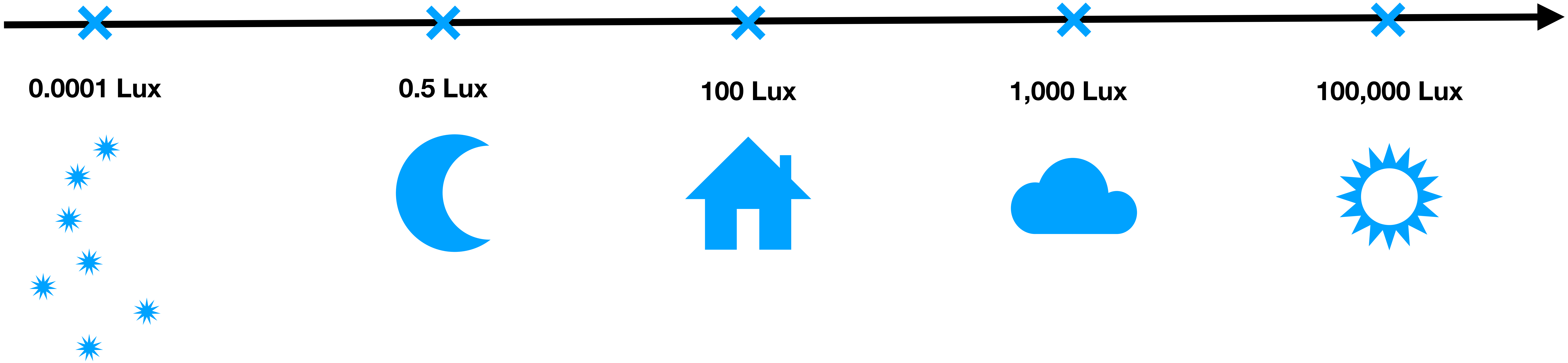
Modern High Dynamic Range Imaging at the Time of Deep Learning

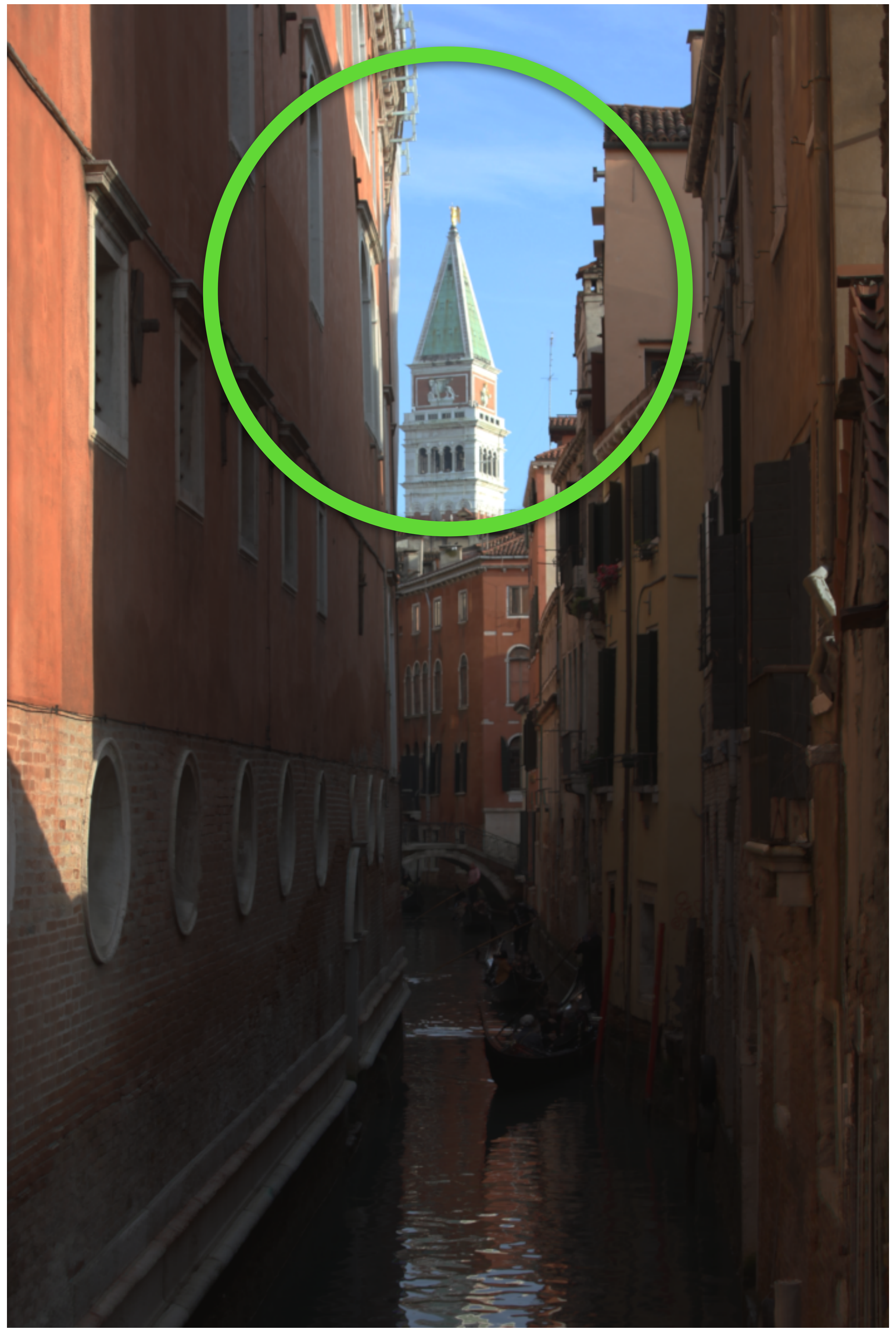
Introduction

Francesco Banterle and Alessandro Artusi

HDR Imaging



HDR Imaging



Short Exposure

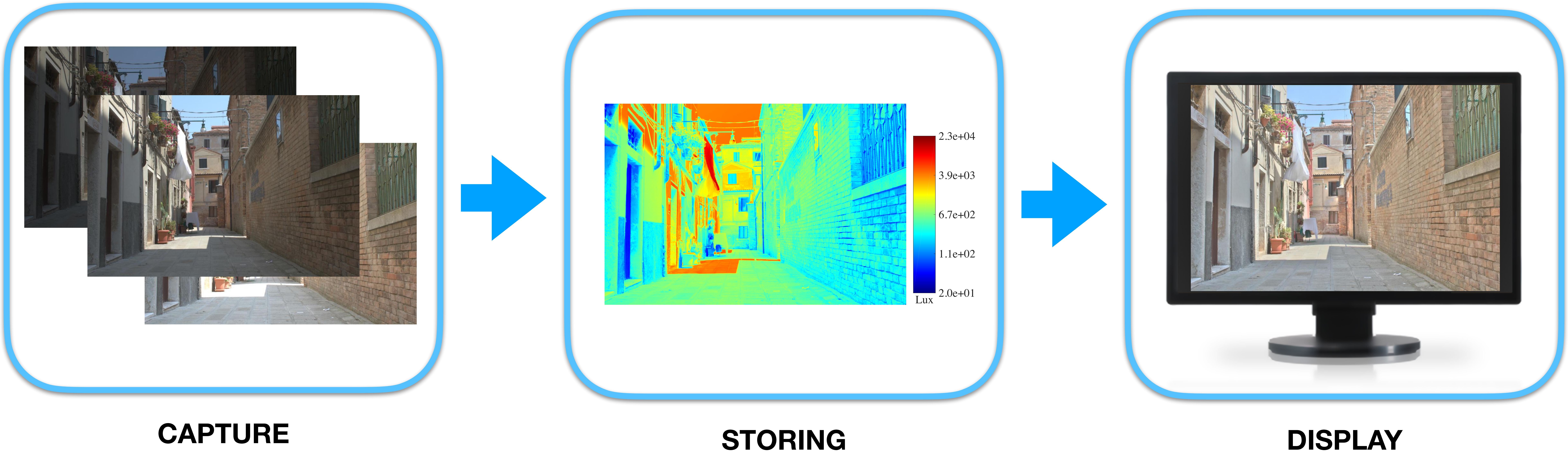
Mid Exposure

Long Exposure

HDR Imaging

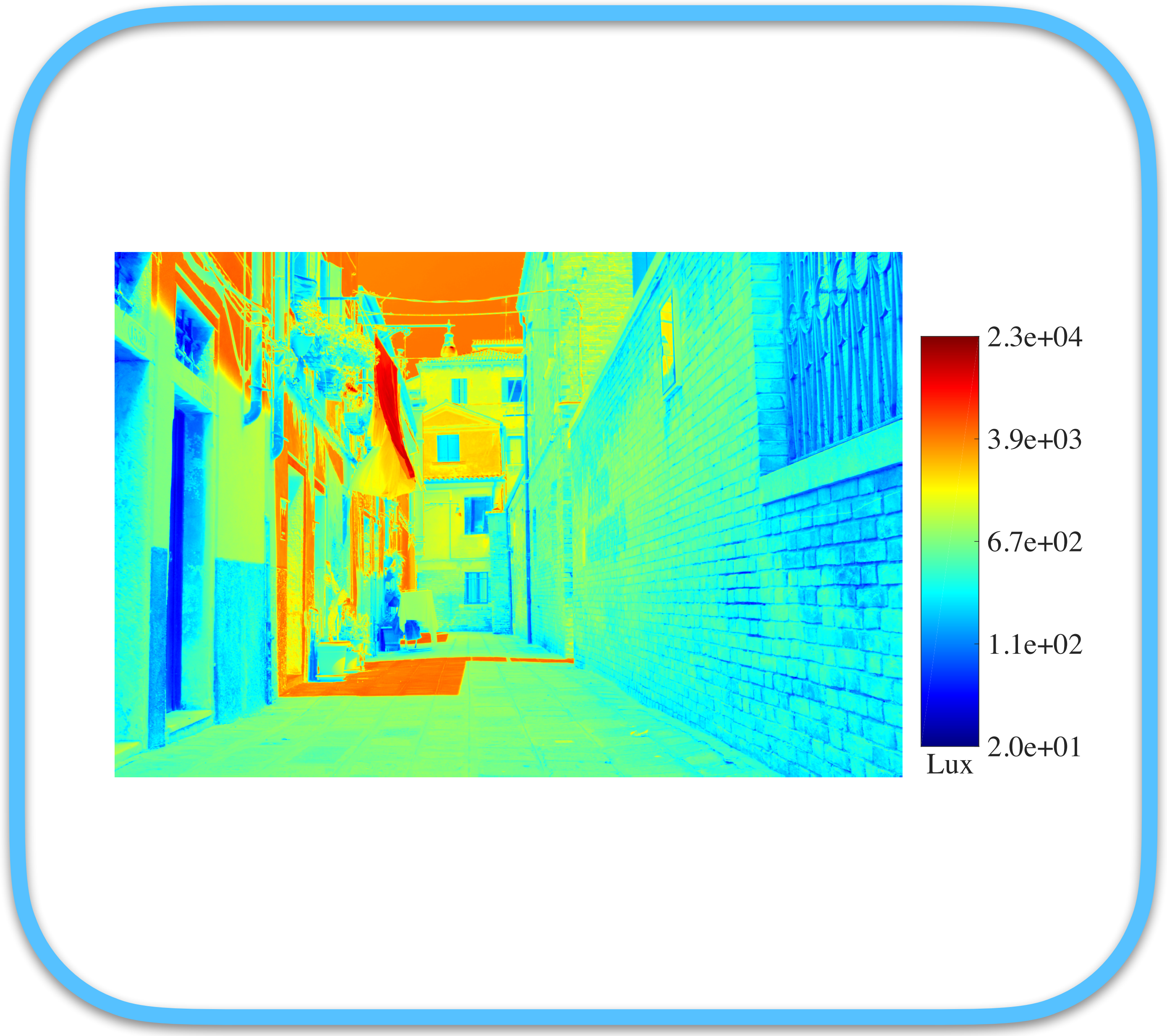
Merged Exposures

The HDR Pipeline


2.3e+04
3.9e+03
6.7e+02
1.1e+02
2.0e+01
Lux
CAPTURE
STORING
DISPLAY

HDR Imaging: Merging

CAPTURE



STORING

DISPLAY

HDR Imaging: Acquisition

HDR Imaging: Merging

- To merge N images, Z_k , at different exposure times, t_k, we sum them up taking into account that they were taken at different shutter speed:

$$E(i,j) = \frac{\sum_{k=1}^{N} w(Z_k(i,j)) \cdot g(Z_k(i,j)) \cdot t_k^{-1}}{\sum_{k=1}^{N} w(Z_k(i,j))}$$

- where $g = f^{-1}$ is the inverse camera response function, and w is a weighting function. Typically, the merge is computed in the log-domain to reduce noise.

HDR Imaging: Merging

- The result $E(i,j)$ is a **radiance map**:
  - Note E is the irradiance symbol; the radiance symbol is L:
    - Technically speaking we should taking into account that:

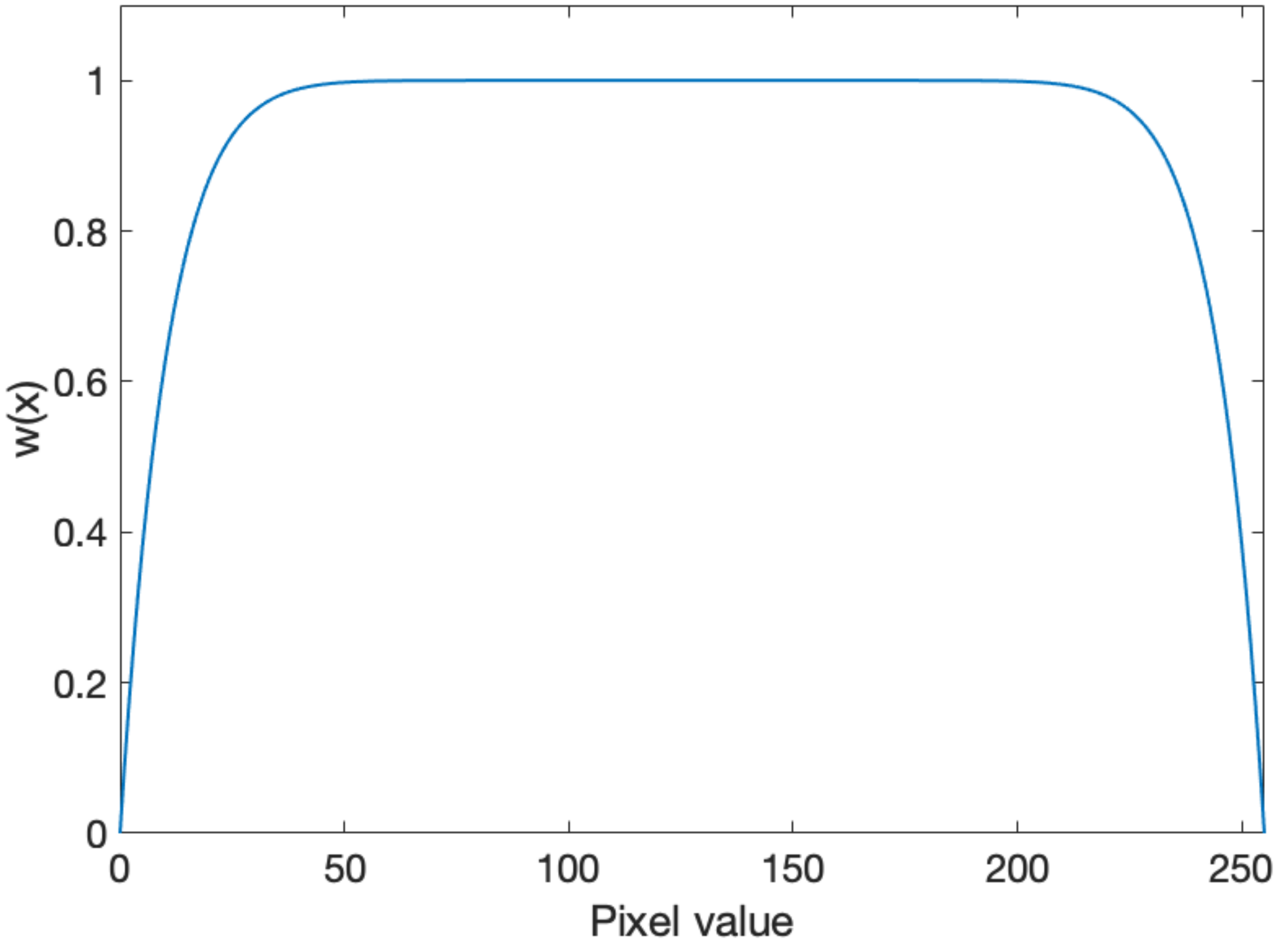
$$E(i,j) = L(i,j)\frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos^4 \alpha$$

    - But… Most lenses already compensate for this!

