

Low-light single image super-resolution

2025년도 동계 세미나



Sogang University

Vision & Display Systems Lab, Dept. of Electronic Engineering



Presented By

Hanni Oh

Outline

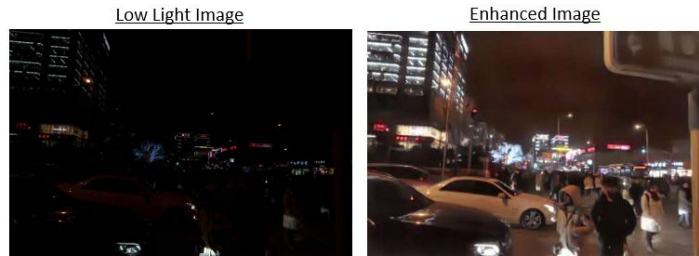
- Background
 - What is low-light image enhancement and super-resolution?
 - Why is low-light super-resolution needed?
- RELISUR: A Real Low-Light Image Super-Resolution Dataset
 - NeurIPS 2021
- LoLiSRFlow: Joint Single Image Low-light Enhancement and Super-resolution via Cross-scale Transformer-based Conditional Flow
 - ArXiv 2024

Background

- What is low-light enhancement and super-resolution?

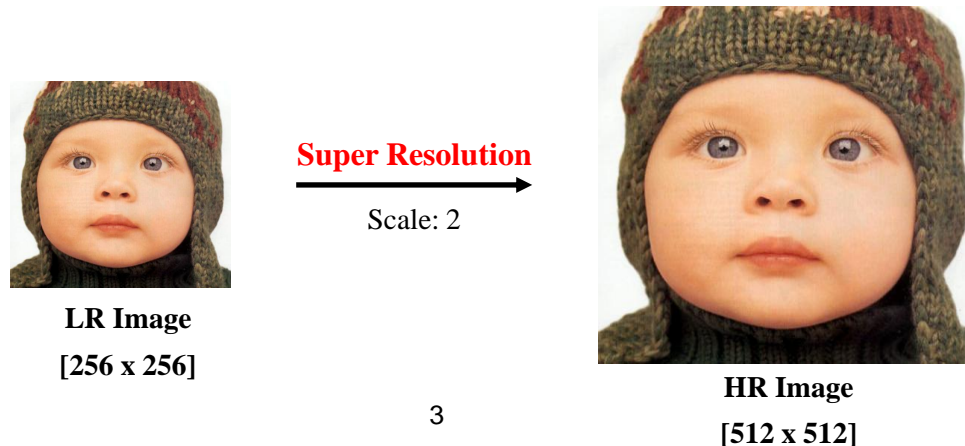
- Low-light enhancement

- 어두운 환경에서 촬영된 이미지를 개선하여 시각적 품질과 정보성을 제공하는 기술
- 밝기 개선, 노이즈 제거, 컬러 복원, 구조 복원 등



- Super-resolution

- 저화질 이미지를 고품질 이미지로 해상도를 높여주는 기술



Background

- Why is low-light super-resolution needed?

- 딥러닝 모델의 강건성 향상

- Daylight나 normal-light에서는 정상적으로 추론되던 모델이 low-light 환경에서는 추론에 실패하는 문제

- Low-light super-resolution을 통해 이를 개선하고, high-level vision 모델의 edge case에 대한 강건성을 향상시킬 수 있음

