNNs extend standard CNNs into the temporal dimension, allowing the model to learn spatio temporal features [41]. Recurrent models, such as LSTM or GRU-based architectures, maintain a memory of previous frames and help smooth transitions between them. Additionally, optical flow techniques align neighboring frames before enhancement, reducing motion blur and temporal flickering [2].

2.2 Classical Video Enhancement Methods

Video enhancement studies such as Kim et al. [20] use Kalman filtering and nonlocal means (NLM) for noise removal, with gamma correction for contrast enhancement. Their method works directly on CFA raw data, reducing memory use, but suffers from color over saturation and format limitations.

Ding et al. [6] introduced Sparse Codes Fusion (SCF) for surveillance videos, which fails under camera motion. Similarly, FACE by Ulhaq et al. [42] works well in static scenes but degrades under dynamic conditions.

Zhang et al. [50] use progressive fusion for underexposed videos but struggle with noise artifacts and fast motion. Aydın et al. [1] propose temporally coherent tone mapping yet chrominance handling is limited.

Lee et al. [25] use fuzzy C-means clustering for adaptive contrast enhancement, but it is sensitive to initial parameters and noise.

2.3 Real Time Based Approaches

Rajan et al. [38] explore privacy-preserving enhancement, combining quality improvement with secure video storage. Similarly, real time driving safety is tackled by Mandal et al. [33] using adaptive gamma correction and bright channel prior, improving visibility without high computational cost.

BSVD [37] enhances temporal video denoising with bidirectional buffers and is suitable for online inference. Edge aware frameworks like in [22] perform lightweight retraining on edge devices to maintain detection accuracy under environmental changes.

2.4 Low-Light Image Enhancement

Autoencoder based LLNet [30] enhances brightness and reduces noise with stacked sparse denoising layers. LIVENet [32] incorporates atmospheric scattering and texture refinement for real-world low-light conditions.

Zero-DCE [12] and Zero-DCE++ [26] estimate pixel-wise curves without reference data, offering speed and simplicity however often at the cost of visual quality.

SCI [31] prioritizes speed and robustness via flexible learning, whereas Starenhancer [40] focuses on real time enhancement with minimal latency.

R2RNet [14] decomposes images into illumination, reflectance, and denoising branches trained on real-world data, performing strongly on downstream tasks like face detection or object detection.

2.5 Low-Light Video Enhancement

Mandal et al. [33] propose a real-time enhancement framework based on adaptive gamma correction and bright channel prior, offering fast yet effective enhancement for streaming video. BSVD [37] integrates bidirectional temporal buffers to denoise low-light video while maintaining temporal coherence for online processing. Edge Adaptive [22] combines edge-aware streams with CNNs, enabling lightweight retraining in adverse video environments such as fog, motion, or extreme low-light. FastLLVE [28] leverages intensity aware lookup tables (IALUT) with temporal consistency modules to achieve high-quality enhancement at the fastest inference speeds among LUT-based video methods.

Kim et al. [20] present a classical Kalman filter-based enhancement that processes CFA raw video frames with low memory overhead, while Ding et al. [6] apply sparse code fusion for contextual enhancement in video. FACE [42] introduces a rule-based night vision pipeline with DNN integration for full automation, and Zhang et al. [50] apply perception-driven fusion strategies for progressive video enhancement. VLight [53] brings a lightweight CNN solution optimized for mobile video enhancement with a single tunable parameter for brightness control.

Aydın et al. [1] address temporal coherence explicitly through edge-aware filtering in HDR video tone mapping.

2.6 Our Baseline and Approach

A milestone in Retinex-based enhancement is KinD [52] and its improved successor KinD++ [51]. KinD++ incorporates structure aware smoothness loss and layer-wise decomposition to better isolate illumination from reflectance. It provides significant improvement in both PSNR and perceptual metrics across multiple datasets and serves as a core baseline in the field.

This thesis adopts KinD++ as our primary methodological foundation, particularly for its illumination-reflectance separation and supervised training strategy. However, to make our method real time and deployable on edge devices, we also draw on techniques from MobileNets [16], integrating depthwise separable convolutions to reduce computational complexity and parameter count. By combining these techniques, this hybrid approach promises a fast low-light enhancement while ensuring speed and preserving image quality.

To sum up, although current state-of-the-art methods demonstrate strong performance in low-light enhancement, they often face trade-offs between computational efficiency and visual quality. To address this gap and effectively respond to our research questions, we propose an approach that builds on the KinD++ [51] framework by integrating depthwise separable convolutions inspired by MobileNets [16]. This combination aims to improve both processing speed and enhancement quality, facilitating real time deployment without compromising visual fidelity.

Method	Architecture Type	Supervision Type	I/O Type	Real-time Suitability	Input Type	Key Feature / Contribution
EnlightenGAN [19]	GAN	Unsupervised	Image	Low	RGB Image	First GAN-based unsupervised low-light enhancer
ImCam [5]	Retinex + GAN	Supervised	Image	Low	Surveillance RGB	Retinex correction + GAN; improves downstream classification
Kim et al. [20]	Classical / Kalman	Supervised	Video	Medium	CFA Raw Data	Kalman filtering, gamma correction, low memory use
Ding et al. [6]	Sparse Coding	Supervised	Video	Low	RGB Video	Context enhancement with sparse code fusion (SCF)
FACE [42]	Rule-based + DNN	Supervised	Video	Low	RGB Video	Fully automated color night vision pipeline
Zhang et al. [50]	Fusion-based	Supervised	Video	Low	RGB Video	Perception-driven progressive fusion
Aydın et al. [1]	Tone Mapping	Supervised	Video	Medium	HDR Video	Temporally coherent tone mapping + edge-aware filter
Lee et al. [25]	Contrast Stretching	Supervised	Image	Low	HSV Image	Adaptive partitioning via fuzzy C-means
Rajan et al. [38]	Classical + Cryptography	Supervised	Video	Medium	RGB Video	Secret sharing + enhancement for surveillance
Mandal et al. [33]	Classical + Prior	Supervised	Video	High	RGB Frames	Fast enhancement with adaptive gamma + bright channel prior
BSVD [37]	DNN	Supervised	Video	High	RGB Frames	Bidirectional temporal buffers for online denoising
Edge Adaptive [22]	CNN + Edge-aware	Supervised	Video	High	Edge Streams	Lightweight retraining on adverse environments
LLNet [30]	Autoencoder (DNN)	Supervised	Image	Medium	RGB Image	Stacked sparse denoising + adaptive brightening
LIVENet [32]	CNN + Retinex	Supervised	Image	Medium	RGB Image	Scattering model + spatial feature transforms
Zero-DCE [12]	Curve Estimation	Zero-Reference	Image	High	RGB Image	Fast enhancement using deep curve estimation
Zero-DCE++ [26]	Curve Estimation	Zero-Reference	Image	High	RGB Image	Improved Zero-DCE; faster inference
SCI [31]	CNN	Supervised	Image	High	RGB Image	Fast, flexible, robust enhancement for downstream tasks
Starenhancer [40]	CNN	Supervised	Image	High	RGB Image	Real-time, style-aware enhancement
R2RNet [14]	CNN + Decomposition	Supervised	Image	Medium	Real-paired RGB	Real-world dataset with three-branch decomposition
VLight [53]	Lightweight CNN	Supervised	Video	High	RGB Frames	Smartphone-optimized; single parameter control
Beyond Brightening [51]	Retinex-based CNN	Supervised	Image	Medium	RGB Image	Illumination-reflectance separation without GT illumination
Physical Priors [46]	Diffusion + Priors	Zero-Reference	Image	Medium	RGB Image	Quadruple priors + pretrained diffusion model
FastLLVE [28]	Lookup Table (IALUT)	Supervised	Video	High	RGB Frames	Fastest LUT-based method with temporal consistency
KinD++ [51]	Retinex-based CNN	Supervised	Image	Medium	RGB Image	Structure-aware decomposition + global adjustment
Our Method	**Retinex + MobileNet**	**Supervised**	**Image / Video**	**High**	**RGB Frames**	**Based on KinD++ framework + depthwise separable convolutions to get fast yet accurate results**

Table 1: Comparison of related methods in terms of architecture type, supervision type, target domain, real-time suitability, input type and key features.

3 Background

Enhancing low-light images (or videos) is a challenging task due to the complex interactions between noise, illumination, and reflectance. An ideal enhancement technique must not only brighten dark regions but also suppress noise, preserve texture, maintain color accuracy, and avoid overexposure or artifacts. These competing goals create trade-offs between visual quality and computational efficiency. Over the years, researchers have developed a wide range of techniques to address this problem.

In this section, we introduce the components and prior knowledge to the low-light image and video enhancement, which we are using in our approach.

3.1 Convolutional Neural Networks

Motivation for CNN-Based Low-Light Enhancement

Convolutional Neural Networks (CNNs) have demonstrated strong effectiveness in low-light image enhancement tasks due to their ability to learn hierarchical feature representations.

In many enhancement frameworks, CNNs are structured into distinct components, such as a decomposition module that separates an image into illumination and reflectance layers and dedicated enhancement modules that process each layer independently. This approach is motivated by the understanding that low-light images typically exhibit two types of degradation: insufficient illumination and a loss of fine detail.

The Challenge of Traditional CNNs

CNNs have become the standard approach for image enhancement tasks due to their ability to capture both local and global image patterns. Traditional CNNs use standard convolutions where each filter processes all input channels simultaneously, creating rich feature representations through cross-channel interactions. However, this approach can be computationally expensive and parameter-heavy, especially for deep networks, making real-time deployment challenging on resource-constrained devices.

Hierarchical Feature Learning

In the context of low-light image enhancement, our methodology takes advan-

tage of the power of CNNs through a carefully designed architecture that leverages hierarchical feature extraction. The network employs a progressive learning approach, where each layer builds upon the representations learned by previous layers, creating increasingly sophisticated understanding of image content. Early layers capture low-level features such as edges, textures, and local contrast patterns, while deeper layers learn high-level semantic information about lighting conditions, material properties, and spatial relationships.

This hierarchical structure is particularly valuable for low-light enhancement because it allows the network to simultaneously address multiple aspects of image degradation. Low-level features help preserve fine details and textures that are often lost in dark conditions, while high-level features enable the network to understand the global lighting context and make informed decisions about enhancement strategies.

The Role of the Separation of Illumination and Reflactance

The decomposition network utilizes convolutional layers to learn the separation of illumination and reflectance components, a process that is fundamental to effective low-light enhancement. This separation is motivated by the physical principles of the Retinex theory, which states that the observed image intensity is the product of illumination (lighting conditions) and reflectance (intrinsic surface properties). By separating these components, the network can address each type of degradation independently and more effectively.

Separation is crucial in low-light conditions, both illumination and reflectance information are compromised, but in different ways. Illumination degradation occurs as overall darkness and uneven lighting, while reflectance degradation results in loss of detail, color distortion, and reduced contrast. Traditional end-to-end enhancement approaches treat these problems as a single optimization task, often leading to suboptimal results where improving one aspect degrades the other. Thus, this separation approach allows the network to apply specialized enhancement strategies to each component, resulting in more natural and effective enhancement.