HDR Imaging: The Weighting Function

- The weighting function selects well-exposed pixels from the input image to avoid noisy and saturated pixels:
  - Such value increase noise or bias in the final HDR image.
- For example:

$$w(x) = 1 - (2x - 1)^{12}$$



HDR Imaging: Camera Response Function

- A Camera Response Function (CRF), f, is a non-linear function of image irradiance:
  - It is a solution for compressing the irradiance values large dynamic range into a fixed range of recordable values; i.e., 8-bit of a JPEG image.
    - RAW images (stored in 10-14 bits) have mostly a linear behavior.
  - It is typically not known, but it can be estimated.

HDR Imaging: Camera Response Function

- Exploiting:

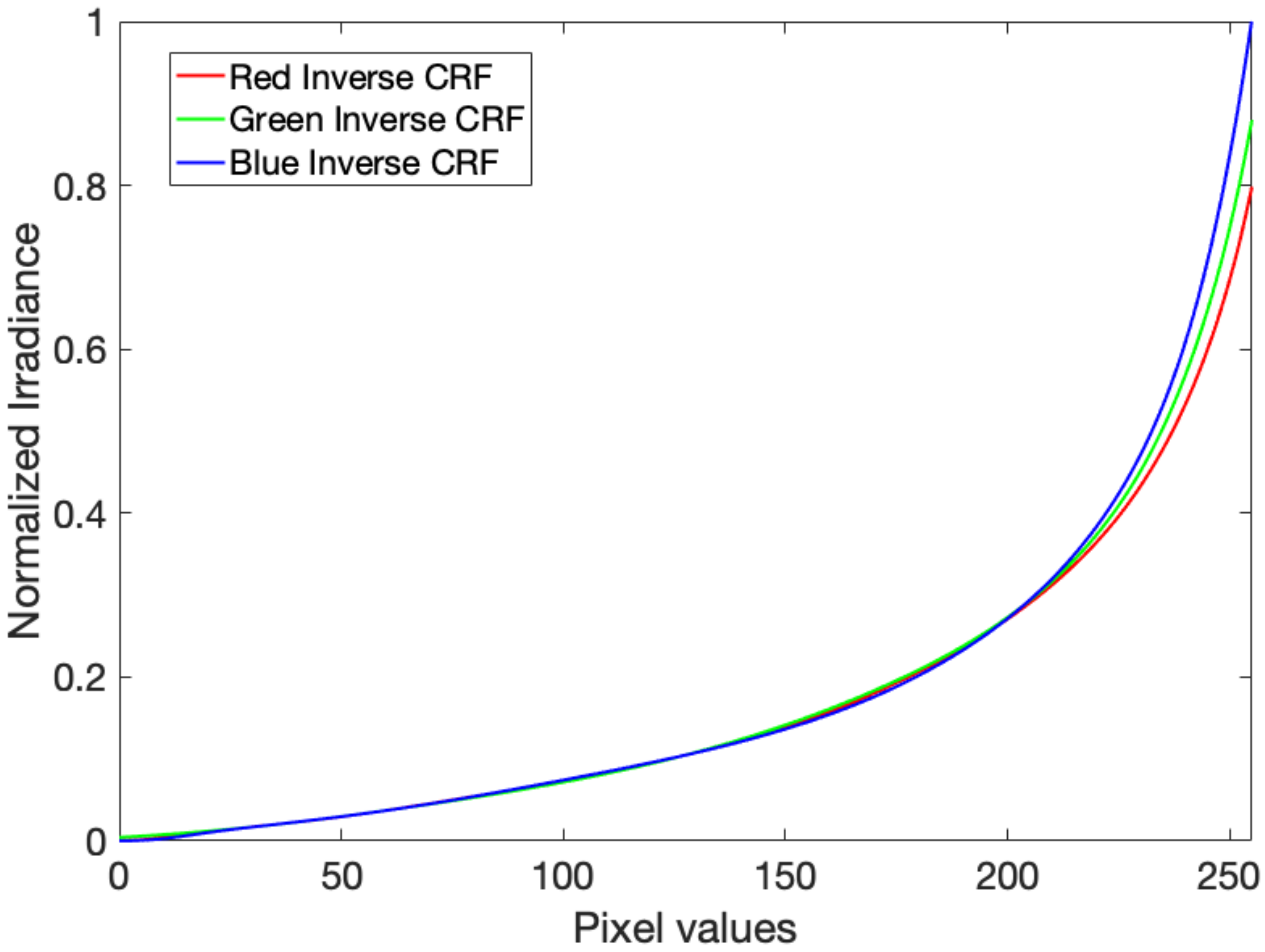
$$Z_k(i,j) = f\big(E(i,j)t_k\big) \rightarrow f^{-1}\big(Z_k(i,j)\big) = E(i,j)t_k \rightarrow \log f^{-1}\big(Z_k(i,j)\big) = \log E(i,j) + \log t_k$$

- A typical estimation method is based on optimization:

$$\mathcal{O} = \sum_{k=1}^{N} \sum_{i,j} \left(\log g(Z_k(i,j)) - \log E(i,j) - \log t_j \right)^2 + \lambda \sum_{x} g''(x)^2$$

- Where $g(x) = f^{-1}(x)$.

HDR Imaging: Camera Response Function



HDR Imaging: Camera Response Function

- Nowadays, many cameras/smartphone manufactures and displays makers have started to agree on some standard CRF or OETF. Most famous examples:
  - PQ:

$$f(Y) = \left(\frac{c_1 + c_2 Y_n^{m_1}}{1 + c_3 Y_n^{m_1}}\right)^{m_2} \text{ where } Y_n = \frac{Y}{10000}$$

  - HLG:

$$f(Y) = \begin{cases} r\sqrt{Y} & Y \in [0,1] \\ a\log(Y-b)+c & Y > 1 \end{cases}$$

HDR Videos

- There are different strategies:
  - Multiple sensors combined with beam splitter capturing frames at different exposures time [Tocci+2011].
  - Varying the exposure shutter speed at each frame [Kang+2003].
  - Varying the exposure time in the bayer filter or assorted pixels [Yasuma+2010].

HDR Videos: Multiple Sensors

Stream 1
Stream 2
Stream 3

t_0 t_1 t_2

Video Courtesy of Jan Fröhlich - Stuttgart HDR Video Dataset

HDR Videos: Varying Exposure at Each Frame

Stream

t_0 t_1 t_2

Video Courtesy of Jan Fröhlich - Stuttgart HDR Video Dataset

HDR Videos: Assorted Pixels

Video Courtesy of Jan Fröhlich - Stuttgart HDR Video Dataset

HDR Videos: Assorted Rows

Video Courtesy of Jan Fröhlich - Stuttgart HDR Video Dataset

HDR Imaging:
Tone Mapping - SDR Visualization

Tone Mapping

- A tone mapping operator (TMO) is a function, $f(\cdot)$, that reduces the dynamic range of a HDR image to fit into a SDR display. We have two main classes:
  - **Global operators**: it uses global statistics of the image to be tone mapped:
    - We want to maintain the global contrast of the original image.
  - **Local operators**: it uses both global and local statistics of the image to be tone mapped:
    - We want to maintain both the local and global contrast of the original image.

Tone Mapping

- Most operators work only on the luminance channel:

$$\begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = \frac{f(L_w)}{L_w} \cdot \begin{bmatrix} R_w \\ G_w \\ B_w \end{bmatrix} = \frac{L_d}{L_w} \cdot \begin{bmatrix} R_w \\ G_w \\ B_w \end{bmatrix}$$

- For the sRGB color space, this is defined as

$$L_w = 0.2126 \cdot R_w + 0.7152 \cdot G_w + 0.0722 \cdot B_w$$

- This is to avoid color distortions when applying the curve on the three color channels.

Tone Mapping: Global Operators

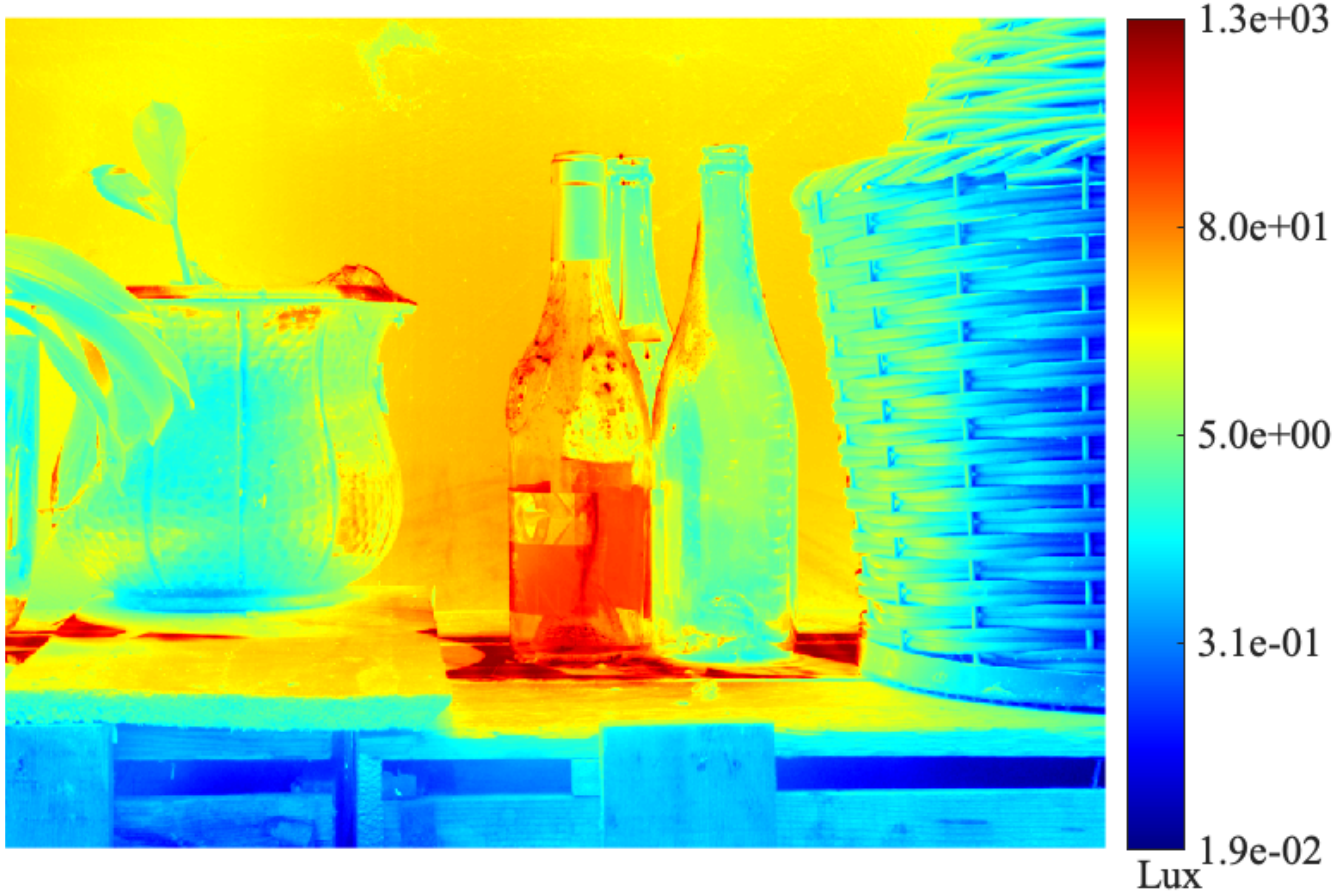
- A classic local TMO is the Reinhard operator [Reinhard+2002]:

$$L_d = f(L_w) = \frac{L_m}{1 + L_m} \qquad L_m = \frac{\alpha}{\hat{L}_w} L_w$$

- where α is a user parameter, and $\hat{Y}$ is the geometric mean of the luminance of the entire image:

$$\hat{L}_w = \exp\left(\frac{1}{n} \sum_{i,j} \log_e L_w(i,j) + \delta \right)$$