※ 자율주행, 안면인식, 번호판 인식 등



Low-light 환경에서 Object detection 추론 실패 예시¹⁾

- Low-light super-resolution

- 기존에는 low-light image enhancement와 super-resolution을 별도로 연구

- 모델 효율성 및 성능 향상을 위한 end-to-end 모델의 필요성이 대두됨

- **RELLISUR: A Real Low-Light Image Super-Resolution Dataset**

- NeurIPS 2021

Introduction

- Two degradations in digital images

- 디지털 이미지의 열화 (degradation)은 이미지의 가시성과 디테일을 저해할 수 있음
- 대표적인 degradation 종류

- Low-light

- ※ 낮은 광자 수로 인해 발생하며, 결과적으로 신호 대 잡음비(SNR) 감소

- $$\sqrt{SNR} = \frac{P_{signal}}{P_{noise}}$$
 (신호대잡음비), 높을 수록 신호가 명확하다는 의미

- ※ Exposure time 및 ISO를 증가시키면 밝기를 개선할 수 있지만, motion blur와 noise가 증가

- ※ 후처리로 밝게 하면 noise와 왜곡이 확대

- Low-resolution

- ※ 카메라의 해상도와 넓은 시야각 렌즈로 인해 디테일이 부족해짐

Introduction

- 기존 연구의 한계

- Low-light enhancement (LLE)

- Brightness 뿐만 아니라 color distortion과 noise reduction을 고려해야 함

- Super-resolution (SR)

- 대부분 HR 이미지를 단순히 blurring 및 downsampling한 synthetic LR 이미지로 학습

- 이는 현실의 복잡한 noise와 blurring 등을 학습하지 못함

- Low-light super-resolution

- 기존 LLE와 SR은 독립적으로 작동하여, 이로 인한 정보 손실과 에러 축적 문제가 발생

- ※ 즉, low-light super-resolution이 별도로 연구된 적이 없었음

- 본 논문의 저자는 이러한 문제의 이유를 low-light super-resolution real dataset의 부재라고 설명

- ※ 일반적으로 사용하는 synthetic dataset은 실제 현실의 degradation을 반영하지 못함

Introduction

- Real Low-Light Image Super-Resolution (RELLISUR)
 - 본 논문에서는 처음으로 Low Light Low Resolution (LLLR)과 Normal Light High Resolution (NLHR) 이미지 쌍으로 구성된 real dataset을 제시
 - 구성
 - 12,000 개의 이미지 쌍
 - 다양한 실내외 장면과 degradation 포함



-5.0 to -3.0 EV low light images

Corresponding $\times 1$, $\times 2$, and $\times 4$ normal light images

RELLISUR dataset 예시

RELLISUR Dataset

- Collection method

- 촬영 장비

- Canon EOS 6D camera

- 촬영 환경

- 다양한 실내 및 실외 장면 포함

- 풍경, 사물, 텍스처 등 다양한 contents와 degradation 반영

- Low-light 이미지 생성

- Exposure time을 변경함으로써 low-light 시나리오 구현

- 조명 수준: from -5.0 EV to -3.0 EV

- Low-resolution 이미지 생성

- 카메라의 zoom lens의 focal length를 변경하여 서로 다른 해상도의 이미지를 초촬영

- ※ 짧은 focal length: LR 이미지를 생성

- ※ 긴 focal length: HR 이미지 생성

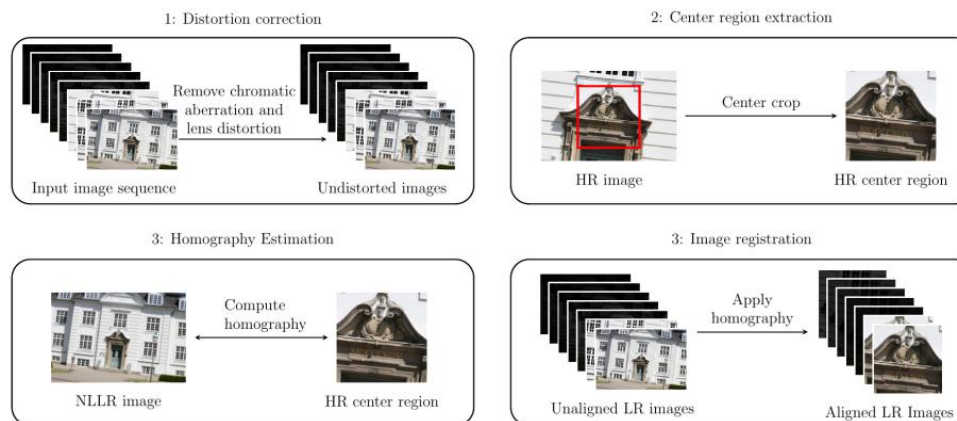
- ✓ 동일한 장면을 다양한 해상도로 촬영하여 $\times 1$, $\times 2$, $\times 4$ scale 구현