3.2 Retinex Theory and RetinexNet

Our work is based on the Retinex theory, which declares that any image can be decomposed into two components: reflectance and illumination. The theory's name comes from the combination of "retina" and "cortex," emphasizing the

biological inspiration behind this approach. The Retinex theory, first proposed by Land and McCann [23], suggests that human color perception is based on three independent mechanisms: RGB (red, green, and blue) and that the perceived color of an object is determined by the ratio of light reflected from the object and the light reflected from surrounding objects. In the implementation, it is modeled this relationship as:

$$I = R \otimes L, \tag{1}$$

where I represents an image, R represents the reflectance component and L represents the illumination component.

This theory led to the development of RetinexNet, a CNN based model, that has been introduced in paper Deep Retinex Decomposition for Low-Light Enhancement [47]. RetinexNet extends Retinex theory by learning the decomposition through deep neural networks, separately enhancing both components and recombining them to produce the final enhanced image. The network consists of three primary components working together in harmony: a decomposition network that separates the input image into its illumination and reflectance components, and two specialized enhancement networks that independently process these components before recombining them to produce the enhanced image.

The original RetinexNet [47] implements the Retinex image formation model by decomposing a low-light image into reflectance and illumination components using a deep neural network framework. The model consists of three sub-networks: Decom-Net, Enhance-Net, and a reconstruction stage.

Decom-Net learns to extract a shared reflectance and separate illumination maps for the low-light and normal-light input images. The training process enforces a reflectance consistency loss to ensure the shared reflectance remains the same across lighting conditions, and a structure-aware total variation loss is applied to the illumination to encourage smoothness while preserving structural boundaries.

Enhance-Net operates on the illumination map predicted from the low-light image. It employs an encoder–decoder architecture with multi-scale concatenation to capture both local and global context for effective illumination refinement. The refined illumination is constrained to the [0,1] range through a sigmoid activation.

The reconstruction stage performs element-wise multiplication between the enhanced illumination and the original reflectance components, following the Retinex [23] equation:

$$EnhancedImage = EnhancedIllumination \otimes Reflectance. \tag{2}$$

This multiplication preserves the physical relationship between the components while applying the enhancement effects primarily through illumination adjustment.

The total loss function combines the reconstruction loss, reflectance consistency loss, and the structure-aware smoothness loss to guide the decomposition and enhancement networks.

Below in Figure 1, the RetinexNet framework that proposed in paper Deep Retinex Decomposition for Low-Light Enhancement [47] can be seen.

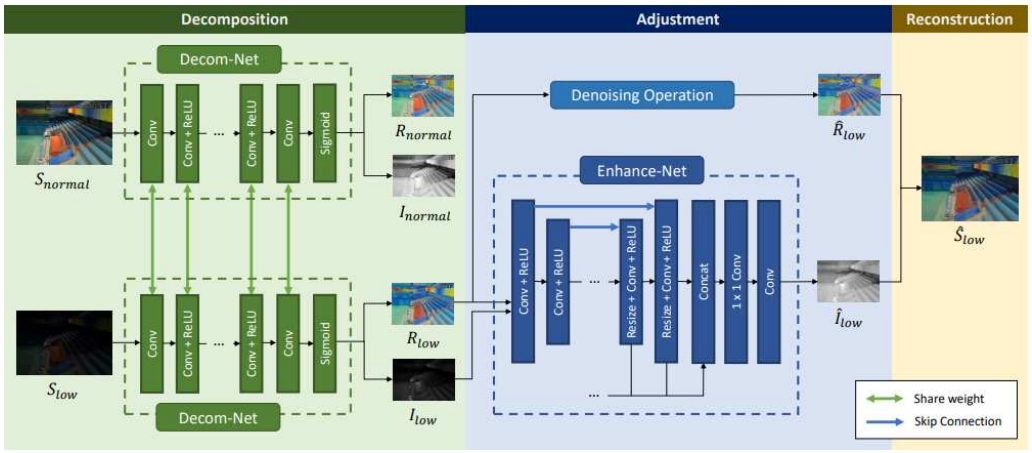


Figure 1: The proposed framework for RetinexNet from the paper [47]. The enhancement process has three steps. The decomposition step decomposes the input image into reflectance and illumination. Then an encoder-decoder based Enhance-Net brightens up the illumination. Multi-scale concatenation is introduced to adjust the illumination from multi-scale perspectives. Final step is the reconstruction of the adjusted illumination and reflectance to get the enhanced result.

3.3 KinD++

KinD++ [51] is an advanced low-light image enhancement framework that builds upon Retinex theory [23], incorporating a refined decomposition and reconstruction architecture. The framework is designed around three core stages, decomposition, enhancement and reconstruction.

The process begins with a Layer Decomposition network, which takes a low-light image and separates it into its reflectance and illumination components, following the Retinex theory. This network is trained to learn the intrinsic struc-

tural and lighting components of the image by convolutional layers.

The reflectance branch further processes the extracted reflectance using a series of convolutional and residual blocks. These layers are designed to preserve detailed textures and suppress noise, which is common in low-light images. Residual connections ensure stable learning, also ensuring that critical details of the scene are preserved.

The illumination branch enhances the illumination component through convolutional layers designed to improve brightness and contrast while maintaining spatial smoothness. Structure aware total variation loss is employed to preserve edge information and prevent artifacts in the illumination map.

In the reconstruction stage, the refined reflectance and illumination maps are recombined using element-wise multiplication:

$$EnhancedImage = EnhancedIllumination \otimes Reflectance. \quad (3)$$

To handle both global and local features effectively, KinD++ integrates multi-scale processing, enabling the network to respond to varying feature sizes and illumination levels.

Throughout the architecture, multiple loss functions are strategically integrated to the training process. These include reconstruction losses that ensure the enhanced image resembles the ground truth, decomposition consistency losses that maintain Retinex theory compliance, and perceptual losses that ensure visual quality.

Below in Figure 2, the framework of Kind++ [51] can be seen.

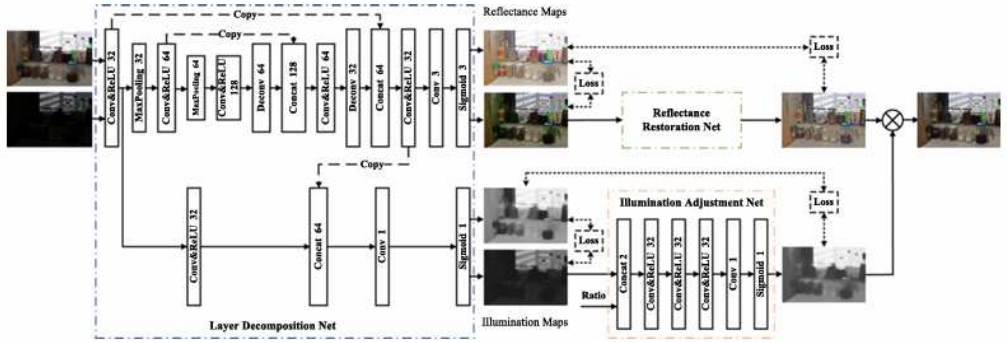


Figure 2: The network architecture of KinD++. Two branches from the input image are the reflectance and illumination. The model is divided into three modules, layer decomposition, reflectance restoration, and illumination adjustment. $\otimes$ is the element-wise multiplication. The digits are channel numbers

3.4 Fundamental Deep Learning Components

In this section we introduce the fundamental deep learning components that are necessary to understand our work.

3.4.1 Convolutional Operations

Kernel Size and Filters

Convolutional Neural Networks (CNNs) operate by applying learnable filters, which are known as kernels, in input images to extract hierarchical features. These filters slide over local regions, allowing the network to detect patterns such as edges, textures, and more complex structures at deeper layers. Typical kernel sizes are 3×3, 5×5 and 7×7, with the 3×3 kernel being the most widely adopted due to its balance between computational efficiency and representational capacity. Stacking multiple layers of smaller kernels, particularly 3×3, effectively increases the input region while keeping parameter counts and computational costs low.

Convolution Types

Standard convolutions apply a set of filters across all input channels simultaneously, producing output feature maps that integrate both spatial and channel wise information. This operation enables the network to model complex spatial

dependencies and textures, which is particularly crucial for tasks like low-light image and video enhancement. In such contexts, capturing subtle gradients and fine grained structural details can significantly influence the restoration quality. Variants of standard convolution, such as depthwise separable and dilated convolutions, have also been introduced to reduce computation or expand input regions, but standard convolution remains a foundational operation for capturing dense local features in early and intermediate layers of enhancement models. These convolution types are introduced in paper by LeCun et al. [24] in details.

3.4.2 Network Architectures

Depthwise Separable Convolutions
Our methodology incorporates MobileNet style [16] depthwise separable convolutions as a key architectural innovation to improve computational efficiency while maintaining enhancement quality.

This approach factorizes the standard convolution operation into two separate steps: a depthwise convolution that performs lightweight filtering, and a pointwise convolution that combines the filtered outputs. The depthwise convolution applies a single filter to each input channel independently, significantly reducing computational complexity compared to standard convolutions. This is followed by a pointwise convolution (1x1 convolution) that creates new features by computing linear combinations of the depthwise convolution outputs.

This factorization dramatically reduces the number of parameters and computational operations required, making the network more efficient without significantly compromising its enhancement capabilities. The reduction in computational complexity is particularly beneficial for real world applications where processing resources may be limited. The use of depthwise separable convolutions throughout the network's architecture enables the development of a lightweight yet effective solution for low-light image enhancement.

The theoretical framework thus combines classical image processing theory with modern deep learning techniques, creating a robust and efficient architecture for low-light image enhancement. This integration of Retinex theory, CNNs, and efficient convolution operations provides a solid foundation for addressing the challenges of low-light image enhancement while maintaining computational efficiency.

Depthwise separable convolutions offer a more efficient alternative to standard convolutions. That convolutions consist of two operations, depthwise convolution that applies a single filter per input channel and pointwise convolution that

uses 1×1 convolutions to combine the outputs of the depthwise convolution.

The computational complexity of depthwise separable convolutions is significantly lower than standard convolutions, standard convolution's computatonal complexity is calculated as:

$$Dk \times Dk \times M \times N \times Df \times Df \tag{4}$$

and depthwise separable convolution's computational complexity is calculated as:

$$Dk \times Dk \times M \times Df \times Df + M \times N \times Df \times Df, \tag{5}$$

where, Dk represents kernel size, M represents input channels, N represents output channels and Df represents the feature map size.

Width Multiplier
The width multiplier α is a hyperparameter introduced to scale the number of channels in a neural network, providing flexible control over the model's capacity and computational complexity. By adjusting α, the number of channels in each layer can be proportionally reduced or expanded, effectively modifying the network's size, speed, and memory footprint. For instance, setting $\alpha = 0.5$ reduces the number of channels by half compared to the baseline architecture, significantly decreasing the number of parameters and floating-point operations.

This trade-off allows practitioners to tailor model architectures to meet the constraints of specific deployment environments, such as mobile or real-time systems, where computational resources are limited. Such configurations are particularly advantageous in real time or limited resourced environments, where speed and memory efficiency are critical.

3.4.3 Training Optimizations

Mixed Precision Training
As introduced in paper Mixed Precision Training [34], mixed precision training uses both 16-bit (half precision) and 32-bit (single precision) floating-point representations during training. This reduces memory usage by 50% and accelerates training on modern GPUs while maintaining model accuracy through gradient scaling.