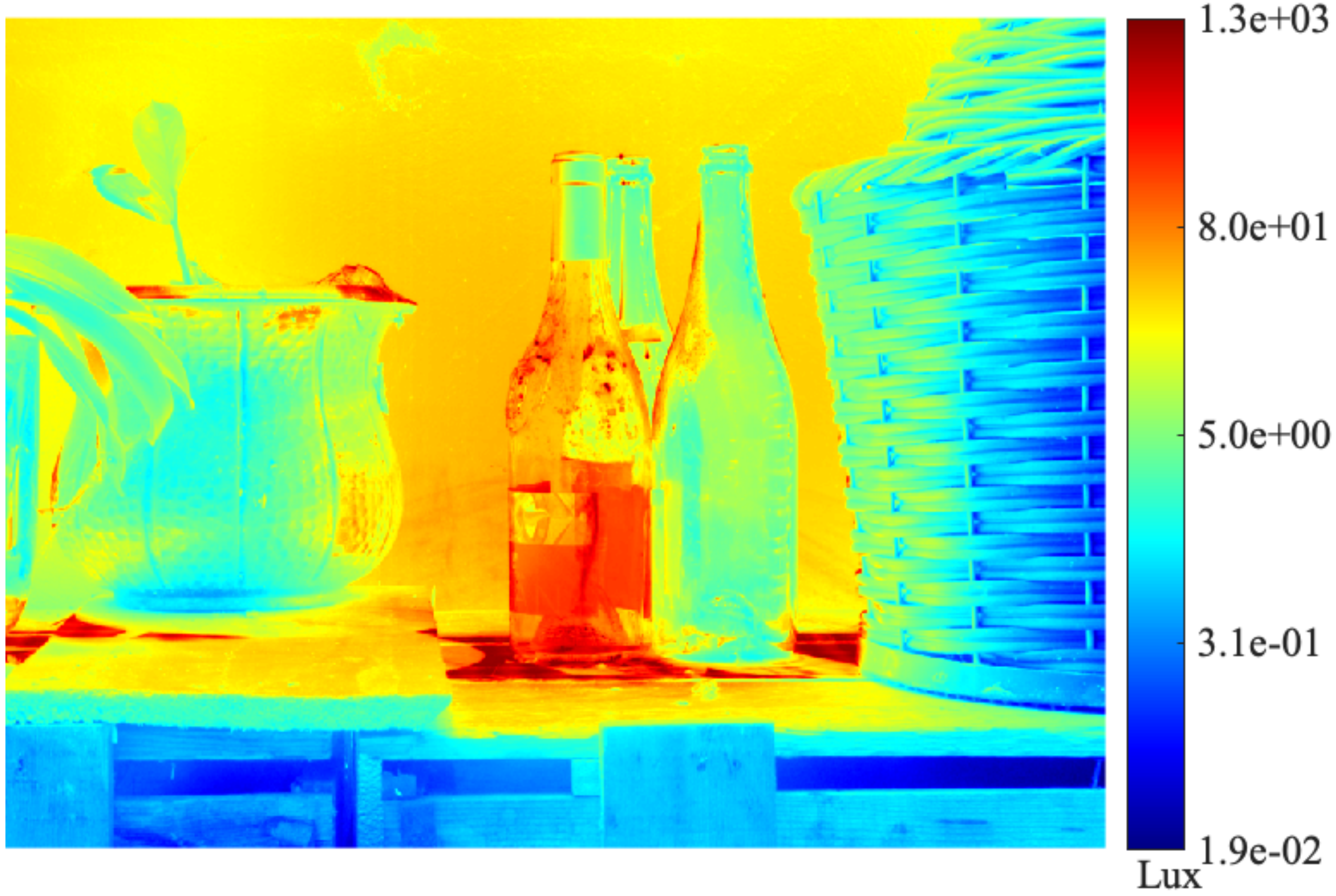
Tone Mapping: Global Operators Example



HDR Image

Reinhard with $\hat{L}_w = 1$

Tone Mapping: Global Operators Example



HDR Image

Reinhard with $\hat{L}_w$ computed

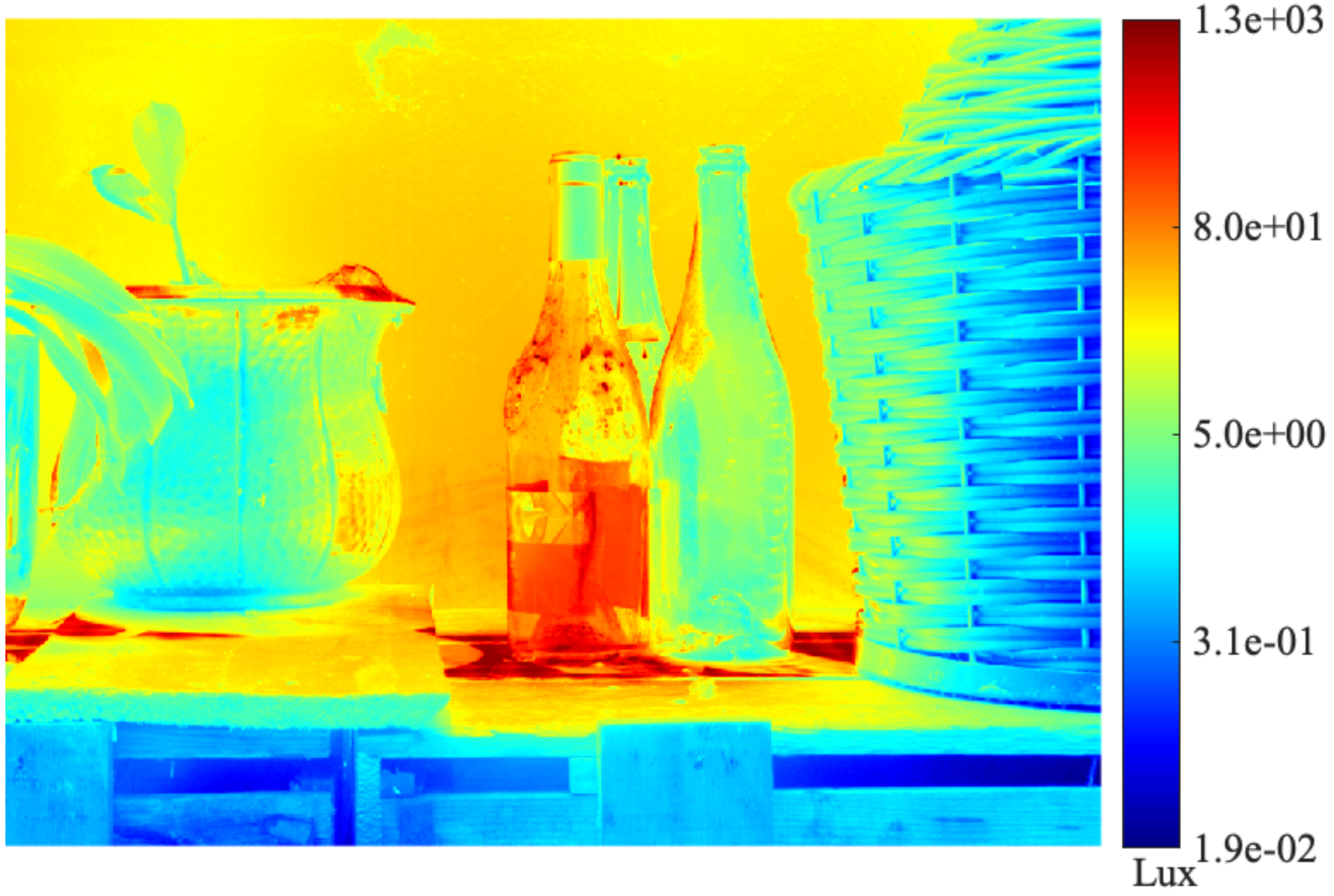
Tone Mapping: Local Operators

- A classic local TMO is a variant of the Reinhard operator [Reinhard+2002]:

$$L_d = f(L_w(i,j)) = \frac{L_m(i,j)}{1 + g(L_m(i,j))} \quad L_m(i,j) = \frac{\alpha}{\hat{L}} L_w(i,j)$$

- where $g(\cdot)$ is a function computing the mean around the pixel (i,j). However, we need to avoid strong edges that may create halos. So $g(\cdot)$ has to be edge-aware; e.g., the bilateral filter.

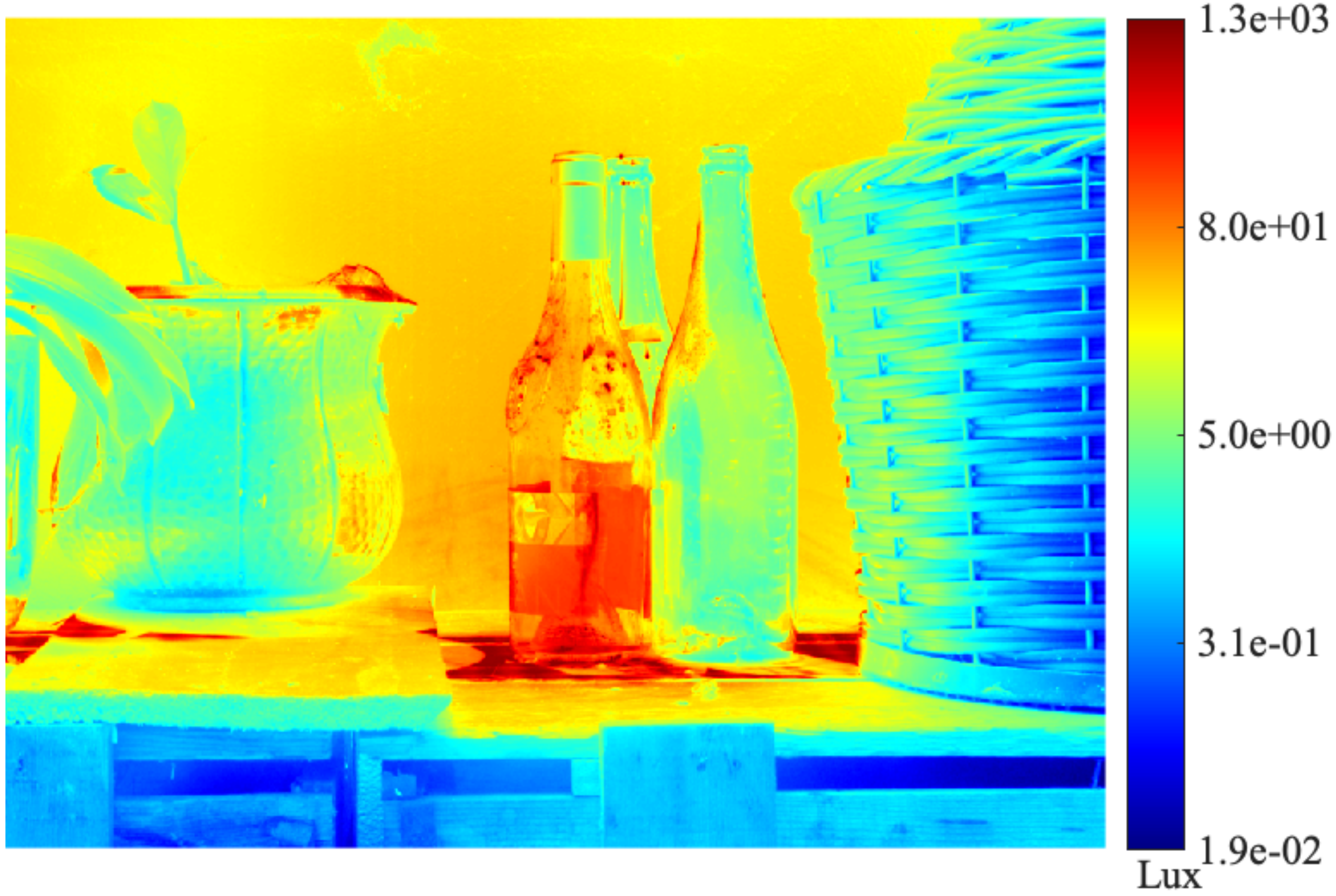
Tone Mapping: Local Operators Example



HDR Image

Reinhard without an edge-preserving filter

Tone Mapping: Local Operators Example



HDR Image

Reinhard with an edge-preserving filter

Color Distortions

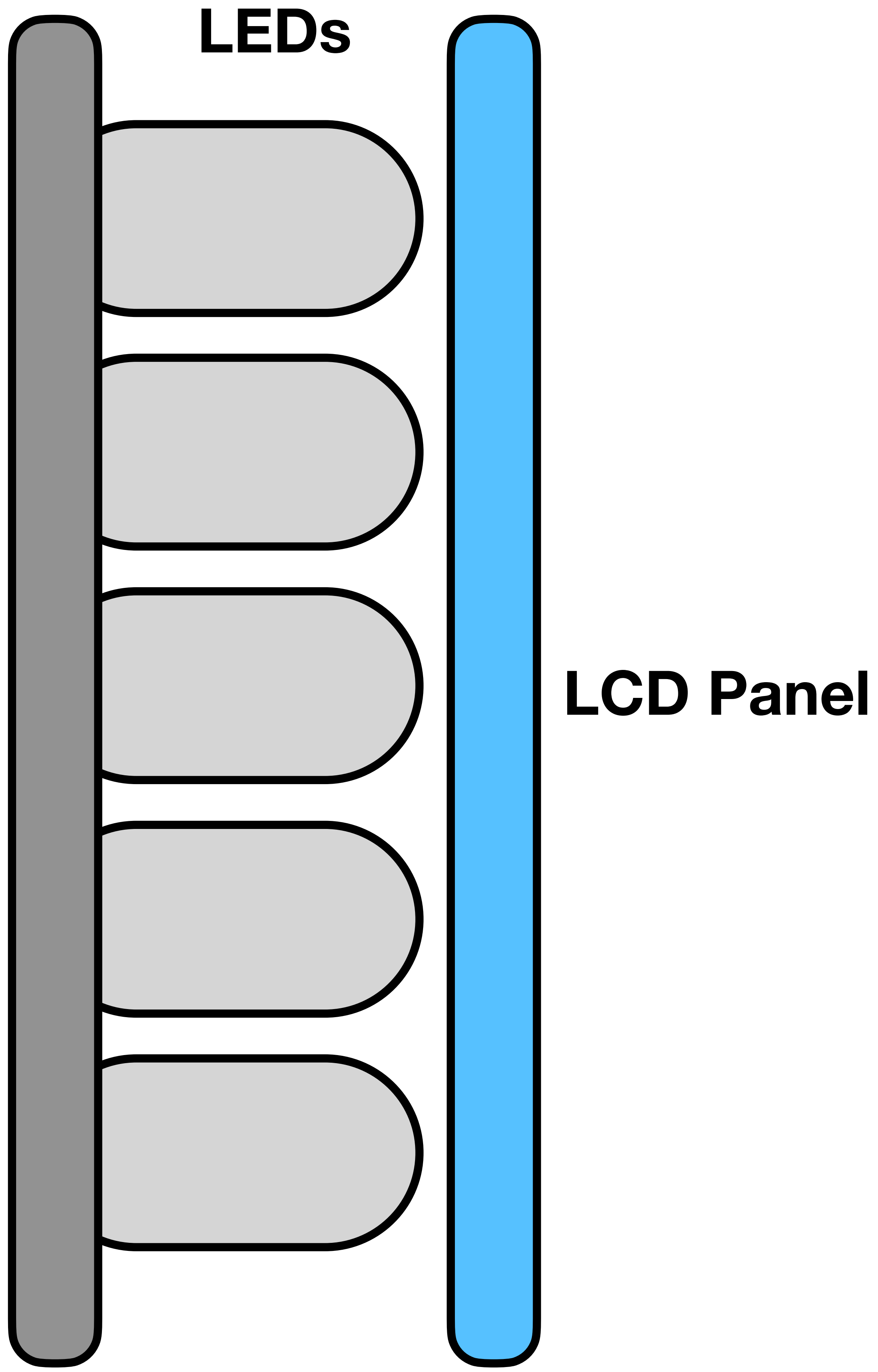
- The problem with processing only the luminance is that we have the following problems:
    - $L_d < L_w$ the saturation of the pixel increases.
    - $L_d > L_w$ the saturation of the pixel decreases.