RELLISUR Dataset

• Preprocessing

• Zooming에서 발생하는 lens의 특성 (warping과 distortion) 및 external factors (wind 등)의 영향을 제거

- 1. 수동으로 out-of-focus 등 잘못된 이미지를 제거
- 2. Adobe Lightroom을 통해 lens correction 진행 (Chromatic aberration, lens distortion, ect.)
- 3. 가장자리의 왜곡을 제거하기 위해 Normal Light High Resolution (NLHR) 이미지를 center crop
- 4. NLLR 이미지와 LLLR 이미지를 NLHR 이미지와 matching하기 위해 MSAC으로 homography 추정
- 5. $\times 1$ LLLR 이미지와 $\times 1$, $\times 2$ NLLR 이미지를 $\times 4$ NLHR 이미지에 맞춰서 CROP



Preprocessing 과정

Experiments

• 실험 개요

- LLE와 SR을 결합한 선행연구가 없기 때문에, 본 논문에서는 sequential 방법과 joint 방법으로 진행
 - Sequential은 LLE와 SR을 순차적으로 진행하며, joint는 LLE 모델에 upscaling module을 추가하거나 SR 모델을 단순 학습

• 실험 목표

- LLLR 이미지를 입력으로 받아, NLHR 이미지를 출력

• 실험 결과

- MIRNet에 upscaling module을 추가하여 학습시킨 방법이 가장 높은 PSNR/SSIM
- Human judgement를 반영한 LPIPS는 ESRGAN이 가장 높은 성능 달성

Table 7: Simultaneous LLE and SR results for different approaches trained and tested on the RELLISUR dataset.

Type	Name	×2			×4		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Sequential	MIRNet + DBPN [39, 46]	20.73	0.73	0.49	19.85	0.74	0.58
	MIRNet + ESRGAN [39, 27]	20.67	0.72	0.47	19.81	0.71	0.56
	MBLLEN + DBPN [40, 46]	17.89	0.60	0.38	17.15	0.58	0.50
	MBLLEN + ESRGAN [40, 27]	17.74	0.56	0.40	17.03	0.50	0.52
Joint	MIRNet [39] + Upscaling module	21.33	0.75	0.41	20.62	0.75	0.53
	ESRGAN [27]	17.67	0.68	0.35	17.28	0.66	0.39

- **LoLiSRFlow: Joint Single Image Low-light Enhancement and Super-resolution via Cross-scale Transformer-based Conditional Flow**

- ArXiv 2024

Introduction

- Challenges in Low-Light and Low-Resolution Image Restoration

- 실제 환경에서 디지털 이미지는 low-light와 low-resolution degradation으로 인해 품질 저하 발생

- Low-light: 낮은 SNR로 인해 노이즈 발생 및 디테일 손실

- Low-resolution: 중요한 디테일 (e.g. 텍스트, 얼굴)을 손실

- 기존 연구의 한계

- Low-light enhancement (LLE)와 super-resolution (SR)을 독립적으로 처리하여 sequential하게 연결

- Sequential 방법의 단점

- ※ LLE에서 발생한 noise와 artifact가 SR 과정으로 전달되어 최종 복원 성능 저하

- ※ LLE와 SR은 서로 상호작용하는 문제이지만, 이를 고려하지 않음

Introduction

- LoLiSRFlow

- Low-light enhancement (LLE)와 super-resolution (SR)을 하나의 프레임워크에서 통합적으로 처리
- 기존 one-to-one mapping 대신, HR 이미지의 확률 분포를 학습하여 자연스러운 복원 진행
- 핵심 아이디어
 - Color ratio map (CR MAP)
 - ※ 조명과 해상도에 독립적인 본질적 특징을 제공
 - Transformer 기반 인코더
 - ※ 해상도와 밝기 변화에 강건한 특성 추출
 - Normalizing flow
 - ※ Reversible network layer로 noise와 distortion을 최소화함