Gradient Scaling
Prevents gradient underflow in mixed precision training by scaling loss values before backpropagation, then restoring the original scale gradients before optimizer updates. [34]

Batch Normalization
Batch Normalization, introduced by Ioffe and Szegedy [17], is a technique designed to address the problem of internal covariate shift during training of deep neural networks. As the distribution of activations changes across mini-batches during training, the inputs to each layer may shift unpredictably, making learning slower and more unstable.

To reduce these slowness and instability, Batch Normalization normalizes the inputs of each mini-batch to have zero mean and unit variance, followed by a learnable scaling and shifting operation. This stabilizes the input distribution throughout the network, allowing for higher learning rates, faster convergence, and improved generalization. It has become a standard component in modern deep learning architectures because of its effectiveness and simplicity.

3.4.4 Architectural Design Patterns

Residual Connections
Residual connections, introduced by He et al. in the ResNet architecture [15], are skip connections that enable the construction and training of very deep neural networks. They address the degradation problem, where deeper models begin to perform worse than shallower ones, not due to overfitting or vanishing gradients, but due to optimization difficulties.

Instead of directly learning a mapping $H(x)$, the network learns a residual function $\mathcal{F}(x) := H(x) - x$, and reformulates the original mapping as:

$$y = \mathcal{F}(x) + x, \tag{6}$$

where $\mathcal{F}(x)$ is the residual mapping to be learned and x is the identity input passed through a shortcut connection. This allows the network to learn easier and identity mappings and accelerates convergence, enabling the training of networks with hundreds or even thousands of layers.

Multi-Scale Processing
Uses parallel convolution branches with different kernel sizes, 3×3, 5×5, 7×7, to capture features at multiple scales, then combines outputs for more comprehensive feature representation.

Encoder-Decoder Architecture
A symmetric network structure consisting of two main parts: the encoder, which progressively reduces the spatial dimensions of the input while increasing the feature depth to extract high level representations; and the decoder, which reconstructs the output by gradually increasing spatial resolution, aiming to recover the original input dimensions or generate a desired output.

3.5 Loss Functions and Training Strategies

3.5.1 Reconstruction Losses

L1 Loss (Mean Absolute Error)
The L1 Loss, also known as Mean Absolute Error (MAE), is a fundamental loss function used in image processing and computer vision tasks. It measures the absolute difference between predicted and target images.

$$L1 = (1/N) * \Sigma |I\_pred(i,j) - I\_target(i,j)| \tag{7}$$

where I_pred stands for the predicted/estimated image from the model, I_target is the ground truth/target image, N is the total number of pixels (width $\times$ height $\times$ channels), (i,j) states the pixel coordinates, the summation is over all pixels and L1 is the resulting loss value.

Illumination Smoothness Loss
The Illumination Smoothness Loss is designed to ensure that the estimated illumination map is spatially smooth and natural looking. It prevents the illumination map from having sharp, unrealistic transitions that would create artifacts in the final enhanced image.
Illumination Smoothness Loss encourages spatial smoothness in illumination maps using total variation:

$$L_{\text{smooth}} = |\nabla_x I| + |\nabla_y I|, \tag{8}$$

where, ∇x I is the gradient of image I in the x-direction (horizontal), ∇y I is the gradient of image I in the y-direction (vertical) and L_smooth is the resulting smoothness loss.
Gradient (∇) represents the rate if change of pixel values. ∇x represents how much of pixel values change from left to right (horizontal) while ∇y represents how much pixel values change from top to bottom (vertical).

3.5.2 Adversarial Losses

Generative Adversarial Networks (GANs) consist of two neural networks: a Generator G and a Discriminator D, engaged in a two-player minimax game. The optimizing function that introduced in paper [11], their interaction is:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log(1-D(G(z)))\right] \quad (9)$$

where, $x \sim p_{\text{data}}(x)$: samples drawn from the real data distribution, $z \sim p_z(z)$: samples from the prior noise distribution (e.g., Gaussian or uniform), $D(x)$: the discriminator's predicted probability that x is real, $D(G(z))$: the discriminator's predicted probability that the generated data are real.

Discriminator Loss The discriminator is trained to maximize this, which in other words minimizing the following binary cross-entropy loss:

$$L_D = -\mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log D(x)\right] - \mathbb{E}_{z\sim p_z(z)}\left[\log(1-D(G(z)))\right] \quad (10)$$

This loss encourages the discriminator to assign high confidence to real data and low confidence to generated data.

Generator Loss The generator aims to fool the discriminator by generating samples that are classified as real. In the original form, this corresponds to minimizing:

$$L_G = \mathbb{E}_{z\sim p_z(z)}\left[\log(1-D(G(z)))\right] \quad (11)$$

However, this form may lead to vanishing gradients in the early training stages when D(G(z)) is close to zero.

3.5.3 Consistency Losses

Equal Reflectance Loss

Equal Reflectance Loss makes sure that reflectance components to be similar between low-light and normal-light image pairs by comparing the distance:

$$L_{\text{refl}} = ||R_{\text{low}} - R_{\text{normal}}||, \quad (12)$$

where, R_{low} is the reflectance map extracted from the low-light image using the

decomposition network and R_{normal} is the reflectance map extracted from the normal-light image.

Mutual Illumination Loss

Mutual Illumination Loss ensures illumination consistency across related images by comparing the distance:

$$L_{\text{illum}} = ||I_{\text{low}} - I_{\text{normal}}||, \tag{13}$$

where, I_{low} is the illumination map extracted from the low-light image and I_{normal} is the illumination map extracted from the normal-light image.

3.6 Data Augmentation and Preprocessing

Spatial Augmentations
Random crops, horizontal flips, and rotations (0°, 90°, 180°, 270°) increase dataset diversity and model robustness.

Tensor Preprocessing
Images or frames are normalized to [0,1] range and converted to CHW format (Channels, Height, Width) for efficient GPU processing.

Multi-threaded Data Loading
Parallel data loading with configurable worker threads to prevent I/O bottlenecks during training.

4 Methodology

In this section, we introduce our methodology of our Retinex-based implementation in details. Our approach for video processing is that we adopt a frame-by-frame approach where videos are decomposed into their frames, with each frame processed through our enhancement model as if it were an image. This methodology ensures that every frame receives the same level of enhancement quality as individual images, maintaining consistency across the entire video sequence. After processing, the enhanced frames are reassembled to reconstruct the complete enhanced video, preserving the original temporal structure while significantly improving visual quality. This approach leverages the proven effectiveness of our image enhancement architecture for video applications without requiring specialized video processing components or temporal modeling.

4.1 Thoretical Framework

4.1.1 Our Framework

Our aim for our model is to enhance low-light visual data in real time. The input to the system is a low-light RGB image or video frame, represented as a 3-channel tensor with pixel intensities in the range [0,1]. The output is an enhanced RGB image or frame of the same resolution, with improved brightness, contrast, and detail preservation. The system operates under a supervised learning framework, where paired low-light and high-light images are used during training. The model is trained to minimize reconstruction and consistency losses that ensure visual quality and accuracy based on Retinex decomposition.

Our work is based on the Retinex theory, RetinexNet and KinD++, which we have already introduced and explained in Section 3.2 and Section 3.3.

Our proposed framework extends the original approach with several key innovations that significantly enhance computational efficiency while maintaining theoretical consistency. The algorithm introduces a width multiplier parameter ($\alpha \in [0.5, 1.0]$) that enables scalable deployment across different computational budgets, allowing the network to adapt from mobile devices to high-performance systems. The decomposition network maintains the same three-stage structure but incorporates MobileNet [16] style optimizations throughout. The encoder begins with a standard convolutional layer ($3 \to 32\alpha$) followed by three depthwise separable convolution layers that progressively increase channel depth ($32\alpha \to 64\alpha \to 128\alpha \to 128\alpha$). Depthwise separable convolutions significantly reduce computational complexity compared to standard convolutions,

while maintaining feature extraction capability. These convolutions decompose the standard convolution operation into two steps: a depthwise convolution that processes each input channel independently, followed by a pointwise convolution that combines the results.

Then we adopt and extend several loss functions from KinD++ to ensure decomposition consistency such as the Equal Reflectance Loss that ensures reflectance consistency between low-light and high-light versions of the same scene, Mutual Illumination Loss that maintains illumination consistency across different lighting conditions, Mutual Input Loss that preserves consistent input differences across different lightning coniditions and Decomposition Consistency Loss that enforces the physical constraint that reconstructed images should match the original inputs. In Section 6, we present a comparison of the effects of using each loss function individually and in combination. This helps us understand how each component contributes to the final result and determine which combination works best for enhancing low-light images.

We use the depthwise separable convolutions from MobileNet [16] Architecture to make our model faster by reducing the computational complexity. We also employ width multiplier from MobileNet architecture for ensuring scalable deployment. In Section 6, our benchmark results with different width multipliers can be seen.

When enhancing a low-light video, we take each frame of the video, enhance each frame with our model and then reassemble the video with the enhanced frames. Video Enhancement Pipeline is illustrated below in Figure 3.

Figure 3: The input video is split into frames, each frame is enhanced by our model, and then the enhanced frames are reassembled into the output video.

The dual decoders follow a similar optimization strategy, employing depthwise separable convolutions for the majority of layers ($128\alpha \rightarrow 64\alpha \rightarrow 32\alpha$) before transitioning to standard convolutions for the final output layers. The reflectance decoder includes an additional intermediate layer ($32\alpha \rightarrow 16\alpha$) before the final 1×1 convolution that produces the 3-channel reflectance map. The illumination

decoder follows an identical structure but outputs a single-channel illumination map. Both decoders maintain sigmoid activation to ensure output constraints.

A significant departure from the original algorithm is the introduction of dual enhancement networks that process both illumination and reflectance components independently. The illumination enhancement network employs a delta-based approach, where the network learns to predict an enhancement delta (ΔI) that is added to the original illumination before applying sigmoid activation:

$$I\_enhanced = \sigma(I + \Delta I). \tag{14}$$

This residual learning strategy improves training stability and convergence compared to direct prediction. The network consists of an initial convolutional layer ($1 \to 32\alpha$) followed by five residual blocks, each containing depthwise separable convolutions with batch normalization and ReLU activation.

The reflectance enhancement network is one of our contributions, as the original RetinexNet did not enhance the reflectance component. This network processes the 3-channel reflectance through an initial convolutional layer ($3 \to 32\alpha$) followed by five multi-scale processing blocks. Each multi-scale block employs parallel depthwise separable convolutions with different kernel sizes ($3\times3, 5\times5, 7\times7$) to capture features at multiple scales, enhancing the network's ability to preserve fine details and textures. The outputs of these parallel branches are averaged and combined with the input through a residual connection, followed by batch normalization and ReLU activation.

The reconstruction stage performs element-wise multiplication between both enhanced components:

$$EnhancedImage = EnhancedIllumination \otimes EnhancedReflectance. \tag{15}$$

This approach provides greater control over the enhancement process and allows for more sophisticated enhancement strategies that can address both lighting and detail preservation simultaneously.