Color Solutions

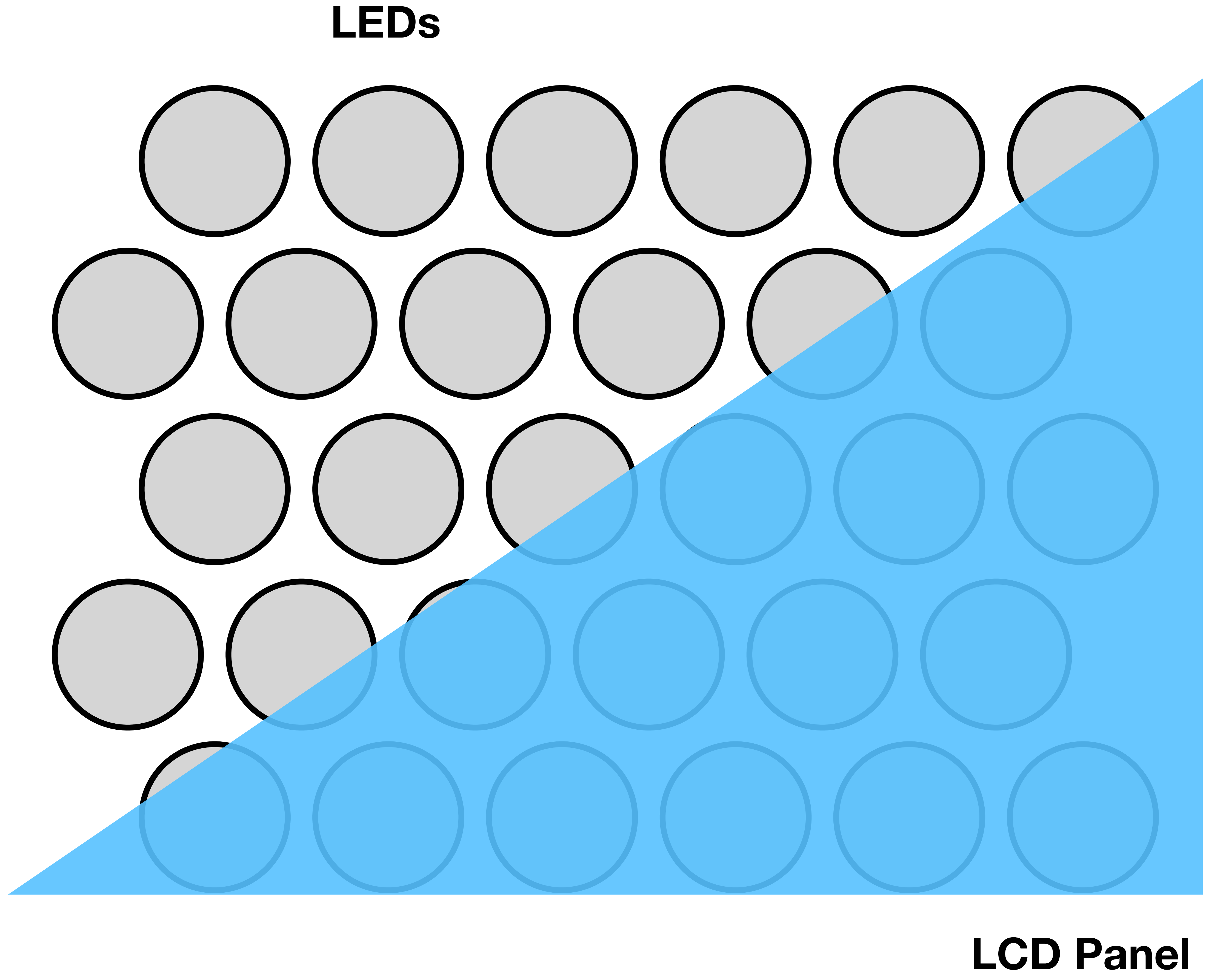
- Main solutions:
  - Desaturate $[R_w, G_w, B_w]/L_w$ by applying a power function in $(0,1]$ [Schlick+1995].
  - Linear desaturation taking into account the TMO derivate [Mantiuk+2009].
  - Hue reset and saturation scale in the LCh color space [Pouli+2013].

HDR Imaging:
Native Visualization - HDR Monitors

Native Visualization: LEDs HDR Monitors

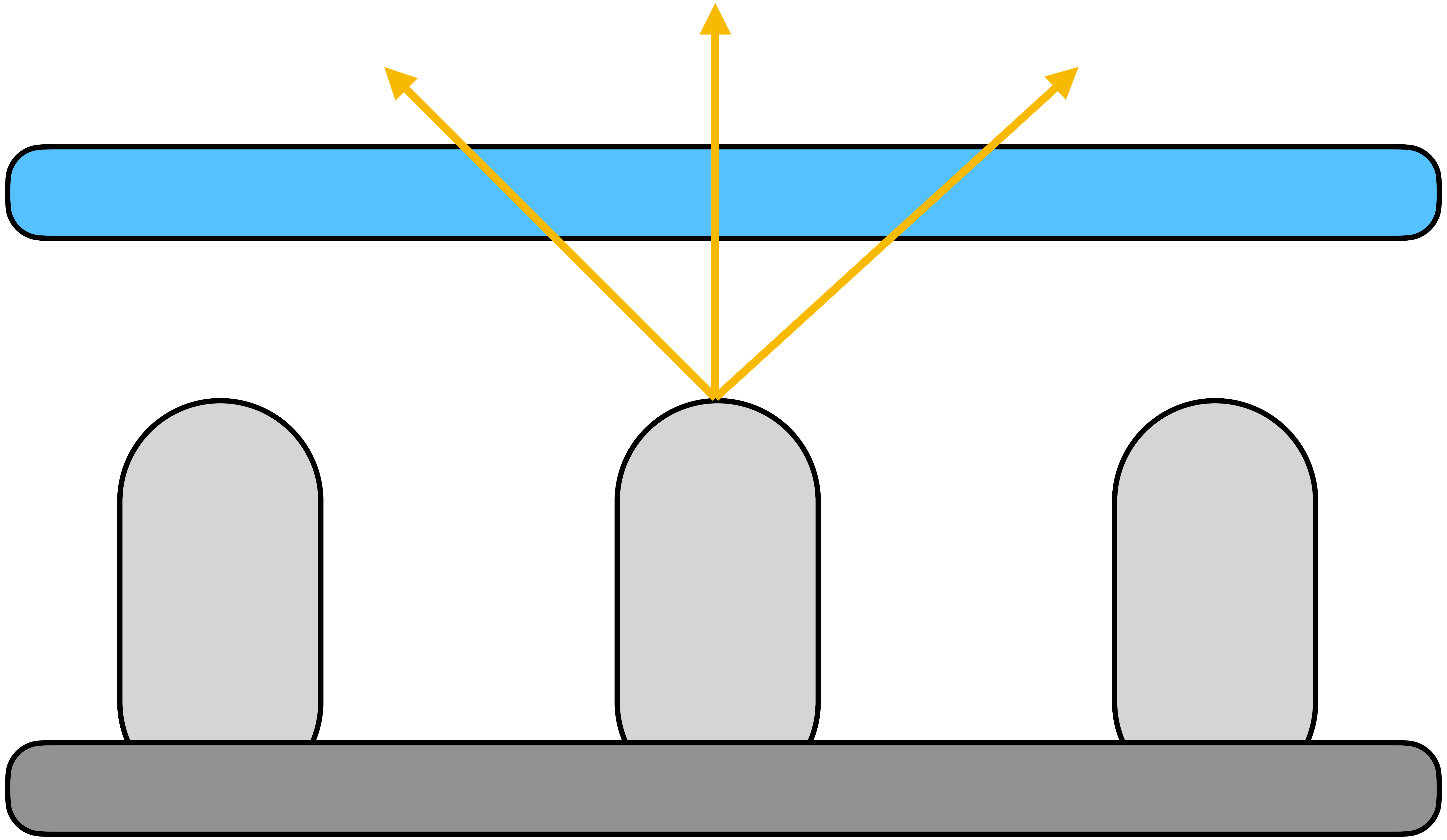


SIDE

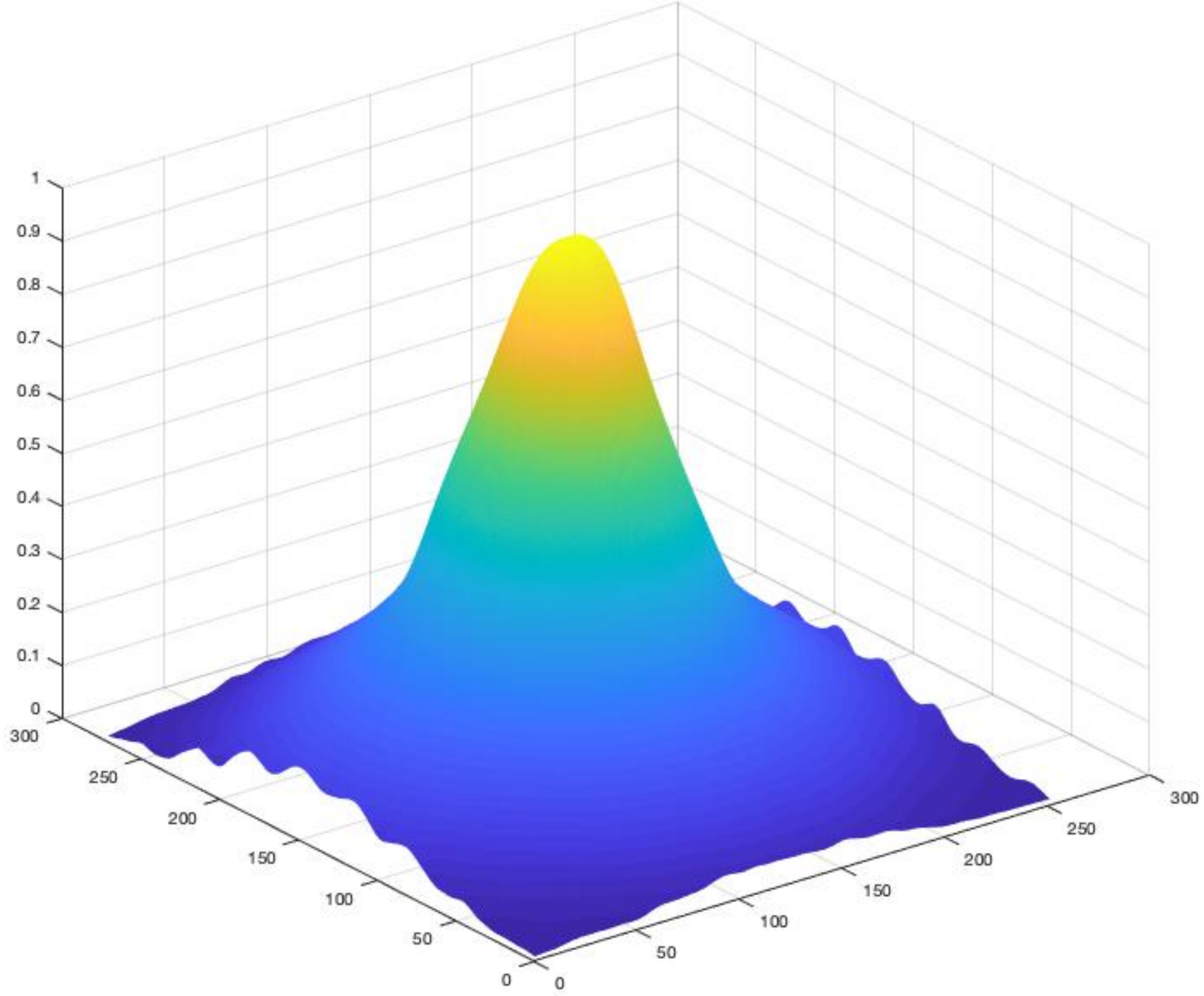


FRONT

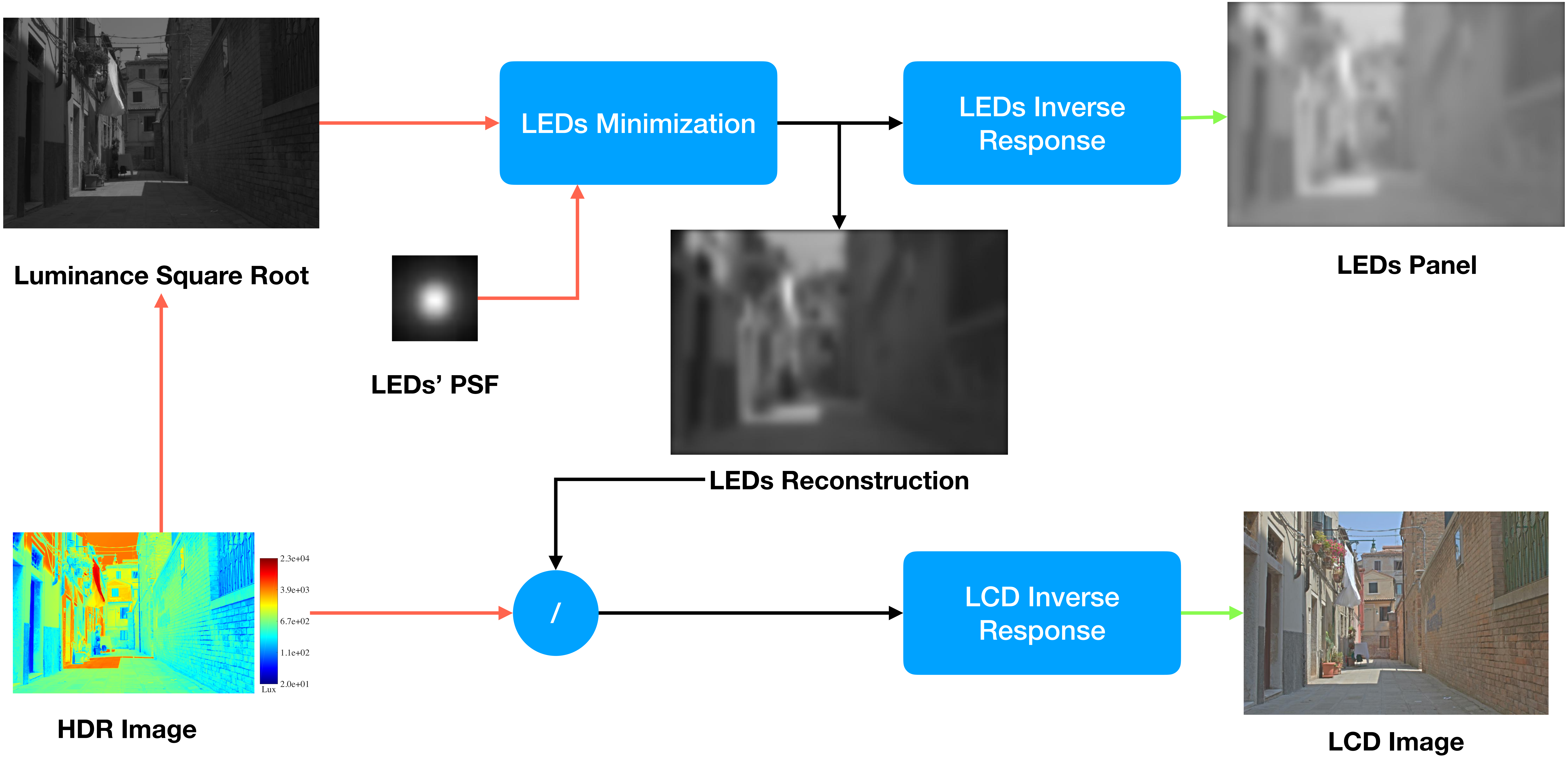
LED-based HDR Monitors: PSF



LED-based HDR Monitors: PSF



Native Visualization: HDR Monitors



HDR Imaging:
Metrics

Image Quality Metrics: with Reference

- A probability map; each pixel has the probability of being detected when compared to the reference by a viewer.
- Q predictor value in the range [0,100]; the higher the better.