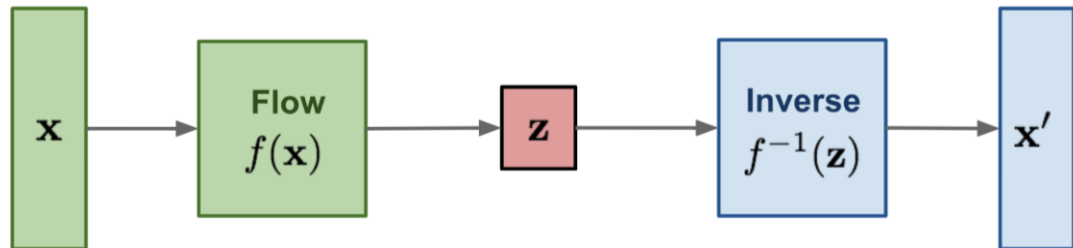
Preliminary

- Normalizing flow

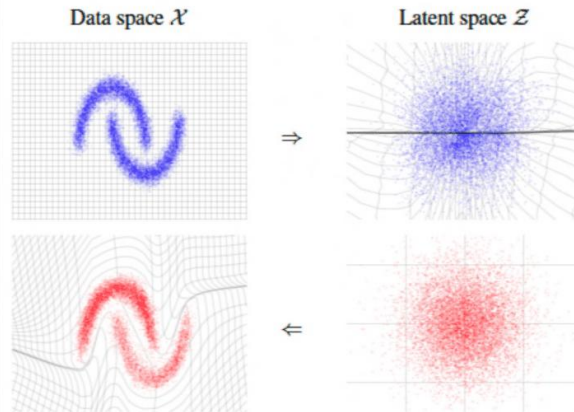
- 간단한 분포(e.g. 정규분포)를 복잡한 데이터 분포로 변환하는 일련의 reversible transformation

- x 를 latent space의 z 로 변환하는 정방향 연산과 z 를 x 로 변환하는 역방향 연산이 정보 손실 없이 가능

Flow-based generative models:
minimize the negative log-likelihood



Inference
 $x \rightarrow z$



Normalizing flow 과정

Preliminary

- Conditional normalizing flow

- 목표

- low-light low-resolution 이미지 x 를 입력 받아, normal-light high-resolution 이미지 y 를 복원

- 단순히 x 에서 y 를 mapping하는 방법 대신, 조건부 확률 분포 $P_{y|x}(y|x)$ 를 학습

- ※ 단순 mapping은 입력과 출력 간의 복잡한 분포를 이해하지 못함

- $z = F_\theta(y|x)$, z : latent variable, x : LLLR image, y : NLHR image, F : invertible flow model

- $y = F_\theta^{-1}(z|x)$

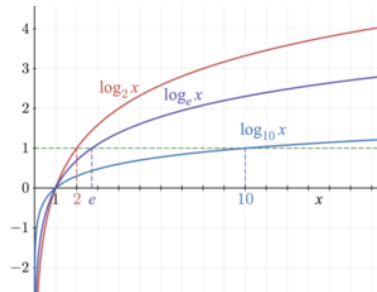
- $P_{y|x}(y|x; \theta) = P_z(F_\theta(y, x)) |det(\mathcal{J}_F)|$, 변수 치환 정리를 사용해 latent space 상의 확률 밀도 함수 $P_{y|x}(y|x; \theta)$ 를 모델링

- $L_{nll}(y, x; \theta) = -\log P_{y|x}(y|x; \theta)$

- $= -\log P_z(F_\theta(y, x)) - \log |det(\mathcal{J}_F)|$.