The training process uses loss functions that extends beyond the original formulation to ensure both enhancement quality and theoretical consistency. The primary reconstruction loss measures the ℓ_1 distance between the enhanced output and the high-quality ground truth. The smoothness loss penalizes spatial gradients in the illumination map to maintain natural lighting transitions. The equal reflectance loss enforces consistency between reflectance components extracted from low-light and high-light versions of the same scene, preventing unrealistic reflectance modifications. The construction loss ensures consistency

between the enhanced components and the final reconstructed output. The total loss combines these components with carefully tuned weights to balance the various objectives. We also use standard loss functions that are commonly used in image enhancement, Reconstruction Loss that ensures the enhanced image macthes the ground truth, and Smoothness Loss that regularizes the smoothness of the illumination maps.

The algorithm's efficiency improvements are substantial, with parameter counts ranging from approximately 12K (for $\alpha = 0.25$) to 100K (for $\alpha = 1.0$) depending on the width multiplier α, compared to the original RetinexNet's $\sim$200K parameters. This reduction is to enable the real-time performance (greater than 30 FPS on GPU) while maintaining or even improving enhancement quality. The combination of depthwise separable convolutions, width multiplier scaling, and enhanced loss functions creates a strong framework approach for low-light enhancement that is both theoretically sound and practically efficient.

Below in Figure 4, our proposed framework can be seen.

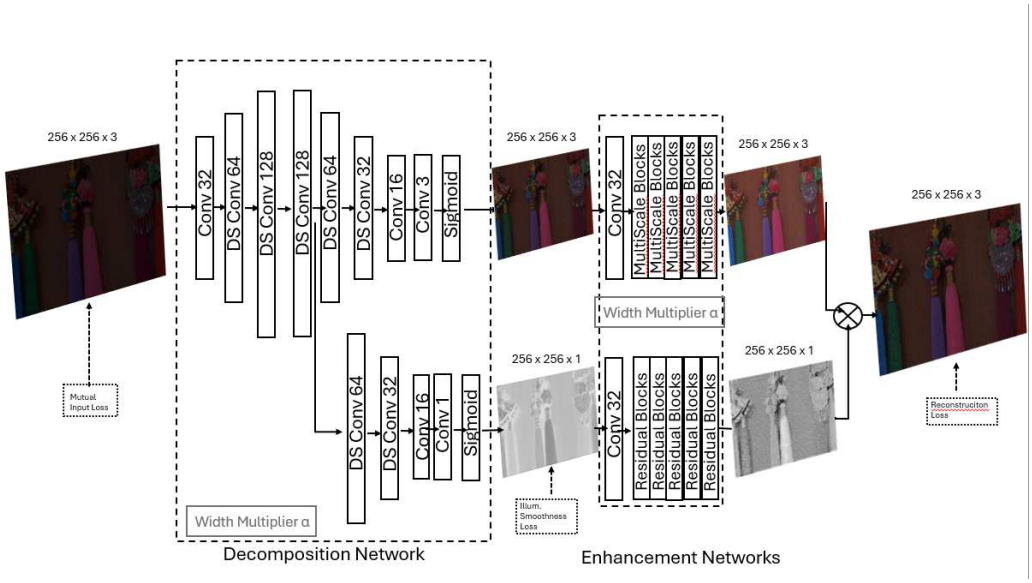


Figure 4: Our proposed framework. The model is composed of a Decomposition Network and two parallel Enhancement Networks. The Decomposition Network separates the input image into reflectance and illumination components using depthwise separable convolutions, a width multiplier α, and sigmoid activations. The reflectance and illumination maps are then independently enhanced using distinct multi-scale and residual block-based networks. The final enhanced image is reconstructed through element-wise multiplication of the enhanced components. The training is guided by mutual input loss, illumination smoothness loss, and reconstruction loss to ensure detail preservation and perceptual quality under low-light conditions.

4.1.2 Network Structure

Our proposed lightweight RetinexNet architecture consists of three main components, Decomposition Network, Illumination Enhancement Network and Reflectance Enhancement Network.

Decomposition Network

Our Decomposition Network takes low-light image with three channels as input and gives an illumination map with one channel and a reflectance map with three channels as output. The architecture of this network is an encoder-decoder structure with depthwise separable convolutions. Key features we have in this network are, a width multiplier for flexible model size, mixed precision training support, batch normalization and ReLU activation function.

Illumination Enhancement Network
Our Illumination Enhancement Network takes Illumination map from decomposition network as the input and gives the enhanced illumination map as the output. The architecture of this network is residual blocks with depthwise separable convolutions. Key features we have in this network are, multiple enhancement blocks, skip connections for better gradient flow and adaptive illumination adjustment.

Reflectance Enhancement Network
Our Reflectance Enhancement Network takes Reflectance map from decomposition network as the input and gives the enhanced reflectance map as the output. The architecture of this network is multi-scale processing blocks. Key features we have in this network are, parallel processing of different scales, feature fusion for better detail preservation and color consistency maintenance.

4.1.3 Convolutional Neural Networks

The enhancement networks employ specialized convolutional architectures tailored to their respective tasks, each designed to address the unique challenges of their specific enhancement goals.

Illumination Enhancement Network focuses on adjusting lighting conditions while preserving spatial coherence. It employs a delta-based enhancement strategy where the network learns to predict enhancement adjustments (ΔI) rather than absolute values. This approach is implemented through residual blocks with depthwise separable convolutions, which are particularly effective for illumination enhancement because they can capture both local lighting variations and global lighting trends. The residual learning strategy

$$(I\_enhanced = \sigma(I + \Delta I)) \tag{16}$$

ensures that the network can make both subtle and significant lighting adjustments while maintaining training stability.

Reflectance Enhancement Network is designed to preserve and enhance material properties, focusing on detail preservation and noise reduction. It employs multi-scale processing blocks that use parallel convolutions with different kernel sizes (3×3, 5×5, 7×7) to capture features at multiple spatial scales simultaneously. This multi-scale approach is crucial for reflectance enhancement because it allows the network to preserve fine details (captured by small kernels) while understanding broader material patterns (captured by larger kernels). The parallel processing ensures that information at all scales is preserved and enhanced

appropriately.

Our methodology addresses the limitation we have introduced in section 3.1, by leveraging depthwise separable convolutions, which decompose the standard convolution operation into two efficient steps: a depthwise convolution that processes each input channel independently, followed by a pointwise convolution that combines the results. This optimization reduces computational complexity by approximately 60% while maintaining feature extraction capability, enabling real-time performance without sacrificing enhancement quality.

Our network architecture incorporates modern CNN design principles such as skip connections and multi-scale processing, which are essential for effective image enhancement. Skip connections enable the network to maintain fine details by providing direct pathways for low-level features to reach deeper layers, preventing the loss of important information that often occurs in deep networks. This is particularly important for low-light enhancement, where preserving fine details is crucial for natural-looking results.

Multi-scale processing allows the network to simultaneously consider information at different spatial scales, which is essential for understanding both local details and global image structure. This capability is implemented through the multi-scale blocks in the reflectance enhancement network and the hierarchical structure of the decomposition network. The design ensures that the network can make informed enhancement decisions based on both local context (important for detail preservation) and global context (important for lighting consistency).

The architecture also ensures end-to-end trainability, allowing all components to be optimized jointly for optimal performance. This joint optimization is crucial because the decomposition and enhancement processes are interdependent - the quality of decomposition affects enhancement results, and enhancement quality influences decomposition accuracy. The comprehensive loss function, which includes reconstruction loss, smoothness loss, equal reflectance loss, and construction loss, ensures that all components work together harmoniously to achieve the best possible enhancement results.

This carefully designed architecture represents a significant advancement in low-light image enhancement, combining the theoretical foundations of Retinex theory with modern deep learning techniques to create a robust, efficient, and effective enhancement system.

The learning process is guided by the network's ability to understand local and global image contexts through its hierarchical structure like we explained in section 3.1. Local context helps the network identify fine details and textures in the reflectance component, while global context enables understanding of lighting patterns and spatial relationships in the illumination component. This dual understanding is essential for accurate decomposition and subsequent enhancement.

4.1.4 Loss Functions

The training process employs comprehensive loss functions to make ablation study from the loss functions, used in standard methods but especially used in KinD++ [51] which are highly promising, that extends beyond the original formulation to ensure both enhancement quality and theoretical consistency. The primary reconstruction loss measures the ℓ_1 distance between the enhanced output and the high-quality ground truth. The smoothness loss penalizes spatial gradients in the illumination map to maintain natural lighting transitions. The equal reflectance loss enforces consistency between reflectance components extracted from low-light and high-light versions of the same scene, preventing unrealistic reflectance modifications. The construction loss ensures consistency between the enhanced components and the final reconstructed output. The total loss combines these components with carefully tuned weights ($\lambda_{\text{smooth}} = 0.01$, $\lambda_{\text{equal}} = 0.009$, $\lambda_{\text{construction}} = 0.1$) to balance the various objectives.

We experimented with different loss function combinations to understand their individual contributions to ensure we use the best one/ones for both enhancement quality and adherence to Retinex theory principles. We talk in details about the results and which loss functions we used in final in the section 6. The total loss function combines several specialized terms, each addressing specific aspects of the low-light enhancement task.

Reconstruction Loss
The primary supervision signal is provided by the L1 reconstruction loss, which measures the pixel-wise difference between the enhanced image and the ground truth:

$$L_{recon} = ||I_{enhanced} - I_{high}||_1 \tag{17}$$

where $I_{enhanced}$ represents the network output and I_{high} is the corresponding high-quality ground truth image.

Decomposition Consistency Loss
To ensure adherence to Retinex theory, we enforce that the decomposed com-

ponents can accurately reconstruct both the input and target images:

$$L_{decomp} = ||R \otimes I - I_{low}||_1 + ||R \otimes I_{enhanced} - I_{high}||_1 \tag{18}$$

where R represents reflectance, I denotes illumination, $I_{enhanced}$ is the enhanced illumination, and $\otimes$ indicates element-wise multiplication.

Equal Reflectance Loss
This constraint enforces the assumption that reflectance properties remain consistent across different lighting conditions:

$$L_{equal\_refl} = ||R_{low} - R_{high}||_1 \tag{19}$$

where R_{low} and R_{high} are the reflectance components extracted from low-light and high-light image pairs, respectively.

Mutual Illumination Loss
Measures consistency between input image pairs to preserve original image relationships, we implement:

$$L_{mutual\_illum} = ||I_{low} - I_{high}||_1 \tag{20}$$

This term helps the network learn proper illumination relationships across the dataset.

Smoothness Regularization
Spatial smoothness in illumination maps is enforced through gradient-based regularization:

$$L_{smooth} = \sum_{x,y} (|\nabla_x I(x,y)| + |\nabla_y I(x,y)|) \tag{21}$$

This prevents artifacts and maintains natural illumination transitions.

Adversarial Loss (Optional)
When GAN training is enabled, an additional adversarial loss using binary cross-entropy improves perceptual quality:

$$L_{adv} = E[\log D(I_{high})] + E[\log(1 - D(I_{enhanced}))] \tag{22}$$

where D represents the discriminator network.

Total Loss Formulation
The final loss function combines all terms with carefully tuned weights:

$$L_{total} = \lambda_1 L_{recon} + \lambda_2 L_{smooth} + \lambda_3 L_{equal\_refl} + \lambda_4 L_{mutual\_illum} + \lambda_5 L_{adv} \tag{23}$$

where the weights are set to $\lambda_1 = 1.0$, $\lambda_2 = 0.01$, $\lambda_3 = 0.01$, $\lambda_4 = 0.01$, and $\lambda_5 = 0.001$ (when GAN training is enabled).