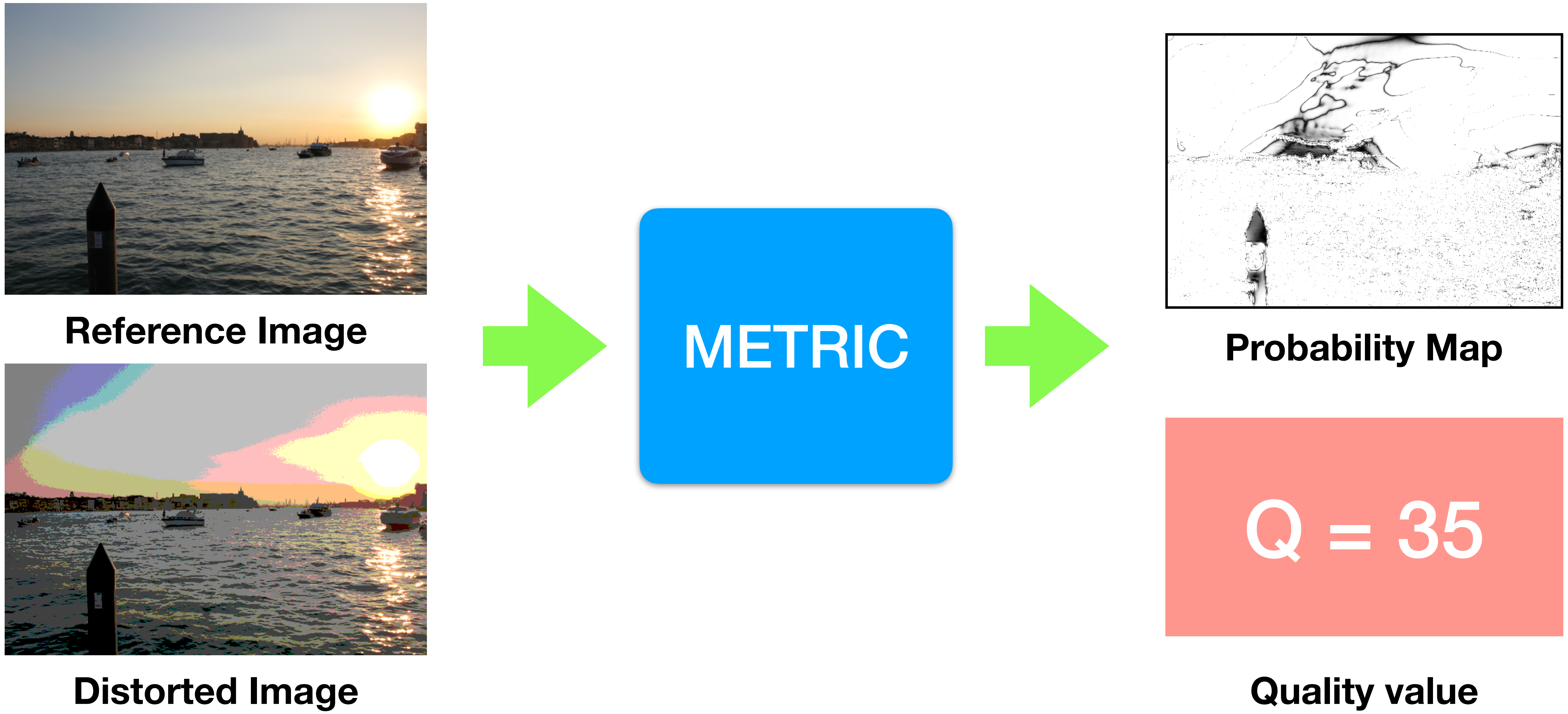
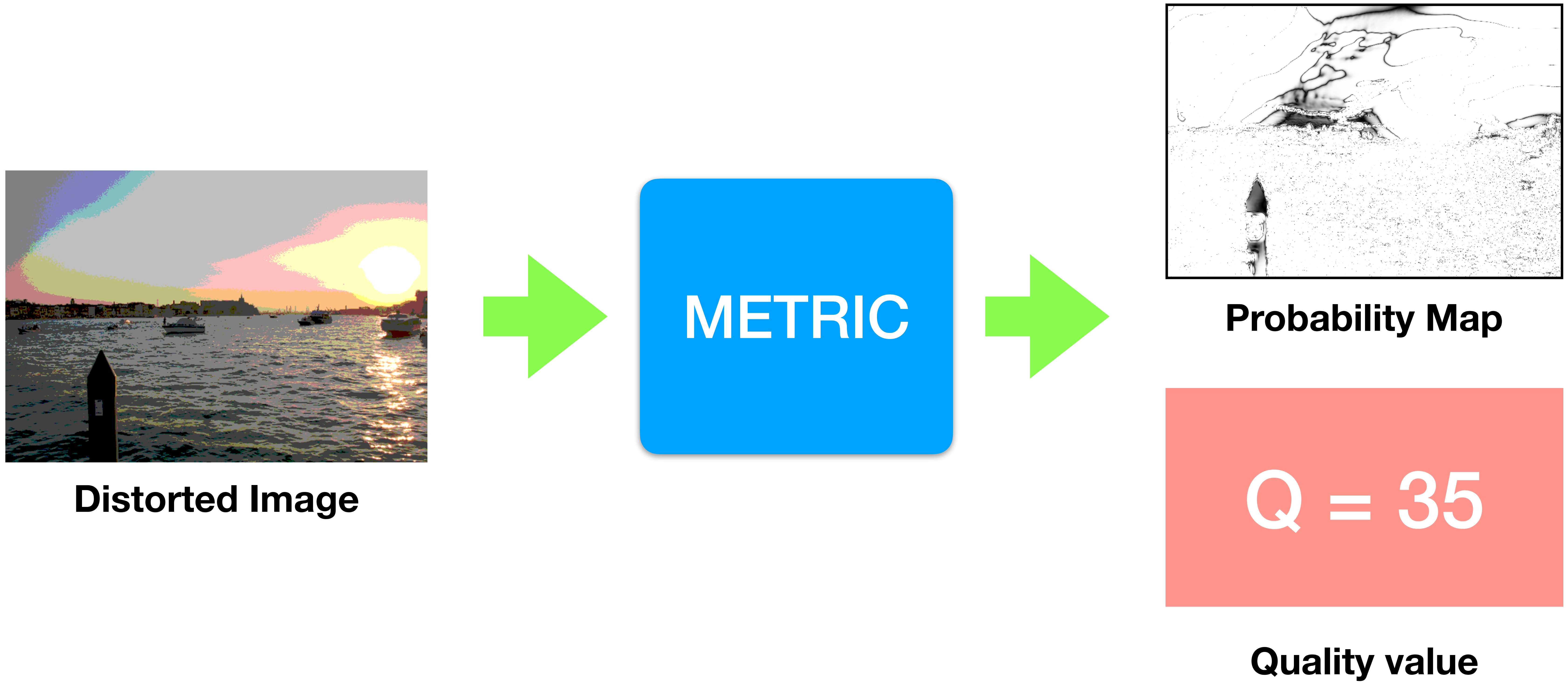


Image Quality Metrics: No Reference

- A probability map; each pixel has the probability of being detected when compared to the reference by a viewer.
- Q predictor value in the range [0,100]; the higher the better.



Metrics for HDR Applications

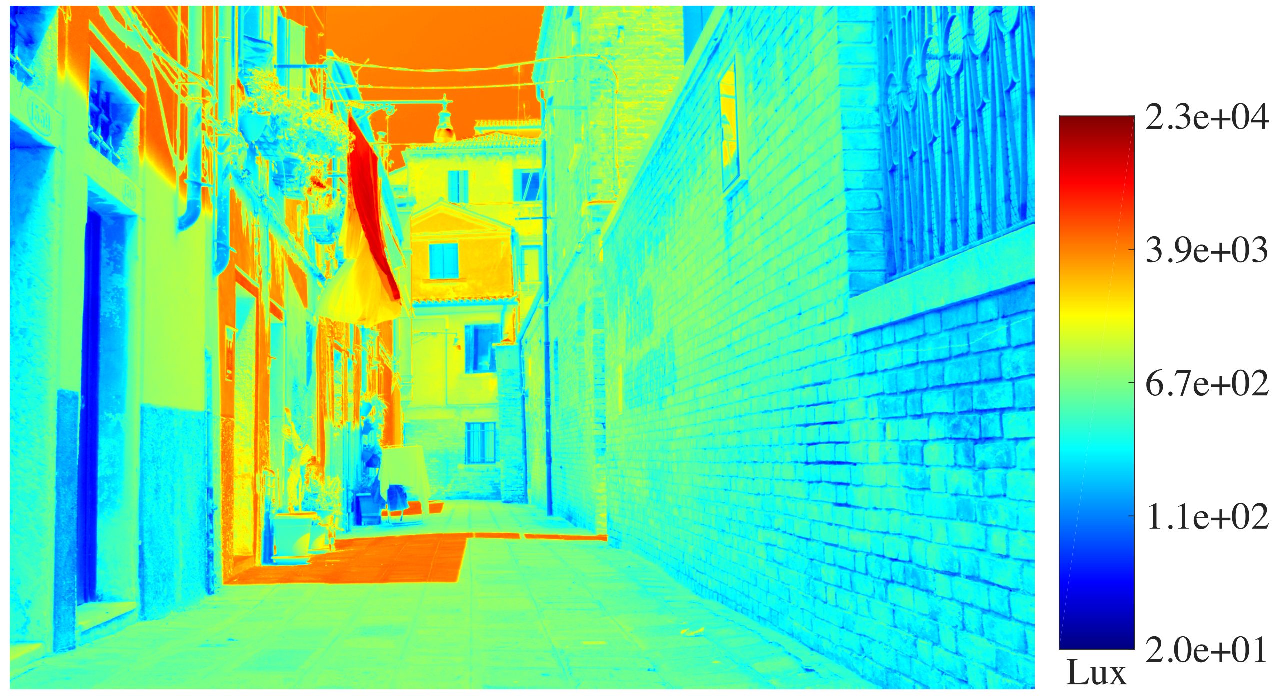
- HDR-VDP 2.2/3.0.6/DRIM:
  - They are reliable metrics for the general case:
    - HDR vs HDR; HDR vs SDR; etc.
  - Computational cost is demanding.
  - **A reference is required!**
- TMQI and TQMI-II:
  - Limited for comparing HDR vs SDR for tone mapping.
  - **A reference is required!**