$P_{y|x}$ 를 학습하기 위해 negative log-likelihood 최적화

- ※ $P_{y|x}$ 이 커질수록 NLL 값은 작아지며, 이는 모델이 데이터를 더 잘 설명한다는 의미임



Preliminary

- Resolution- and illumination-invariant map

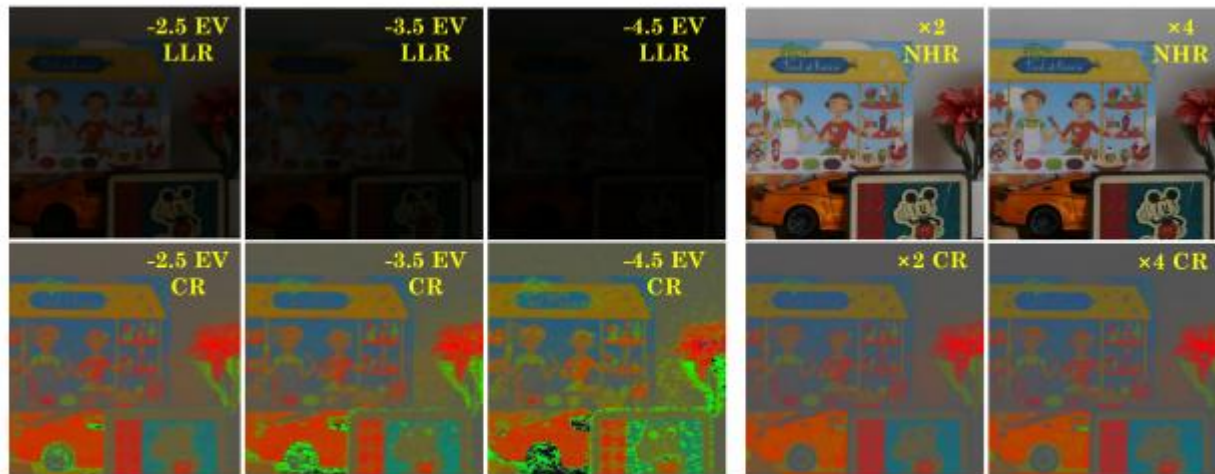
- 입력 이미지 I 가 주어졌을 때, 해상도와 조도에 독립적인 color ratio map을 계산

$$-CR(I) = \frac{I}{\text{sum}(I)}$$

- Training phase에서 랜덤하게 data pair $\{x, y\}$ 를 선택하여

- 세 가지 다른 수준의 exposure의 low-light low-resolution 이미지 x 와 normal-light high-resolution 이미지 y 로 구성

- Exposure에 상관없이 color ratio map은 보존되는 것을 확인



Color ratio map 예시

Methodology

- LoLiSRFlow

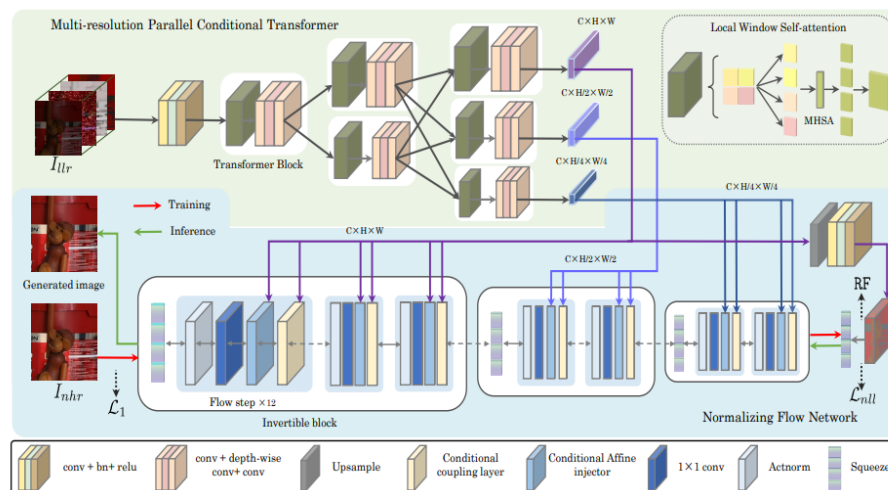
- 입력 이미지 I 가 주어졌을 때, 해상도와 조도에 독립적인 color ratio map을 계산
- 두 가지 main module로 구성