4.1.5 Width Multiplier

The width multiplier implementation introduces a flexible scaling mechanism that allows for dynamic adjustment of the network's capacity and computational requirements. This implementation involves multiplying the number of channels in each layer by a scaling factor α, where $0 < \alpha \leq 1$. The width multiplier affects all layers uniformly, including both the depthwise and pointwise convolutions in the depthwise separable convolution blocks. The implementation maintains the architectural integrity while reducing the model's complexity proportionally. The width multiplier is implemented in a way that preserves the relative proportions of feature channels across different layers, ensuring that the network maintains its ability to learn hierarchical features effectively. The implementation includes proper handling of channel dimensions to ensure compatibility with subsequent layers and operations, particularly in cases where the number of channels needs to be rounded to the nearest integer.

4.1.6 Mixed Precision Training

The implementation of mixed precision training in our architecture represents a sophisticated approach to optimizing both training speed and memory efficiency. This implementation utilizes a combination of FP16 (16-bit floating point) and FP32 (32-bit floating point) precision levels, where the majority of the network operations are performed in FP16 while maintaining certain critical operations in FP32. The implementation includes a dynamic loss scaling mechanism that automatically adjusts the scaling factor to prevent underflow in the FP16 computations. The forward pass is primarily executed in FP16, with careful handling of the activation functions to prevent numerical instability. The backward pass and weight updates are performed in FP32 to maintain training stability and accuracy. The implementation includes proper handling of batch normalization layers, where the running statistics are maintained in FP32 to ensure numerical stability. The mixed precision training implementation also incorporates gradient clipping and proper initialization strategies to prevent training divergence. The implementation is designed to be compatible with modern hardware accelerators that support mixed precision operations, particularly NVIDIA GPUs with Tensor Cores. This approach results in significant memory savings and computational speedup while maintaining the model's accuracy and training stability.

4.2 Implementation Details

The training process incorporates several optimization strategies to ensure effective learning and model convergence.

4.2.1 Data Preprocessing and Augmentation

Input images are pre processed by resizing to 256×256 pixels and normalizing to the [0,1] range. Data augmentation techniques enhance training diversity through random cropping to the target resolution, horizontal flipping with 50% probability and random rotation among {0°, 90°, 180°, 270°}

4.2.2 Optimization Configuration

The network is optimized using the Adam optimizer [21] with the following hyperparameters learning rate $\alpha = 0.0002$, momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, batch size = 16 and training epochs = 100.

4.2.3 Learning Rate Scheduling

A ReduceLROnPlateau scheduler monitors validation loss and reduces the learning rate by a factor of 0.5 when the loss plateaus for 5 consecutive epochs. This adaptive scheduling enables fine-tuning in later training stages and helps achieve better convergence.

4.2.4 Mixed Precision Training

To improve computational efficiency and reduce memory usage, we employ PyTorch's Automatic Mixed Precision (AMP) training. This technique uses FP16 precision for forward passes and FP32 precision for gradient computation, maintaining numerical stability while accelerating training.

4.2.5 Model Selection and Validation

Training progress is monitored through comprehensive validation after each epoch. The model with the lowest validation loss is selected as the final model, ensuring optimal generalization to unseen data. Key metrics tracked include reconstruction loss, individual component losses (smoothness, equal reflectance, mutual illumination), adversarial loss (when GAN is in use).

The implementation ensures reproducibility through fixed random seeds across all components, automatic device selection (GPU when available, CPU fallback), regular checkpoint saving for training resumption, comprehensive logging and visualization of training metrics.

4.2.6 Retinex Implementation

The RetinexNet class we implemented represents a comprehensive neural network architecture designed for low-light image enhancement, based on the foundational Retinex theory [23]. This architecture is implemented as a PyTorch module, which enables effortless integration with PyTorch's deep learning framework and automatic differentiation system.

The network's constructor accepts two crucial parameters that define its architecture characteristics. The width multiplier parameter, defaulting to 1.0, serves as a network-wide multiplier that controls the model's capacity by adjusting the number of channels throughout the network. This multiplier enables flexible scaling of the model's size and computational requirements. The number of enhance blocks parameter, defaulting to 5, determines the depth of the enhancement networks by specifying the number of enhancement blocks to be used, allowing for adjustable computational complexity and enhancement capability.

The implementation of the Retinex theory in our architecture follows a systematic approach that decomposes the input image into its fundamental components. The decomposition network employs a series of convolutional layers with carefully designed kernel sizes and activation functions to separate the illumination and reflectance components. The decomposition network employs four convolutional layers with 3×3 kernels for spatial feature extraction. The first layer uses standard convolution, followed by three depthwise separable convolution layers to efficiently increase channel dimensions. Each convolutional layer is succeeded by a ReLU activation function, which introduces non-linearity and ensures the non-negativity of the output, a crucial requirement for both illumination and reflectance maps.

4.2.7 Depthwise Separable Convolution Implementation

The implementation of depthwise separable convolutions in our architecture represents a significant optimization in terms of computational efficiency and model complexity. This implementation decomposes the standard convolution operation into two distinct steps: depthwise convolution and pointwise convolution. The depthwise convolution applies a single convolutional filter per input channel, effectively performing spatial filtering while maintaining channel independence. This is followed by a pointwise convolution, which employs 1×1 kernels to combine the features across channels. The mathematical formulation of this process can be expressed as follows: for an input tensor of size (H, W, C_in) and an

output tensor of size (H, W, C_out), the depthwise separable convolution reduces the number of parameters from (K $\times$ K $\times$ C_in $\times$ C_out) to (K $\times$ K $\times$ C_in + C_in $\times$ C_out), where K represents the kernel size. This reduction in parameters leads to a significant decrease in computational complexity while maintaining the network's representational capacity. The implementation relies on PyTorch's default weight initialization, which provides stable training performance. Additionally, batch normalization layers are incorporated after each convolution operation to normalize the feature maps and improve training stability. The implementation also includes proper padding strategies to maintain spatial dimensions and ensure consistent feature map sizes throughout the network.

5 Data and Preprocessing

In this research, we approached real-time video enhancement as a frame-by-frame image processing task. Accordingly, we trained and evaluated our model using a combination of datasets mostly having low-light and normal-light image pairs.

5.1 Datasets

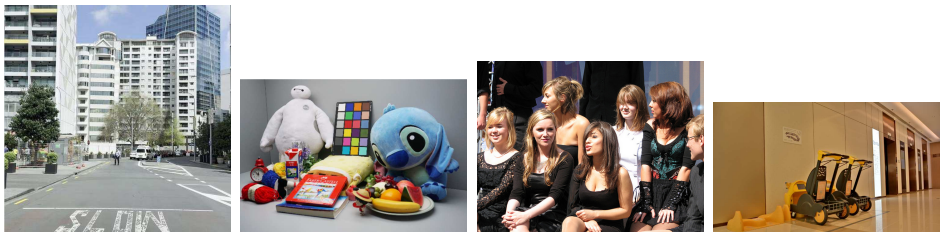
Our primary dataset is the LoLI-Street Dataset [18], which provides 30,000 paired low-light and high-light images for training and validation. Its large size and diverse scenes make it well-suited to our approach of treating video frames as individual images.

To increase the diversity and robustness of our training data, we incorporated several additional datasets. The LOL dataset, introduced in "Deep Retinex Decomposition for Low-Light Enhancement" [47], contributed 500 image pairs, with 485 used for training and 15 for testing. This dataset primarily contains indoor scenes with natural noise from the photo capture process, and all images are standardized to a resolution of 400×600 pixels. We also included the dataset introduced at the NTIRE 2024 Low Light Enhancement Challenge [29], which provides 438 training image pairs and 46 test images, featuring both indoor and outdoor scenes.

To further expand our dataset, we used the Flickr30k dataset introduced in the paper "Flickr30k Entities: Collecting Region-to-Phrase Correspondences for Richer Image-to-Sentence Models" [35]. This addition significantly increased the size and diversity of our training data.

In total, our dataset consisted of approximately 52,000 image pairs for training, 7,000 for validation, and 15 low-light test images. While we had test data from multiple datasets, we chose to perform our evaluation on the LOL test set to enable comparison with baseline models discussed in our literature review.

Below in Figure 5 references of images can be found from each dataset.

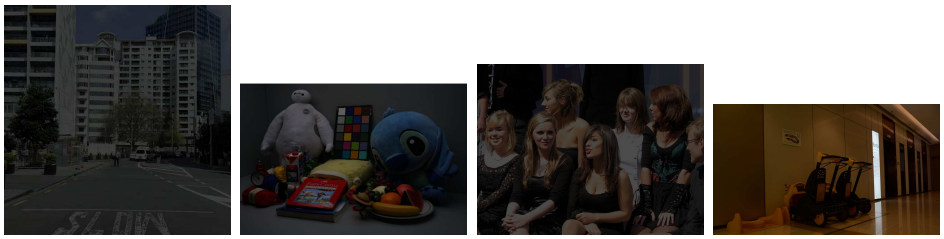


(a) A high-light image example from the LoLI-Street Dataset

(b) A high-light image example from the LoL Dataset

(c) A high-light image example from the Flickr30k Dataset

(d) A high-light image example from the NTIRE Dataset



(e) The low-light pair of the image above of LoLI-Street Dataset

(f) The low-light pair of the image above of LoL Dataset

(g) The low-light pair of the image above of Flickr30k Dataset

(h) The low-light pair of the image above of NTIRE Dataset

Figure 5: Some sample pairs from the LoLI-Street, LoL, NTIRE and Flickr30k datasets

5.2 Preprocessing

All images were resized to a uniform resolution of 256×256 pixels and normalized to a [0,1] range by dividing pixel values by 255. This ensured consistent input dimensions and value ranges across datasets, which is essential for stable training and convergence.

For training data, we applied data augmentation techniques including random cropping, horizontal flipping, and random rotations (0°, 90°, 180°, and 270°) to increase model robustness and prevent overfitting. Validation and test data were only resized to preserve evaluation consistency.

We used batch processing with a batch size of 16, and enabled parallel data loading with 4 worker threads. The $pin\_memory = True$ option in PyTorch was used to speed up data transfer to the GPU. Error handling was integrated into the pipeline to automatically detect and skip corrupted images.

This preprocessing pipeline, by combining standardization, data augmentation, efficient loading, and error handling, provided a solid foundation for training our

real time low-light enhancement model capable of generalizing across diverse lighting conditions and image types.

6 Experiments & Results

6.1 Experimental Setup

To ensure the reproducibility and validity of our proposed approach, this section outlines the experimental setup used throughout the study. The setup includes a detailed description of the hardware and software environments, as well as the baseline methods.

6.1.1 Hardware Configuration

The computational experiments conducted in this study were carried out using the gpu-long partition of the ALICE high-performance computing (HPC) cluster at Leiden University. This partition is specifically designed for GPU-accelerated tasks that require extended runtime, making it highly suitable for deep learning applications such as our project of low-light video enhancement.

The gpu-long partition is equipped with NVIDIA A100 graphics processing units (GPUs), each offering 40 GB of memory. These GPUs provide the necessary parallel processing power and memory capacity to handle complex neural networks and large-scale video data efficiently.

The nodes in this partition are also provisioned with up to 512 GB of RAM, enabling the handling of memory-intensive tasks, including the loading and processing of high-resolution video sequences. Data storage and retrieval are supported by high-speed shared storage systems that ensure quick access to training data and intermediate results.