HDR Open Problems: Acquisition

HDR Problems:
Merging Exposures in Dynamic Scenes

Stack of 8-bit images

MERGE

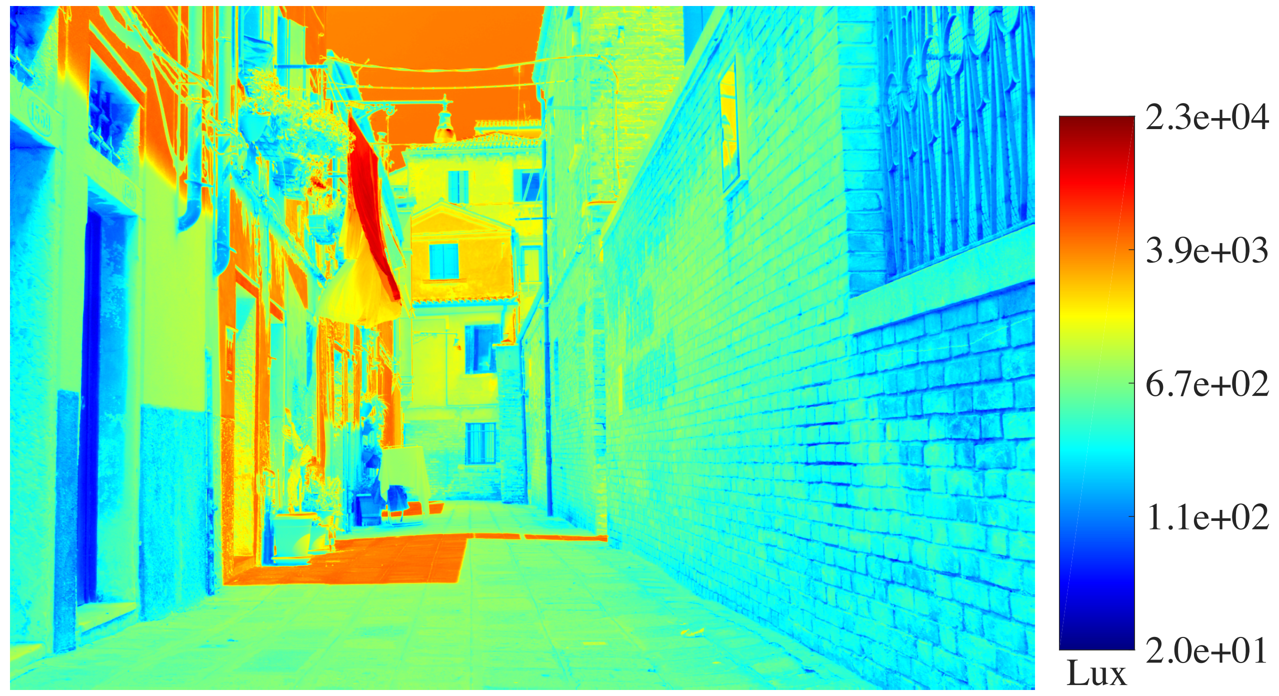


Scene-referred HDR image

HDR Problems:
Merging Exposures in Dynamic Scenes

Stack of 8-bit images

MERGE

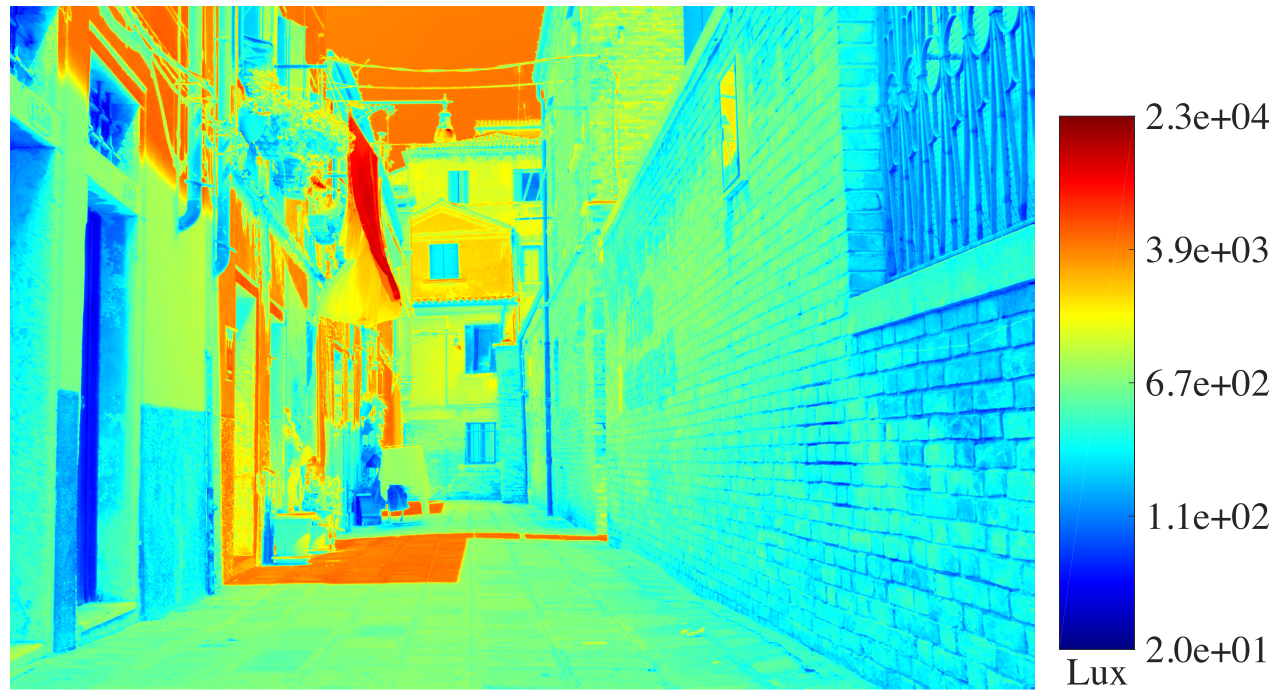


Scene-referred HDR image

HDR Problems:
Merging Exposures in Dynamic Scenes

Stack of 8-bit images

MERGE

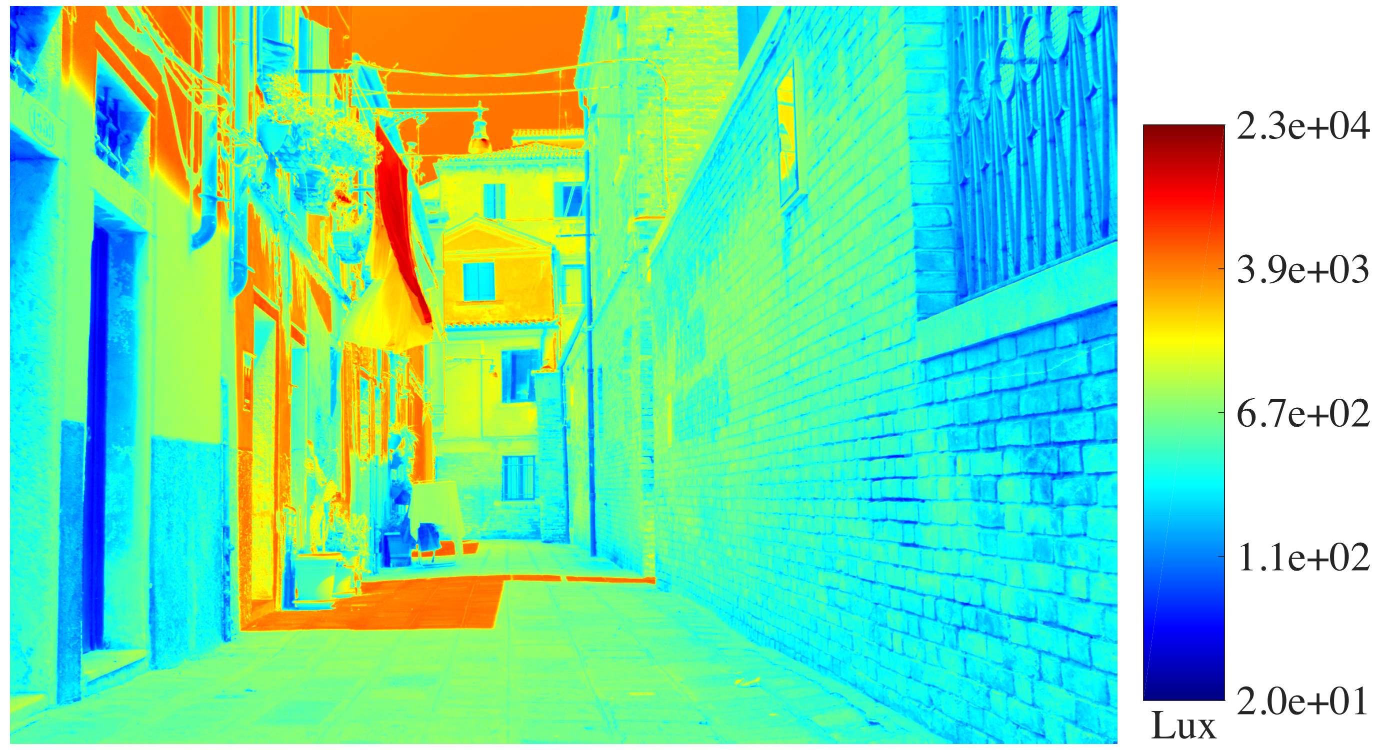


Scene-referred HDR image

HDR Problems:
Single-Image Acquisition / Inverse Tone Mapping

ITMO

MERGE



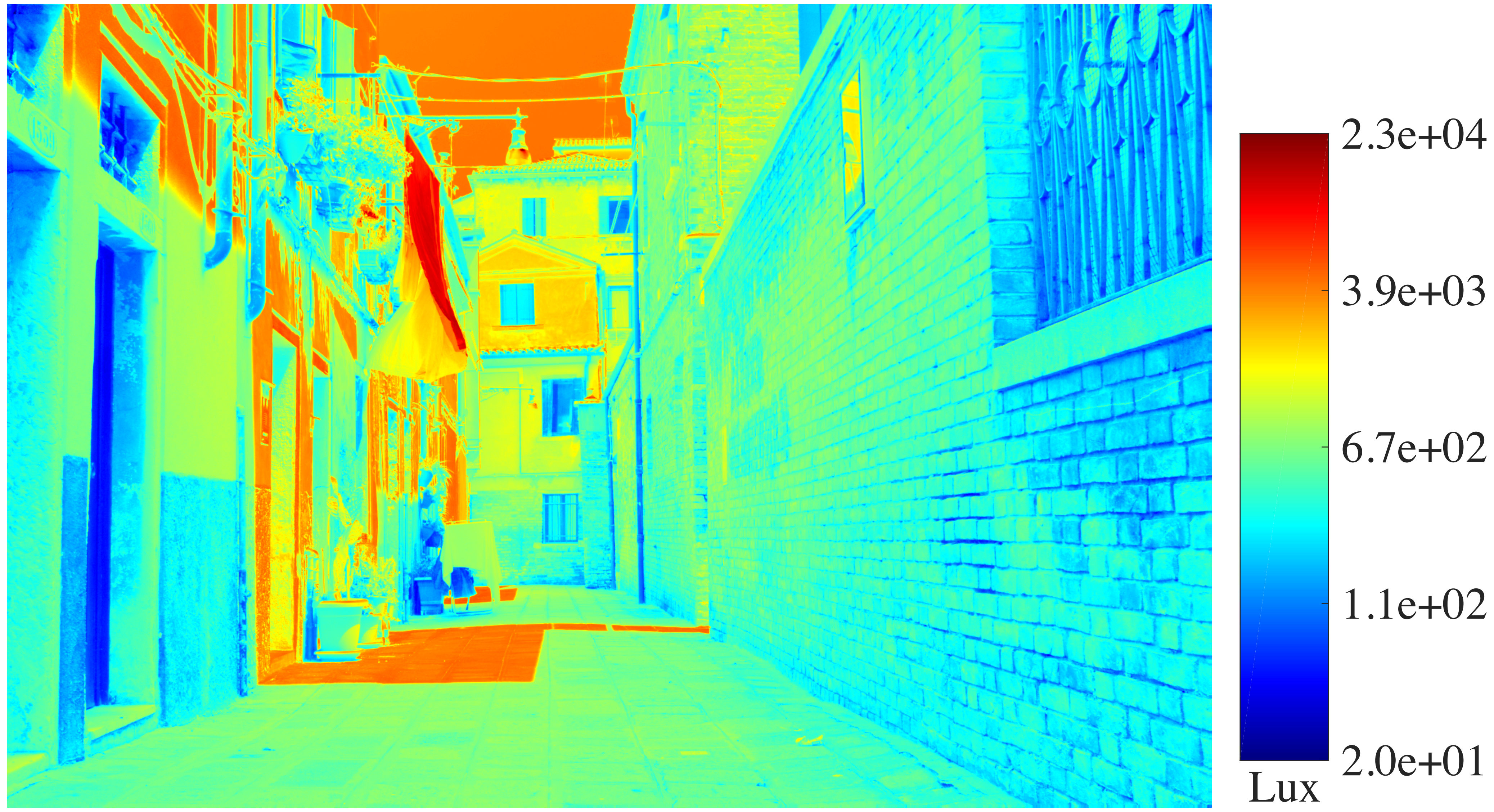
Single 8-bit images

Stack of 8-bit images

Scene-referred HDR image

HDR Open Problems: Visualization

HDR Problems: Tone Mapping



Scene-referred HDR image

TMO

8-bit Tone Mapped Image

HDR Open Problems:
How to Measure the Performance?

How to Measure Performance?

- How do we convert large experiments into metrics?
- Can we speed-up high quality but computationally expensive metrics?
- Can we have no-reference metric?

To Recap

To Recap

- In this tutorial, we will address how to use Deep Learning methods for:
  - Acquiring HDR content;
  - Display HDR images and videos;
  - Metrics for comparing HDR content.

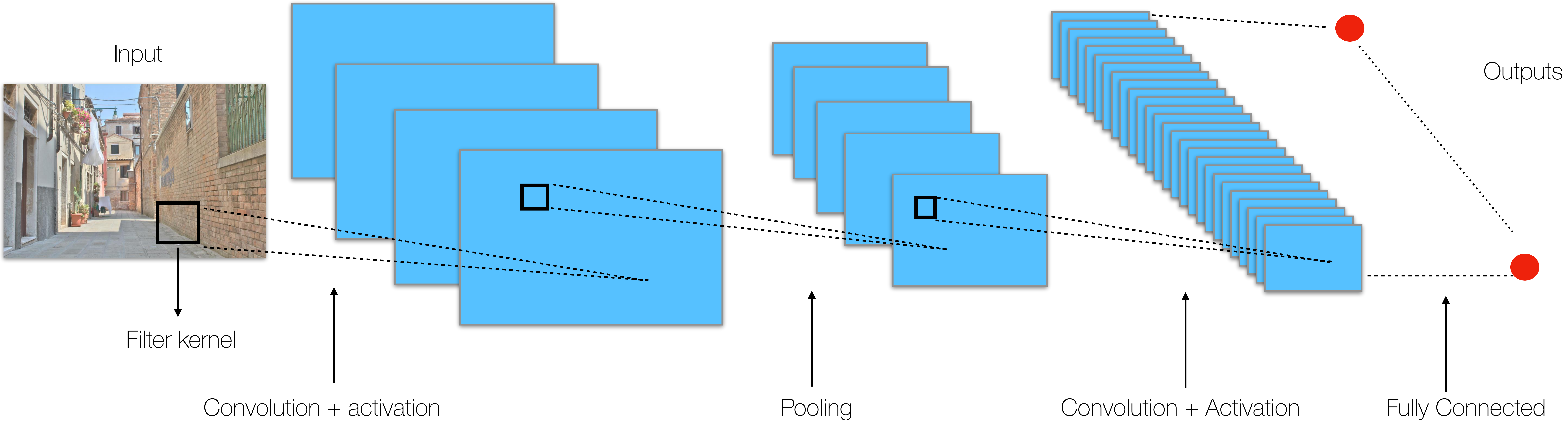
Questions?