- Multi-resolution parallel transformer-based conditional encoder E

- ※ Low-light low resolution 이미지 x 를 입력으로 받아 color ratio map과 여러 scale의 conditional feature map을 출력

- Invertible normalizing flow network F

- ※ Normal-light high-resolution 이미지 y 를 latent encoding z 로 mapping

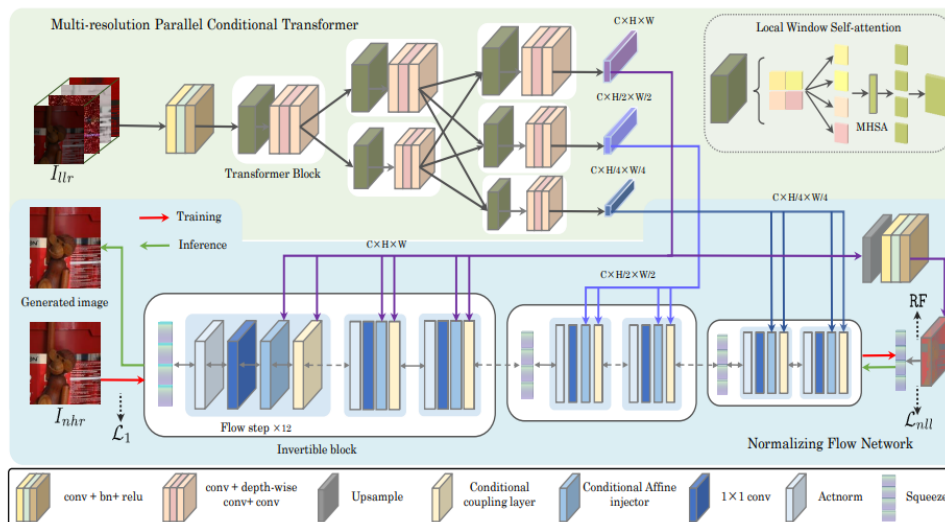


LoLiSRFlow architecture

Methodology

- Multi-resolution parallel conditional encoder

- 기존 encoder는 latent embedding을 위해 one-to-one mapping을 학습
 - 즉, low-light low-resolution 입력 x 를 color ratio map으로 mapping하는 식
- 그러나 전통적인 residual convolutional network 기반 encoder는 high-frequency 정보를 잃고 과도한 artifact를 생성함
- 따라서 본 연구에서는 multi-scale transformer 기반 encoder를 통해 여러 resolution의 feature를 혼합



LoLiSRFlow architecture

Methodology

- Invertible normalizing flow network

- 총 3개 scale의 invertible block으로 구성되었고, 각 block은 12 flow step을 포함하며 각 flow step은 4개의 같은 flow layer를 공유

- 각 flow step마다 squeeze와 split 연산이 수행

- ※ Squeezing operation: resolution을 줄이고 channel 수를 늘림

- ※ Split operation: channel 순서대로 feature를 절반씩 자름

- 각 flow block은 actnorm layer, 1×1 invertible convolution, conditional coupling layer, affine injector로 구성