The availability of such hardware resources was critical to the success of this research, as they allowed for both the training of deep learning models and the evaluation of their performance in a timely and efficient manner.

6.1.2 Software Environment

All experiments and implementations in this study were carried out using Python version 3.9.21, managed within a Conda virtual environment. This isolated environment ensured reproducibility and consistency across different computational nodes on the high-performance computing infrastructure. The software environment was tailored specifically for deep learning-based image and video enhancement tasks, incorporating both foundational and task-specific libraries.

The core of the deep learning framework was built using PyTorch 2.6.0, which was compiled with CUDA 12.4 support and Torchvision 0.21.0, enabling flexible

model development and efficient GPU-based training. Numerical computations were handled with NumPy 2.0.2 and SciPy 1.13.1, while data manipulation and preprocessing were facilitated using Pandas 2.2.3 and Scikit-Image 0.24.0.

To support model evaluation and performance tracking, several visualization and monitoring tools were integrated. These included Matplotlib 3.9.4 for plotting, TensorBoard 2.19.0 for real-time tracking of training metrics, and TQDM 4.67.1 for progress visualization during iterative operations.

Image processing and augmentation were supported by OpenCV 4.11.0.86 and Pillow 11.1.0, both of which enabled efficient handling of input data and model outputs. Perceptual quality metrics that used in evaluation were calculated using the LPIPS 0.1.4 and PyIQA 0.1.13 libraries, which are widely used in image enhancement research to determine visual accuracy beyond pixel-level comparisons.

This environment configuration, enabled through Conda, provided a modular platform for the development, training, and evaluation of low-light enhancement models, ensuring compatibility, reproducibility, and scalability throughout the research process.

6.2 Evaluation Methodology

The evaluation of our RetinexNet based implementation employs a comprehensive approach that combines both quantitative metrics and qualitative visual assessment. This evaluation allows us to thoroughly assess the model's performance in terms of both objective measurements and subjective visual quality. The evaluation is performed on the LOL dataset to ensure that performance comparison is enabled with other methods. The evaluation process is designed to measure not only the quality of the enhancement, but also the preservation of important image characteristics such as color accuracy, structural details and natural appearance.

6.2.1 Quantitative Metrics

The quantitative evaluation of our Retinex based enhancement network implementation employs a comprehensive set of metrics to evaluate various aspects of image quality and enhancement performance. The evaluation includes both traditional image quality metrics and advanced perceptual metrics. Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) are used to measure the pixel-wise accuracy and structural similarity between enhanced and ground truth images. The Mean Squared Error (MSE) provides a direct measure of the reconstruction error. For perceptual quality assessment, we employ the Learned

Perceptual Image Patch Similarity (LPIPS) metric, which uses deep learning to measure perceptual similarity. Natural Image Quality Evaluator (NIQE) and the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) are used to assess the naturalness and quality of the enhanced images without requiring reference images. Color specific metrics include Color Difference (DeltaE) in color space and color saturation measurements. Contrast metrics such as RMS contrast, Michelson contrast, and Weber contrast are used to evaluate the enhancement of image details.

In addition to image evaluation, we ran video-based tests to assess the real-time performance of the enhancement network. Frames per second (FPS) were measured to evaluate the model's inference speed and suitability for live or streaming applications.

All these metrics are computed using standardized implementations and validated against established benchmarks to ensure reliable assessment of the model's performance.

6.2.2 Qualitative Evaluation

The qualitative evaluation of our Retinex based enhancement network implementation focuses on visual assessment of the enhanced images and their components. The evaluation process includes a detailed visual analysis of the decomposition results, including the illumination map, enhanced illumination map, and reflectance map. The visual assessment is conducted through side-by-side comparisons of original and enhanced images, allowing for direct evaluation of the enhancement quality. The evaluation also includes visualization of the intermediate components of the Retinex decomposition, providing insights into how the model separates and enhances different aspects of the image. The qualitative assessment considers multiple aspects of the enhanced images, including naturalness of the enhanced results, preservation of fine details and textures, color accuracy and consistency, absence of artifacts or distortions and overall visual appeal and aesthetic quality.

The qualitative evaluation is performed on a diverse set of test images covering various scenarios, including low-light conditions, mixed lighting situations, complex textures and details, different color temperatures in various scene types (indoor, outdoor, portrait, landscape), three different light conditioned videos and a live stream.

Below in Figure 6 and Figure 7 two examples for the enhanced image can be found.

Figure 6: Qualitative comparison of the original and our network's enhanced image. Left side is the original image, right side is the enhanced image.

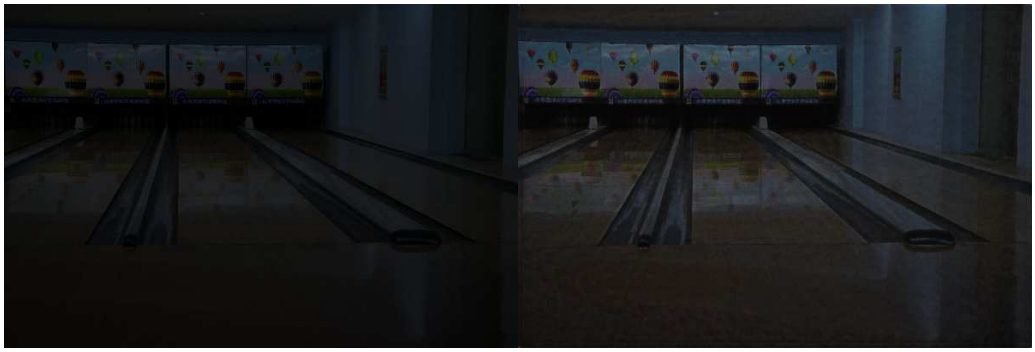


Figure 7: Qualitative comparison of the original and our network's enhanced image. Left side is the original image, right side is the enhanced image.

The components of the images (the original image, the enhanced image, reflactance map, illumination map, enhanced illumination map and original image with the enhanced image) can be seen below in Figure 8 and Figure 9.

Figure 8: Components of the image. From left to right, the original image, the enhanced image, reflactance map, illumination map, enhanced illumination map and original image with the enhanced image.

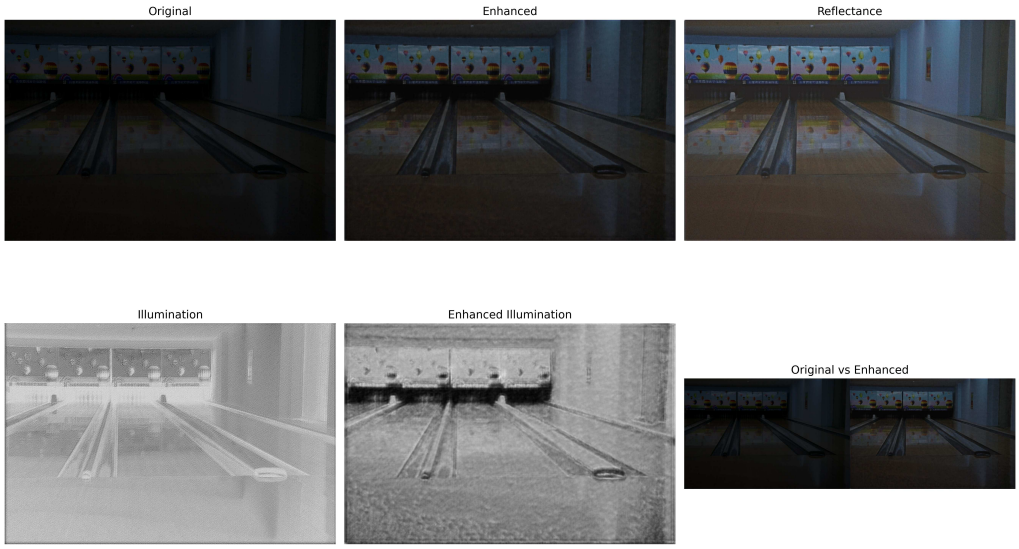


Figure 9: Components of the image. From left to right, the original image, the enhanced image, reflactance map, illumination map, enhanced illumination map and original image with the enhanced image.

6.3 Ablation Study

This section presents the results of our experiments using various loss functions and compares the performance of models trained with and without GAN components.

Below in Table 2 the comparison of loss functions with or without the usage of GANs can be found. Equal Refl. Loss stands for Equal Reflactance Loss. Met. 1 - Met. 8 stand for methods from 1 to 8, in order, with GAN only equal reflactance loss, with GAN equal reflactance and mutual input loss, with GAN equal reflactance and mutual input loss as well as total loss, with GAN only mutual input loss, without GAN only equal reflactance loss, without GAN equal reflactance and mutual input loss, without GAN equal reflactance and mutual input loss as well as total loss and without GAN only mutual input loss.

Table 2: Comparison of different enhancement variants on the LOL Dataset's test set. Each column represents a different configuration. Checkmarks indicate inclusion of GAN and specific loss functions. Best values per metric are in bold. Arrows indicate whether higher (↑) or lower (↓) is better.

Metric / Config	Met. 1	Met. 2	Met. 3	Met. 4	Met. 5	Met. 6	Met. 7	**Met. 8**
GAN	✓	✓	✓	✓	✗	✗	✗	✗
Equal Refl. Loss	✓	✓	✓	✗	✓	✓	✓	✗
Mutual Input Loss	✗	✓	✓	✓	✗	✓	✓	✓
All Losses	✗	✗	✓	✗	✗	✗	✓	✗
PSNR ↑	18.51	18.32	18.51	18.32	21.34	18.32	21.34	**23.29**
SSIM ↑	0.469	0.464	0.469	0.464	0.598	0.464	0.598	**0.718**
NIQE ↓	8.33	**8.28**	8.33	**8.28**	8.39	**8.28**	8.39	8.85
BRISQUE ↓	47.06	45.55	47.06	45.55	49.16	45.55	49.16	**31.71**
LPIPS ↓	0.320	0.320	0.320	0.320	**0.217**	0.320	**0.217**	0.226
Brightness ↑	**3.06**	**3.10**	**3.06**	**3.10**	2.49	**3.10**	2.49	2.05
Contrast ↑	1.94	**2.00**	1.94	**2.00**	1.66	**2.00**	1.66	1.57
FPS ↑	78.98	84.94	33.86	74.06	74.71	76.76	**86.89**	84.94
DeltaE ↓	11.82	11.87	11.82	11.87	10.63	11.87	10.63	**9.60**

Table 2 presents the detailed evaluation of our models trained using various combinations of loss functions, with and without the use of GAN components.

The evaluation metrics used in this study are metrics like PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) quantify the fidelity of reconstruction and the preservation of structural information in comparison to ground truth images, where higher values indicate better performance. Perceptual quality is assessed using NIQE (Natural Image Quality Evaluator), BRISQUE (Blind/Referenceless Image Spatial Quality Evaluator) and LPIPS (Learned Perceptual Image Patch Similarity), all of which are no-reference or deep-feature-based metrics that estimate how natural or visually pleasing an image appears; lower values are better for these metrics. DeltaE measures the perceptual color difference between the enhanced and original images, with lower scores mean more accurate color reproduction. FPS (Frames Per Second) is included to assess runtime efficiency, as higher FPS reflects better suitability for real-time applications such as live video enhancement.