Modern High Dynamic Range Imaging at the Time of Deep Learning

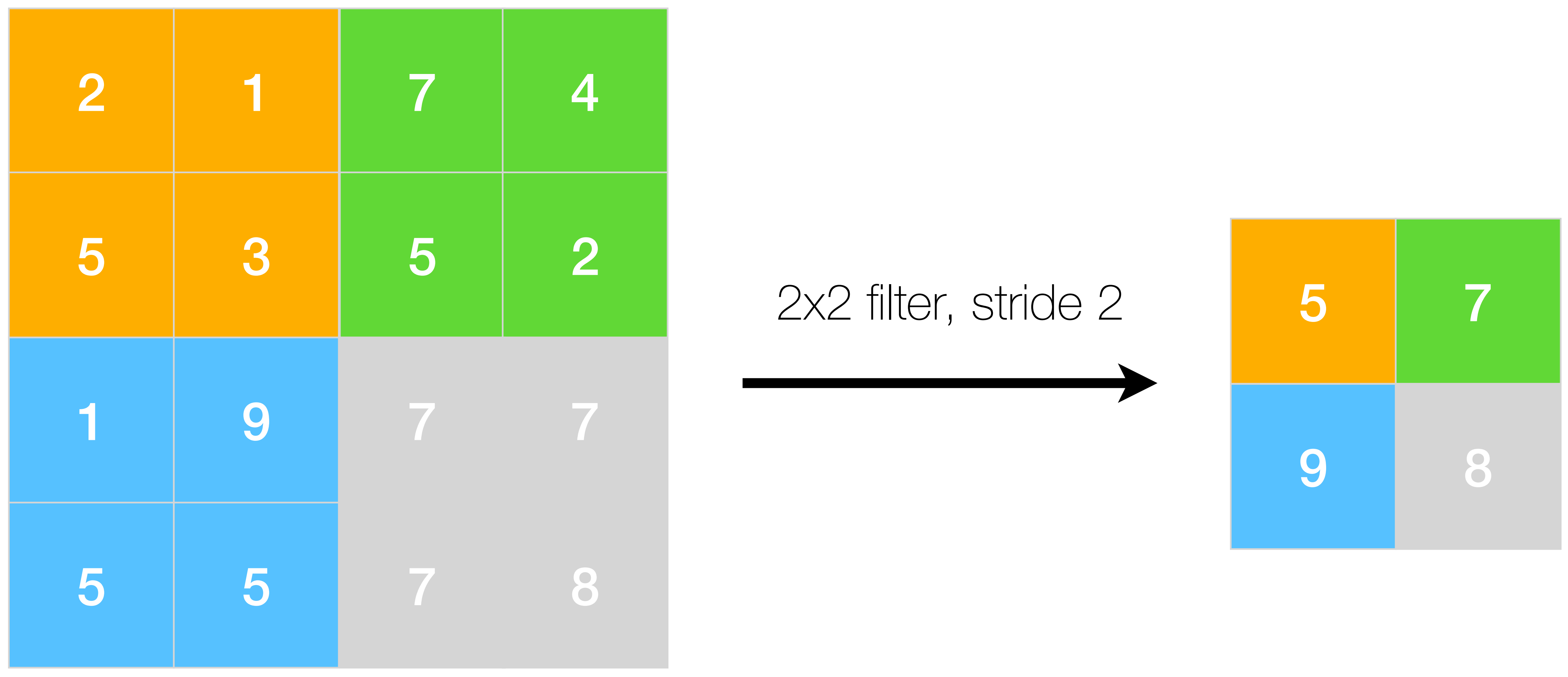
Main Deep Learning Architectures for Imaging

Francesco Banterle and Alessandro Artusi

Convolutional Neural Networks

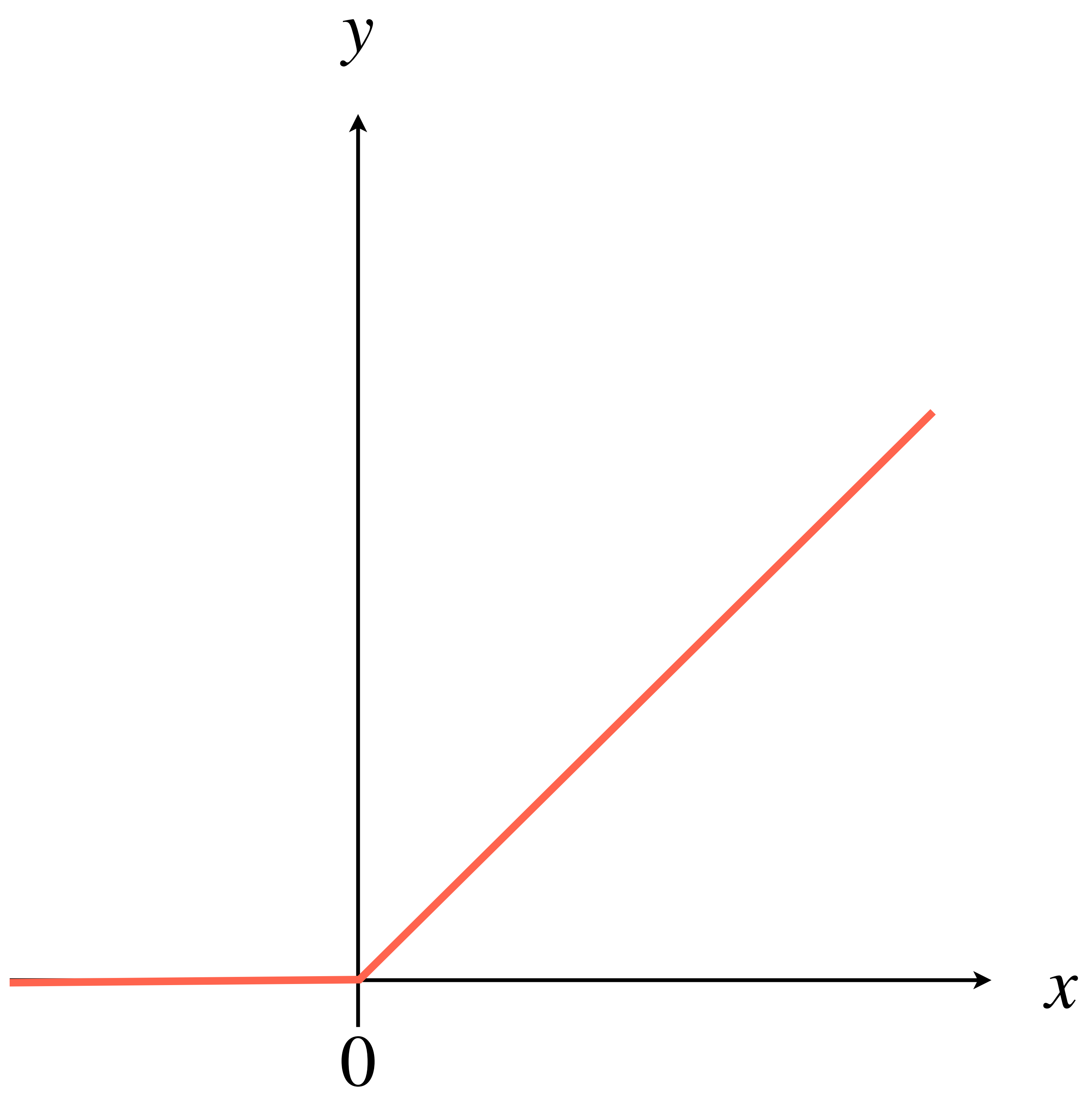


Pooling - Downsampling (e.g., max function)



- Controlling overfitting
- Reducing the number of parameters
- Memory footprint
- Reducing the number of computations

Activation Functions (layers) Categories - most used

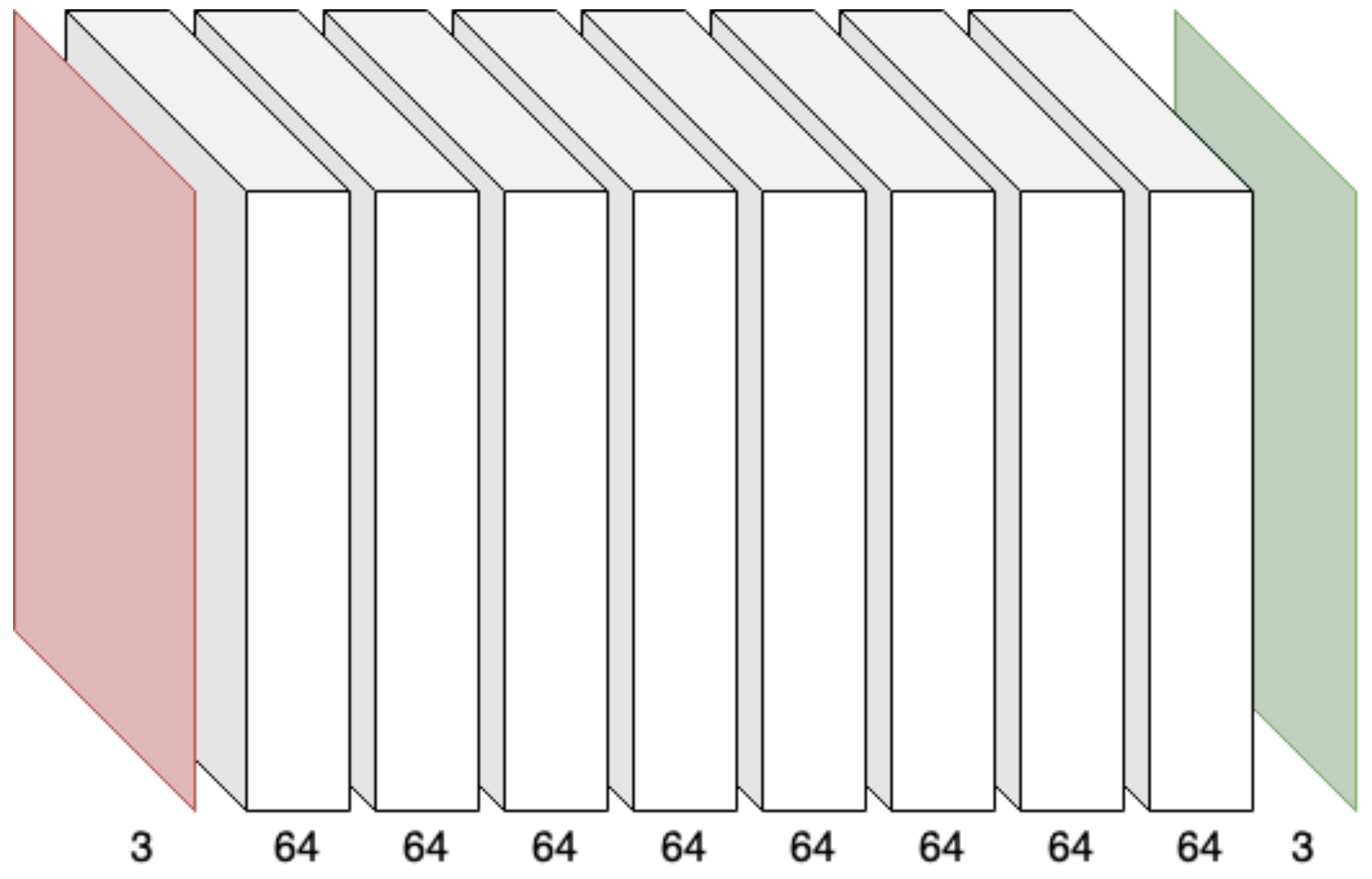


$$ReLU(x) = \max(0, a + x'b)$$

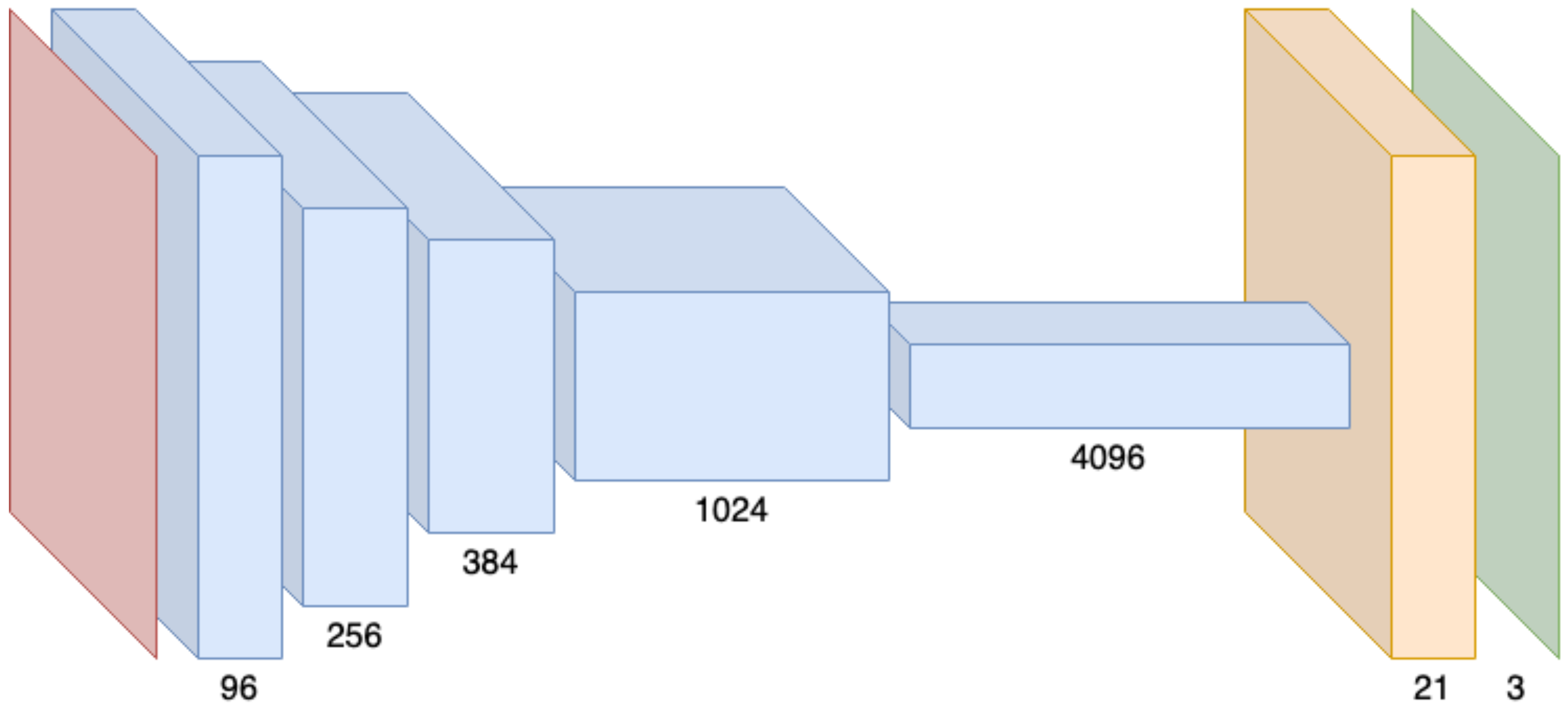
1. Ridge activation functions:
   1.1 Linear
   1.2 ReLU
   1.3 Logistic