- ※ Actnorm layer: 평균과 표준편차를 학습하여 채널을 reversible하게 정규화

- ※ 1×1 invertible convolution: reversible한 1×1 convolution layer

- ※ Conditional coupling layer: 데이터를 일부를 고정한 상태에서 나머지를 변환

- ※ Affine injector: conditional information을 모델에 주입하여 변환을 더 정교하게 수행

Methodology

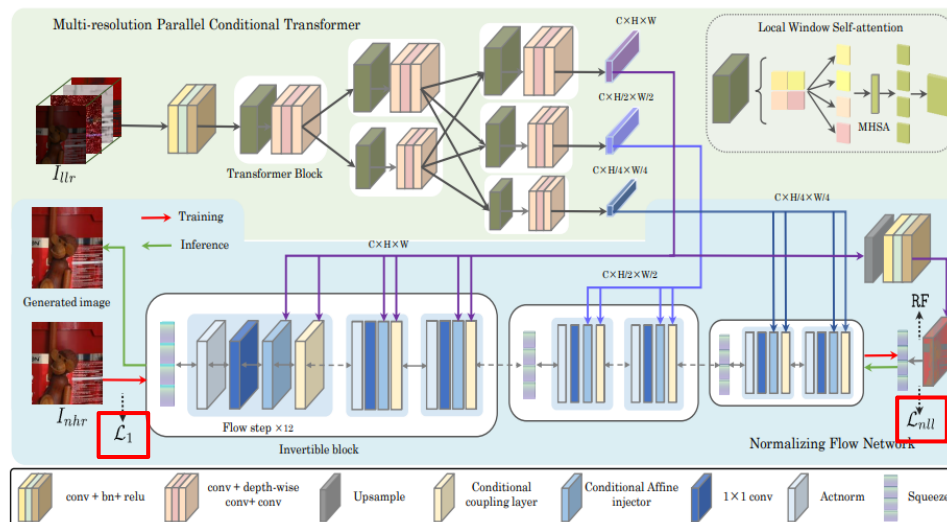
- Loss

- Negative log-likelihood loss L_{nll} 과 L_1 loss를 함께 학습

$-L_1 = \| F_{\theta}^{-1}(z/x) - y \|$, z : latent image, y : normal-light high-resolution image

$-L_{nll}(y, x; \theta) = -\log P_{y|x}(y|x; \theta)$, x : low-light low-resolution image
 $= -\log P_z(F_{\theta}(y, x)) - \log |\det(J_F)|$.

$-L_{total} = L_{nll} + \gamma L_1, \gamma = 1.5$



Training되는 loss

Experiment

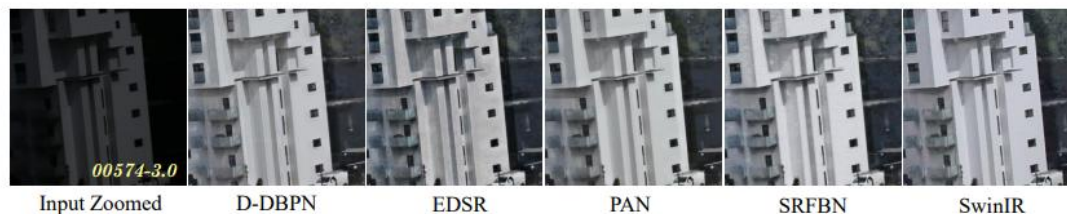
- Dataset
 - RELLISUR
- Data augmentation
 - Random cropping, random rotation, random flip
 - Low-light low-resolution 이미지에 histogram equalization 수행
- 정량 결과
 - 가장 높은 PSNR/SSIM 달성

Scale	Metrics	D-DBPN	EDSR	PAN	SRFBN	SwinIR	RUAS→SwinIR	MIRNet	SRFormer	HAT	Ours
×2	PSNR	18.70	18.38	18.78	18.42	18.38	17.15	21.05	19.55	20.21	23.40
	SSIM	0.682	0.679	0.693	0.662	0.640	0.617	0.720	0.704	0.719	0.753
×4	PSNR	17.96	17.69	18.10	17.67	17.07	19.78	19.34	18.72	19.75	21.58
	SSIM	0.674	0.679	0.700	0.665	0.663	0.662	0.704	0.705	0.715	0.742

Experiment

• 정성 결과

- 다른 방법론보다 LoLiSRFlow가 밝기를 과도하게 높이지 않으면서도 noise를 더하지 않는, GT와 비슷한 결과가 나옴
- 과하게 어두운 환경에서도 texture detail을 살리는 결과가 도출



감사합니다.