The results show that removing the GAN component and training the model only with Mutual Input (MI) Loss gains the best overall performance across multiple metrics. This model achieves the highest PSNR (23.29) and SSIM (0.718), meaning strong pixel-level accuracy and structural preservation. It also achieves the lowest BRISQUE (31.71) and DeltaE (9.60), shows perceptually pleasing outputs. It also performs competitively in terms of LPIPS (0.226) and FPS (84.94), showing a good balance between visual quality and real-time applicability.

In contrast, models with GAN components tend to score slightly worse in metrics, but marginal gains in brightness and contrast. With the usage of GAN, these aesthetic metrics are improved. However, they suffer from lower PSNR, SSIM, and higher DeltaE.

That is why, our best performed model is the model without GAN and mutual input loss only.

Below in Table 3, the quantitative comparison of our method with other enhancement methods as mentioned in a comparison table in paper [51] on the LOL dataset can be found.

Table 3: Quantitative comparison on the LOL dataset. Best values per metric are in bold.

Method	**PSNR** ↑	**SSIM** ↑	**NIQE** ↓	**DeltaE** ↓
BIMEF [48] (Ying et al. 2017)	13.8753	0.5771	7.6992	21.2383
CRM [49] (Ying et al. 2018)	17.2033	0.6442	8.0182	15.7743
Dong [7] (Dong et al. 2011)	16.7165	0.5824	9.1358	15.6163
LIME [13] (Guo et al. 2017)	16.7586	0.5644	9.1272	14.9474
MF [8] (Fu et al. 2016)	16.9662	0.6422	9.7125	15.5635
RRM [27] (Li et al. 2018)	13.8765	0.6577	5.9416	20.7342
DUPE [43] (Wang et al. 2019)	16.7975	0.5187	8.4736	19.5868
SRIE [9] (Fu et al. 2016)	11.8552	0.4979	7.5349	25.2829
Retinex-Net [47] (Wei et al. 2018)	16.7740	0.5594	9.7289	15.8936
DPE [4] (Chen et al. 2018)	13.1728	0.4787	4.4931	12.2534
NPE [44] (Wang et al. 2013)	16.9697	0.5894	9.1352	15.3318
GLAD [45] (Wang et al. 2018)	19.7182	0.7035	6.7972	12.2776
KinD [52] (Zhang et al. 2019)	20.7261	0.8103	4.1352	9.8632
KinD++ [51] (Zhang et al. 2021)	21.3003	0.8226	**3.8807**	**8.7425**
Our Method	**23.29**	**0.718**	8.85	9.60

Table 3 provides a comparative evaluation of our method against other enhancement methods on the LOL dataset using four key image quality metrics: PSNR, SSIM, NIQE, and DeltaE.

Our method achieves the highest PSNR (23.29), indicating superior pixel-level enhancement performance. The SSIM score (0.718) demonstrates strong structural preservation, and while it is slightly lower than KinD (0.8103) and KinD++ (0.8226), it still outperforms most other methods. In terms of NIQE, our model scores 8.85, which is higher (worse) than some learning-based methods like KinD++ (3.8807), shows room for improvement in perceptual realism. How-

ever, the DeltaE value (9.60) indicates excellent color accuracy, ranking closely behind KinD++ (8.7425) but better than all other methods in the table.

The results show that our approach effectively balances quantitative accuracy and perceptual quality, especially in PSNR and color reproduction.

An important part of our study was testing whether adding a GAN improves results. As seen in Table 2, the GAN-based models did not improve the PSNR or SSIM scores. In fact, they sometimes performed worse and introduced unwanted artifacts. We also found that GAN training made the process less stable. For this reason, we decided not to use a GAN in our final model, and we believe this choice helped us get more reliable and consistent results.

Although our method does not outperform all other methods across every metric, it achieves superior results on some metrics and shows promising performance on others. As a result of these aspects, we are very satisfied with our results. Our model achieves strong performance in terms of image quality, color accuracy, and speed. It improves over previous methods in key areas while remaining lightweight and easy to deploy. We believe this makes it a useful solution for real-time low-light image and video enhancement in real-world applications.

6.4 Benchmark Results

Below in Figure 10 the benchmark results can be seen.

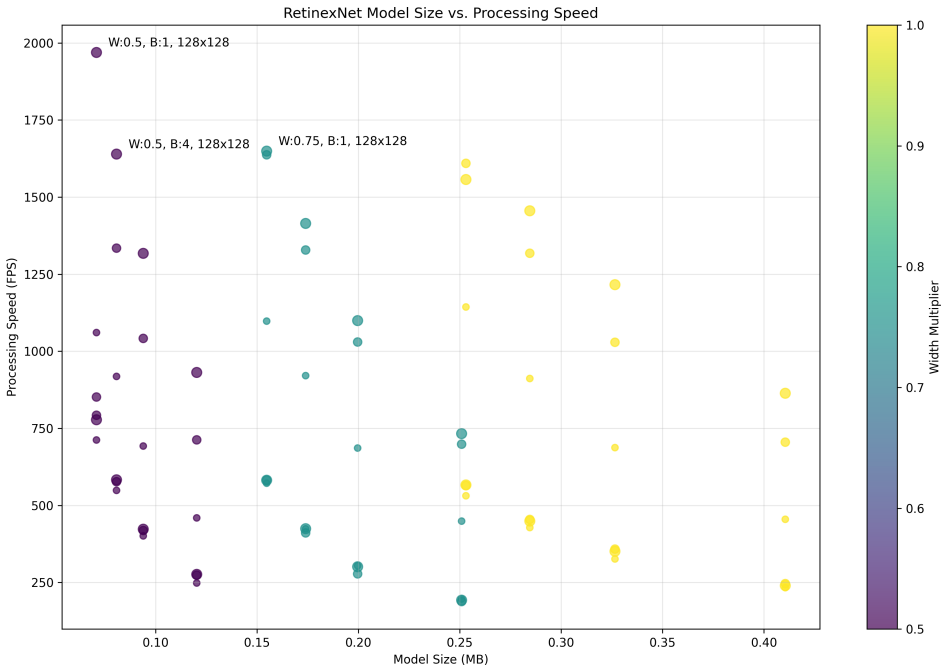


Figure 10: Benchmark results showing the impact of width multiplier α, input resolution, and batch size on inference speed (FPS), model size, and parameter count. As α increases, models become larger and slower but potentially more expressive, while smaller α values offer faster performance with reduced model complexity making them suitable for limited resource or real time applications.

Figure 10 presents a benchmark analysis highlighting how different configurations of our model perform in terms of processing speed (measured in frames per second, FPS) relative to their model size (in megabytes). The color gradient in the plot represents varying values of the width multiplier α, which controls the network's complexity by scaling the number of channels in the model.

From the graph, we can see a clear trade-off between model size and inference speed. Models with smaller α values (e.g., $\alpha = 0.5$, shown in darker colors) are lightweight and offer significantly higher FPS, making them ideal for real-time or resource-constrained applications. These models are especially fast when used with smaller input resolutions or larger batch sizes, as indicated by the annotation (e.g., B=4, 128×128).

On the other hand, models with higher α values (e.g., $\alpha = 1.0$, shown in yellow) are larger in size and slower in terms of FPS, but they are likely to be more expressive and capable of delivering better enhancement quality. These models may be more suitable when computational resources are not a major constraint and quality is the primary goal.

This benchmark shows how tuning α provides flexibility to balance performance and efficiency. Depending on the application needs, whether it's low-latency deployment on mobile devices or high-quality processing on powerful machines, users can select an appropriate model configuration.

7 Conclusion

This thesis presented a lightweight and real-time capable approach to low-light image and video enhancement by integrating Retinex theory [23] with modern efficient deep learning components. Building on the KinD++ [51] architecture, we introduced MobileNet [16] style optimizations, including depthwise separable convolutions and channel width multipliers, which substantially reduce model complexity and computational cost without sacrificing perceptual quality.

We evaluated our method across multiple benchmark datasets using both quantitative metrics (PSNR, SSIM, LPIPS) and qualitative visual analysis. The results demonstrate that our approach maintains competitive enhancement performance while achieving real-time processing speeds. These results shows that our architectural modifications, particularly the dual enhancement networks and efficient convolution strategies, offer a better trade-off between performance and efficiency.

Our answers to our research questions while we are ending our paper are below.

Main Research Question: How can we design computationally efficient network architectures specifically optimized for low-light video enhancement that maintain acceptable visual quality while achieving real-time performance?

To answer this, we designed a lightweight enhancement model based on Retinex theory, with several MobileNet-style architectural optimizations. By incorporating depthwise separable convolutions, a width multiplier parameter, and a dual-branch structure for reflectance and illumination processing, we were able to reduce model size to under 35K parameters (for $\alpha = 0.5$) while still achieving real-time performance (>80 FPS on GPU). Despite the reduction in complexity, the model maintains strong enhancement performance across several key quality metrics, including PSNR (23.29), SSIM (0.718), and DeltaE (9.60). This confirms that real-time low-light enhancement is achievable without relying on large or computationally expensive networks.

Research Question 2: How does the inclusion or exclusion of GAN based training affect the performance and visual quality of low light video enhancement models?

The inclusion or exclusion of GAN-based training affects both the objective performance and perceptual quality of low-light video enhancement models as shown in Table 2, models trained without GAN components, particularly the

configuration using only Mutual Input (MI) Loss, outperform GAN-based models across most quantitative and perceptual metrics. The NoGAN + MI Only variant achieves the highest PSNR (23.29) and SSIM (0.718) values, indicating superior pixel-wise fidelity and structural similarity to ground truth images. It also yields the lowest BRISQUE (31.71) and DeltaE (9.60) scores, demonstrating enhanced perceptual naturalness and better color accuracy. Additionally, this model maintains high real-time efficiency with FPS of 84.94, making it ideal for live applications.

The models that use GANs tend to focus more on enhancing visual appeal, since they outperformed slightly better in brightness and contrast scores, but generally perform worse in terms of PSNR, SSIM, DeltaE, and perceptual metrics like LPIPS.

Therefore, our study shows for real-time low-light video enhancement tasks where stability, color fidelity, and structural consistency are critical, non-GAN approaches with targeted loss functions are more effective.

Research Question 3: How does our proposed approach compare with existing state-of-the-art methods in terms of quantitative metrics assessment?

Our approach outperforms or matches existing state-of-the-art methods on several quantitative metrics. Specifically, it achieves the highest PSNR and SSIM values on the LOL dataset among the compared methods, indicating superior overall enhancement quality. It also records the lowest DeltaE value, demonstrating effective color recovery. As seen in Table 3, while traditional methods like MF or LIME perform slightly better in some perceptual metrics like NIQE, they often fail to preserve structural consistency or realistic color in complex scenes.

Our model's high frame rate and small memory footprint also give it a practical edge over many deeper and slower state-of-the-art methods.

As a result, our quantitative evaluations show that the proposed method is not only competitive with other existing methods, but in some cases outperforms, existing state-of-the-art solutions.

8 Future Work

While this thesis demonstrates the effectiveness of combining Retinex-based decomposition with loss functions from KinD++ [51] lightweight MobileNet [16] style components, several aspects remain open for future exploration and development.

Temporal Enhancement

One limitation of our current architecture is that it enhances each frame independently without exploiting temporal information. Extending the model to process short video clips using temporal consistency mechanisms like 3D convolutions or recurrent architectures like ConvLSTM [39] could improve stability and reduce flickering artifacts in live video streams.

Adaptation for Edge Devices

Although our architecture already incorporates MobileNet-style depthwise separable convolutions to reduce computational complexity, additional optimization is needed to support real-time enhancement on devices such as smartphones and embedded systems. Future work could explore methods to make the model even faster and smaller for running easily on mobile phones or small devices. For instance, techniques like reducing the number of using smaller numbers or automatically designing lighter versions of the model could help. These improvements can lower the time it takes to run the model, use less memory, and save battery, while still keeping the image quality high.

Real-World Dataset Collection and Generalization

While our model is trained on a diverse combination of existing datasets we talked about in Section 5 and evaluated on the LOL dataset, many existing datasets consist of synthetically darkened or controlled-scene imagery. A valuable direction would be to collect a real-world low-light video dataset featuring diverse lighting conditions, motion blur, and noise patterns. This could be modeled on datasets like the See-in-the-Dark (SID) dataset [3], which focuses on extreme low-light conditions. Additionally, testing the model on challenging tasks such as night-time driving or underwater video would allow us to assess generalizability under complex real-world conditions.

References

[1] Tunç Ozan Aydin, Nikolce Stefanoski, Simone Croci, Markus Gross, and Aljoscha Smolic. Temporally coherent local tone mapping of hdr video. *ACM Transactions on Graphics (TOG)*, 33(6):1–13, 2014.

[2] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5972–5981, 2022.

[3] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3291–3300, 2018.

[4] Yu-Sheng Chen, Yu-Ching Wang, Man-Hsin Kao, and Yung-Yu Chuang. Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6306–6314, 2018.

[5] Jiongyu Dai, Qiang Li, Haining Wang, and Lingjia Liu. Understanding images of surveillance devices in the wild. *Knowledge-Based Systems*, 284:111226, 2024.

[6] Xianshu Ding, Hang Lei, and Yunbo Rao. Sparse codes fusion for context enhancement of night video surveillance. *Multimedia Tools and Applications*, 75:11221–11239, 2016.

[7] Xuan Dong, Yi Pang, and Jiangtao Wen. Fast efficient algorithm for enhancement of low lighting video. In *ACM SIGGRApH 2010 posters*, pages 1–1. 2010.

[8] Xueyang Fu, Delu Zeng, Yue Huang, Yinghao Liao, Xinghao Ding, and John Paisley. A fusion-based enhancing method for weakly illuminated images. *Signal processing*, 129:82–96, 2016.

[9] Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping Zhang, and Xinghao Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2782–2790, 2016.

[10] Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009.

[11] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[12] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong. Zero-reference deep curve estimation for low-light image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1780–1789, 2020.

[13] Xiaojie Guo, Yu Li, and Haibin Ling. Lime: Low-light image enhancement via illumination map estimation. *IEEE Transactions on image processing*, 26(2):982–993, 2016.

[14] Jiang Hai, Zhu Xuan, Ren Yang, Yutong Hao, Fengzhu Zou, Fang Lin, and Songchen Han. R2rnet: Low-light image enhancement via real-low to real-normal network. *Journal of Visual Communication and Image Representation*, 90:103712, 2023.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[18] Md Tanvir Islam, Inzamamul Alam, Simon S. Woo, Saeed Anwar, IK Hyun Lee, and Khan Muhammad. Loli-street: Benchmarking low-light image enhancement and beyond. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 1250–1267, December 2024.

[19] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *IEEE transactions on image processing*, 30:2340–2349, 2021.

[20] Minjae Kim, Dubok Park, David K Han, and Hanseok Ko. A novel approach for denoising and enhancement of extremely low-light video. *IEEE Transactions on Consumer Electronics*, 61(1):72–80, 2015.

[21] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Yuxin Kong, Peng Yang, and Yan Cheng. Adaptive on-device model update for responsive video analytics in adverse environments. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(1):857–873, 2025.

[23] Edwin H. Land. The retinex theory of color vision. *Scientific American*, 237(6):108–128, 1977.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[25] Seungwon Lee, Nahyun Kim, and Joonki Paik. Adaptively partitioned block-based contrast enhancement and its application to low light-level video surveillance. *SpringerPlus*, 4:1–11, 2015.

[26] Chongyi Li, Chunle Guo, and Chen Change Loy. Learning to enhance low-light image via zero-reference deep curve estimation. *IEEE transactions on pattern analysis and machine intelligence*, 44(8):4225–4238, 2021.

[27] Mading Li, Jiaying Liu, Wenhan Yang, Xiaoyan Sun, and Zongming Guo. Structure-revealing low-light image enhancement via robust retinex model. *IEEE transactions on image processing*, 27(6):2828–2841, 2018.

[28] Wenhao Li, Guangyang Wu, Wenyi Wang, Peiran Ren, and Xiaohong Liu. Fastllve: Real-time low-light video enhancement with intensity-aware look-up table. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 8134–8144, 2023.

[29] Xiaoning Liu, Zongwei Wu, Ao Li, Florin-Alexandru Vasluianu, Yulun Zhang, Shuhang Gu, Le Zhang, Ce Zhu, Radu Timofte, Zhi Jin, Hongjun Wu, Chenxi Wang, Haitao Ling, Yuanhao Cai, Hao Bian, Yuxin Zheng, Jing Lin, Alan Yuille, Ben Shao, Jin Guo, Tianli Liu, Mohao Wu, Yixu Feng, Shuo Hou, Haotian Lin, Yu Zhu, Peng Wu, Wei Dong, Jinqiu Sun, Yanning Zhang, Qingsen Yan, Wenbin Zou, Weipeng Yang, Yunxiang Li, Qiaomu Wei, Tian Ye, Sixiang Chen, Zhao Zhang, Suiyi Zhao, Bo Wang, Yan Luo, Zhichao Zuo, Mingshen Wang, Junhu Wang, Yanyan Wei, Xiaopeng Sun, Yu Gao, Jiancheng Huang, Hongming Chen, Xiang Chen, Hui Tang, Yuanbin Chen, Yuanbo Zhou, Xinwei Dai, Xintao Qiu, Wei Deng, Qinquan Gao, Tong Tong, Mingjia Li, Jin Hu, Xinyu He, Xiaojie Guo, Sabarinathan, K Uma, A Sasithradevi, B Sathya Bama, S. Mohamed Mansoor Roomi, V. Srivatsav, Jinjuan Wang, Long Sun, Qiuying Chen, Jiahong

Shao, Yizhi Zhang, Marcos V. Conde, Daniel Feijoo, Juan C. Benito, Alvaro García, Jaeho Lee, Seongwan Kim, Sharif S M A, Nodirkhuja Khujaev, Roman Tsoy, Ali Murtaza, Uswah Khairuddin, Ahmad 'Athif Mohd Faudzi, Sampada Malagi, Amogh Joshi, Nikhil Akalwadi, Chaitra Desai, Ramesh Ashok Tabib, Uma Mudenagudi, Wenyi Lian, Wenjing Lian, Jagadeesh Kalyanshetti, Vijayalaxmi Ashok Aralikatti, Palani Yashaswini, Nitish Upasi, Dikshit Hegde, Ujwala Patil, Sujata C, Xingzhuo Yan, Wei Hao, Minghan Fu, Pooja choksy, Anjali Sarvaiya, Kishor Upla, Kiran Raja, Hailong Yan, Yunkai Zhang, Baiang Li, Jingyi Zhang, and Huan Zheng. Ntire 2024 challenge on low light image enhancement: Methods and results, 2024.

[30] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognition*, 61:650–662, 2017.

[31] Long Ma, Tengyu Ma, Risheng Liu, Xin Fan, and Zhongxuan Luo. Toward fast, flexible, and robust low-light image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5637–5646, 2022.

[32] Dhruv Makwana, Gayatri Deshmukh, Onkar Susladkar, Sparsh Mittal, et al. Livenet: A novel network for real-world low-light image denoising and enhancement. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5856–5865, 2024.

[33] Gouranga Mandal, Diptendu Bhattacharya, and Parthasarathi De. Real-time fast low-light vision enhancement for driver during driving at night. *Journal of Ambient Intelligence and Humanized Computing*, 13(2):789–798, 2022.

[34] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.

[35] Bryan A. Plummer, Liwei Wang, Chris M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *CoRR*, abs/1505.04870, 2015.

[36] C. Poynton. *Digital Video and HDTV Algorithms and Interfaces*. Computer Graphics. Morgan Kaufmann., 2003.

[37] Chenyang Qi, Junming Chen, Xin Yang, and Qifeng Chen. Real-time streaming video denoising with bidirectional buffers. In *Proceedings of*

the 30th ACM International Conference on Multimedia, pages 2758–2766, 2022.

[38] Arathy Rajan and VP Binu. Enhancement and security in surveillance video system. In *2016 International Conference on Next Generation Intelligent Systems (ICNGIS)*, pages 1–5. IEEE, 2016.

[39] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

[40] Yuda Song, Hui Qian, and Xin Du. Starenhancer: Learning real-time and style-aware image enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4126–4135, 2021.

[41] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[42] Anwaar Ulhaq, Xiaoxia Yin, Jing He, and Yanchun Zhang. Face: Fully automated context enhancement for night-time video sequences. *Journal of Visual Communication and Image Representation*, 40:682–693, 2016.

[43] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6849–6857, 2019.

[44] Shuhang Wang, Jin Zheng, Hai-Miao Hu, and Bo Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE transactions on image processing*, 22(9):3538–3548, 2013.

[45] Wenjing Wang, Chen Wei, Wenhan Yang, and Jiaying Liu. Gladnet: Low-light enhancement network with global awareness. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 751–755. IEEE, 2018.

[46] Wenjing Wang, Huan Yang, Jianlong Fu, and Jiaying Liu. Zero-reference low-light enhancement via physical quadruple priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26057–26066, 2024.

[47] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *CoRR*, abs/1808.04560, 2018.

[48] Zhenqiang Ying, Ge Li, and Wen Gao. A bio-inspired multi-exposure fusion framework for low-light image enhancement. *arXiv preprint arXiv:1711.00591*, 2017.

[49] Zhenqiang Ying, Ge Li, Yurui Ren, Ronggang Wang, and Wenmin Wang. A new low-light image enhancement algorithm using camera response model. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 3015–3022, 2017.

[50] Qing Zhang, Yongwei Nie, Ling Zhang, and Chunxia Xiao. Underexposed video enhancement via perception-driven progressive fusion. *IEEE Transactions on Visualization and Computer Graphics*, 22(6):1773–1785, 2015.

[51] Yonghua Zhang, Xiaojie Guo, Jiayi Ma, Wei Liu, and Jiawan Zhang. Beyond brightening low-light images. *International Journal of Computer Vision*, 129:1013–1037, 2021.

[52] Yonghua Zhang, Jiawan Zhang, and Xiaojie Guo. Kindling the darkness: A practical low-light image enhancer. In *Proceedings of the 27th ACM international conference on multimedia*, pages 1632–1640, 2019.

[53] Yiming Zhou, Callen MacPhee, Wesley Gunawan, Ali Farahani, and Bahram Jalali. Real-time low-light video enhancement on smartphones. *Journal of Real-Time Image Processing*, 21(5):155, 2024.