2. Radial activation functions:
   2.2 Gaussian
   2.3 Multi-quadratics
   2.3 Polynomials

Fully Convolutional Neural Networks

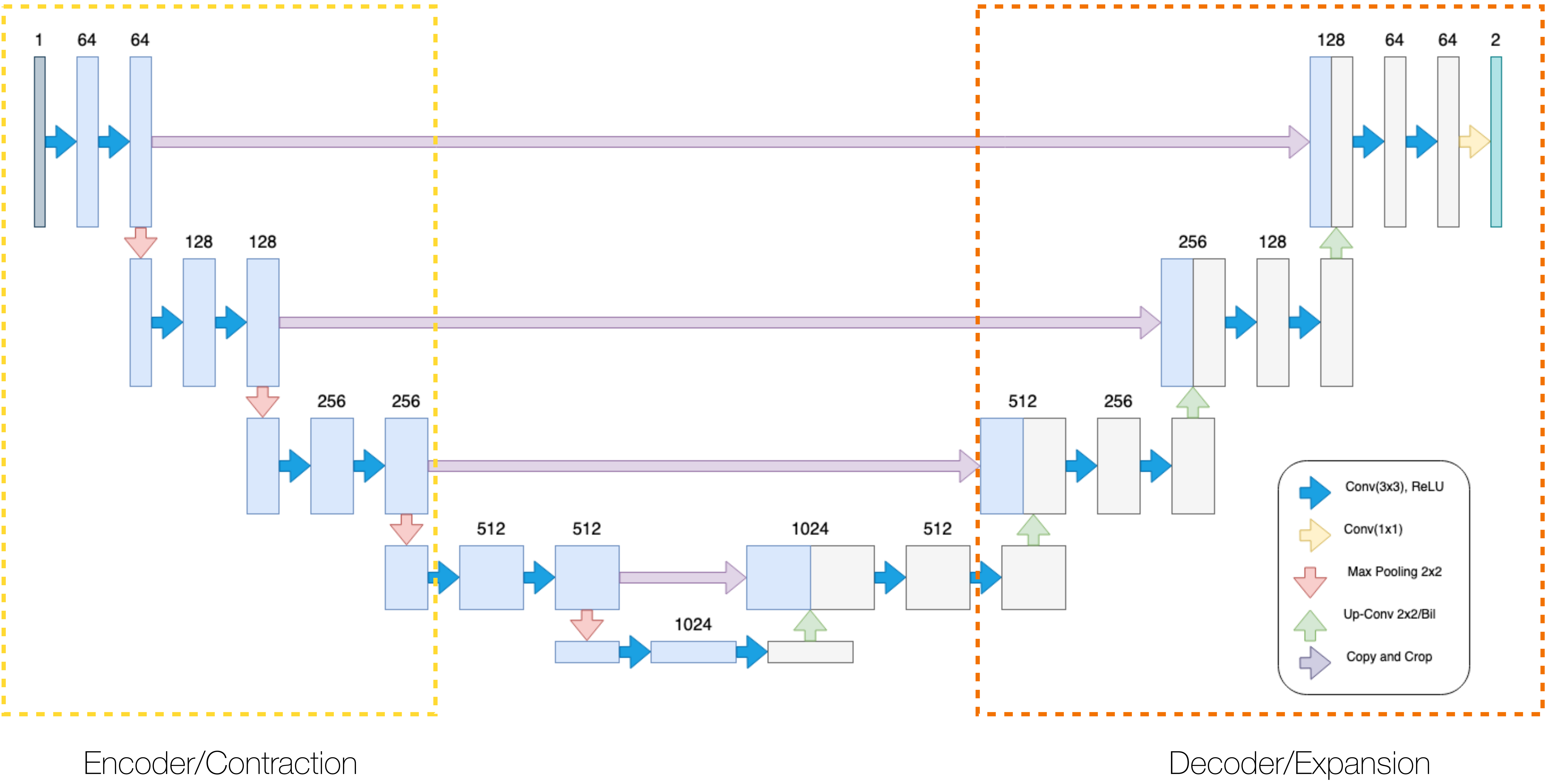


FCN with only convolutional layers (with activation functions).
Skip connections may be added to recover fine details.

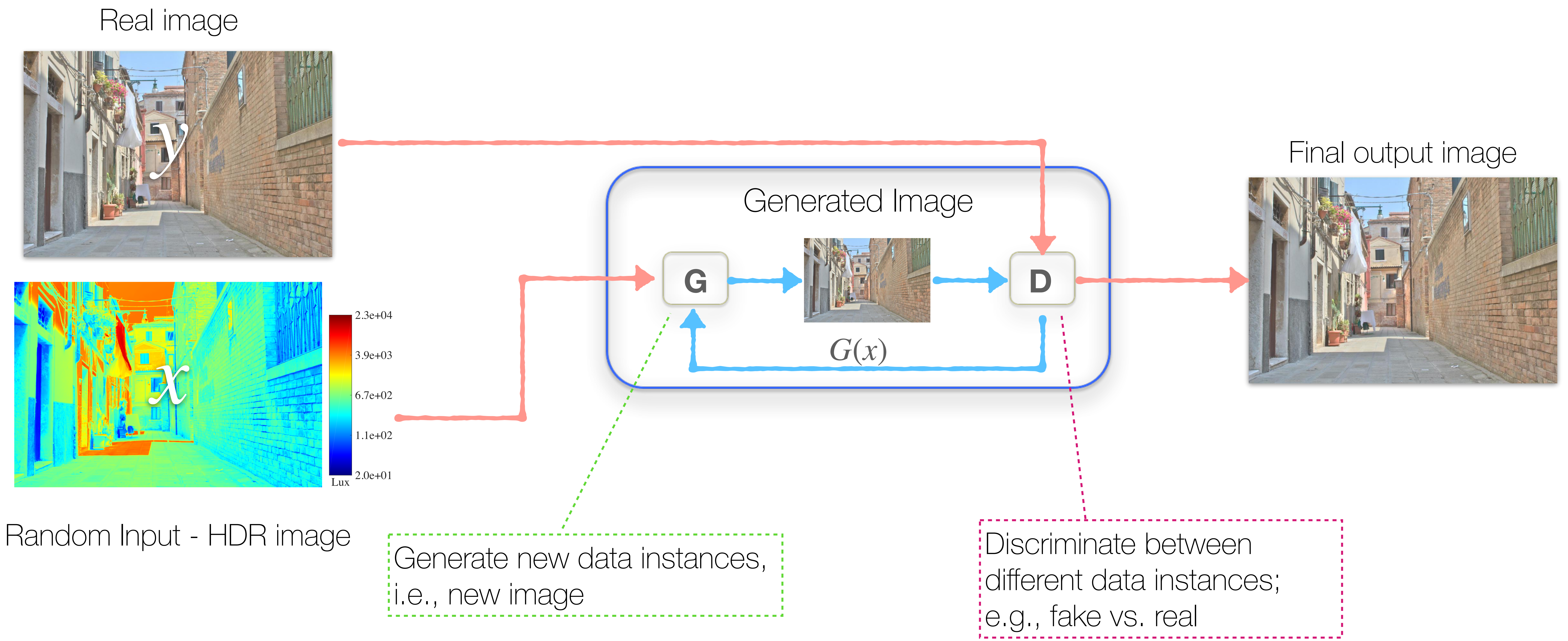


FCN with convolutions (with activation functions), downsampling, pooling, and upsampling.
Skip connections may be added to recover fine details.

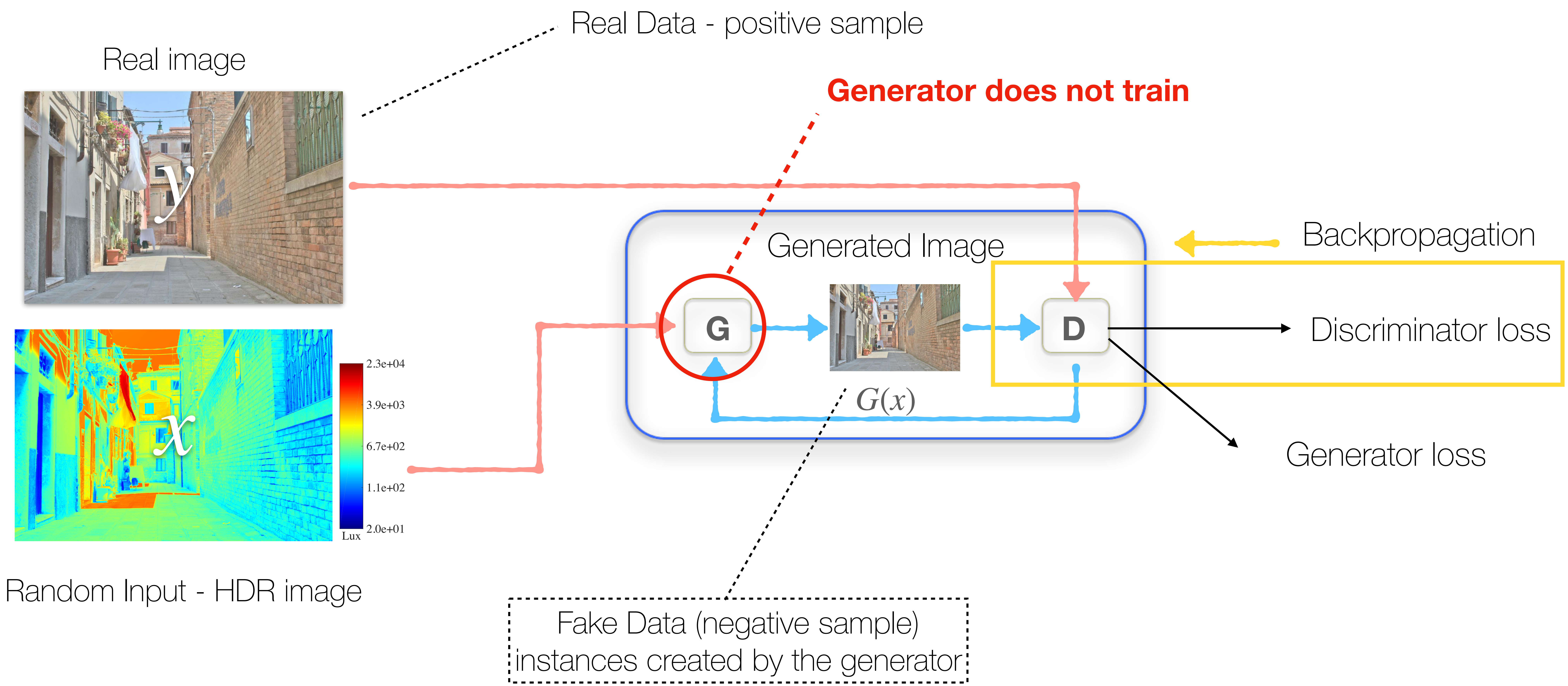
The U-Net


1
64
64
128
128
256
256
512
512
1024
1024
512
512
256
256
128
128
64
64
2
Conv(3x3), ReLU
Conv(1x1)
Max Pooling 2x2
Up-Conv 2x2/Bil
Copy and Crop
Encoder/Contraction
Decoder/Expansion

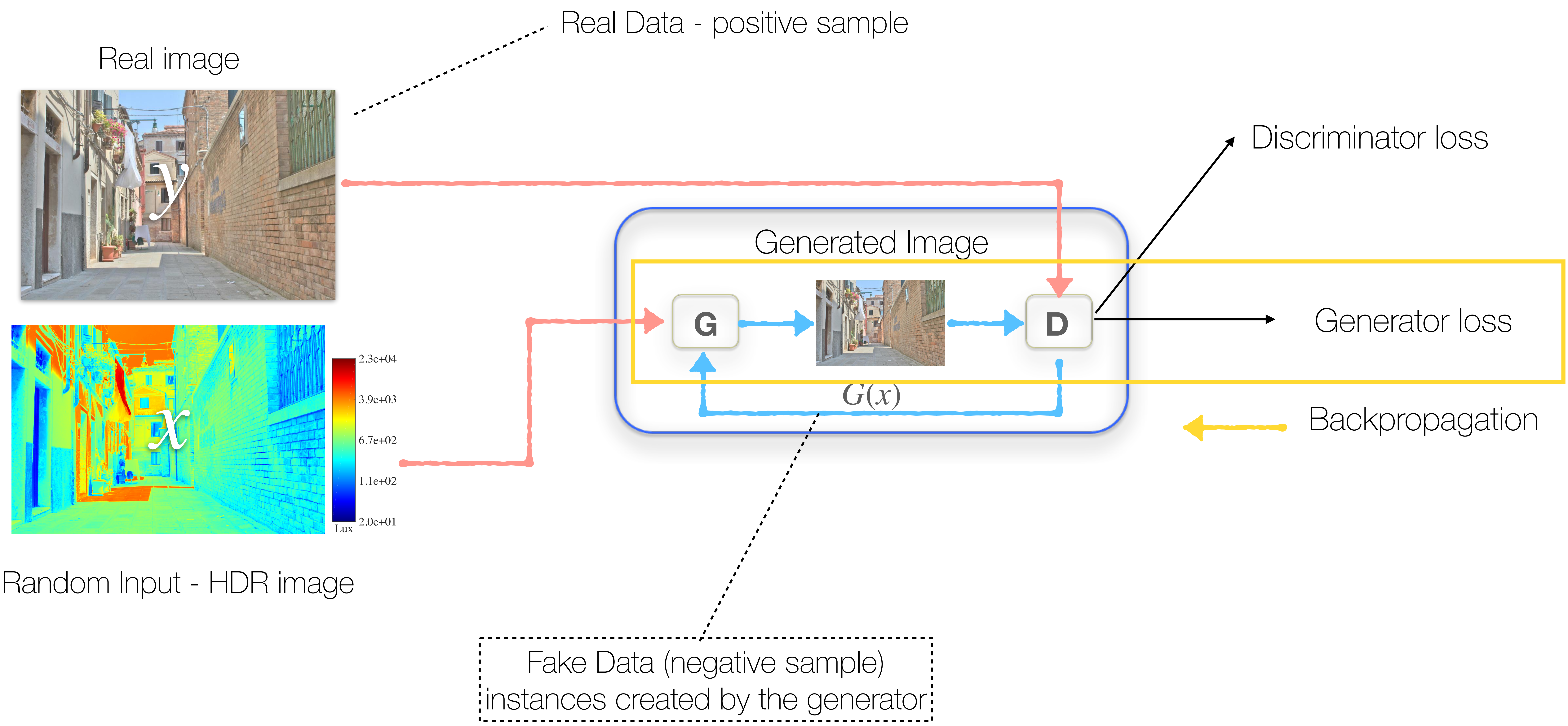
Generative Adversarial Networks (GANs)



GANs: Backpropagation in the Discriminator



GANs: Backpropagation in the Generator



GANs: Loss Function - e.g., Minimax loss

$$L_{GAN}(G, D) = \underbrace{\mathbb{E}_y[\log D(y)]}_{\text{Discriminator loss}} + \underbrace{\mathbb{E}_x[1 - \log D(G(x))]}_{\text{Generator loss}}$$

$D(y)$ = discriminator estimated probability that the real data instance y is real

$\mathbb{E}_y$ = expected value over all the real y instances

$G(x)$ = generator instance output value when given random input/input image x

$D(G(x))$ = discriminator estimated probability that a fake instance is real

$\mathbb{E}_x$ = expected value over all fake generated instances

Modern High Dynamic Range Imaging at the Time of Deep Learning

Multiple Exposures Reconstruction

Francesco Banterle and Alessandro Artusi

Introduction

- HDR reconstruction from multiple-exposures:
  - If we don’t place the camera on a stable tripod the camera moves!
  - If we have wind or people, there will be movement!
  - All this means, we will have artifacts!

Introduction: Camera Movement

- What if we capture a stack of exposure images free-hand without a tripod?

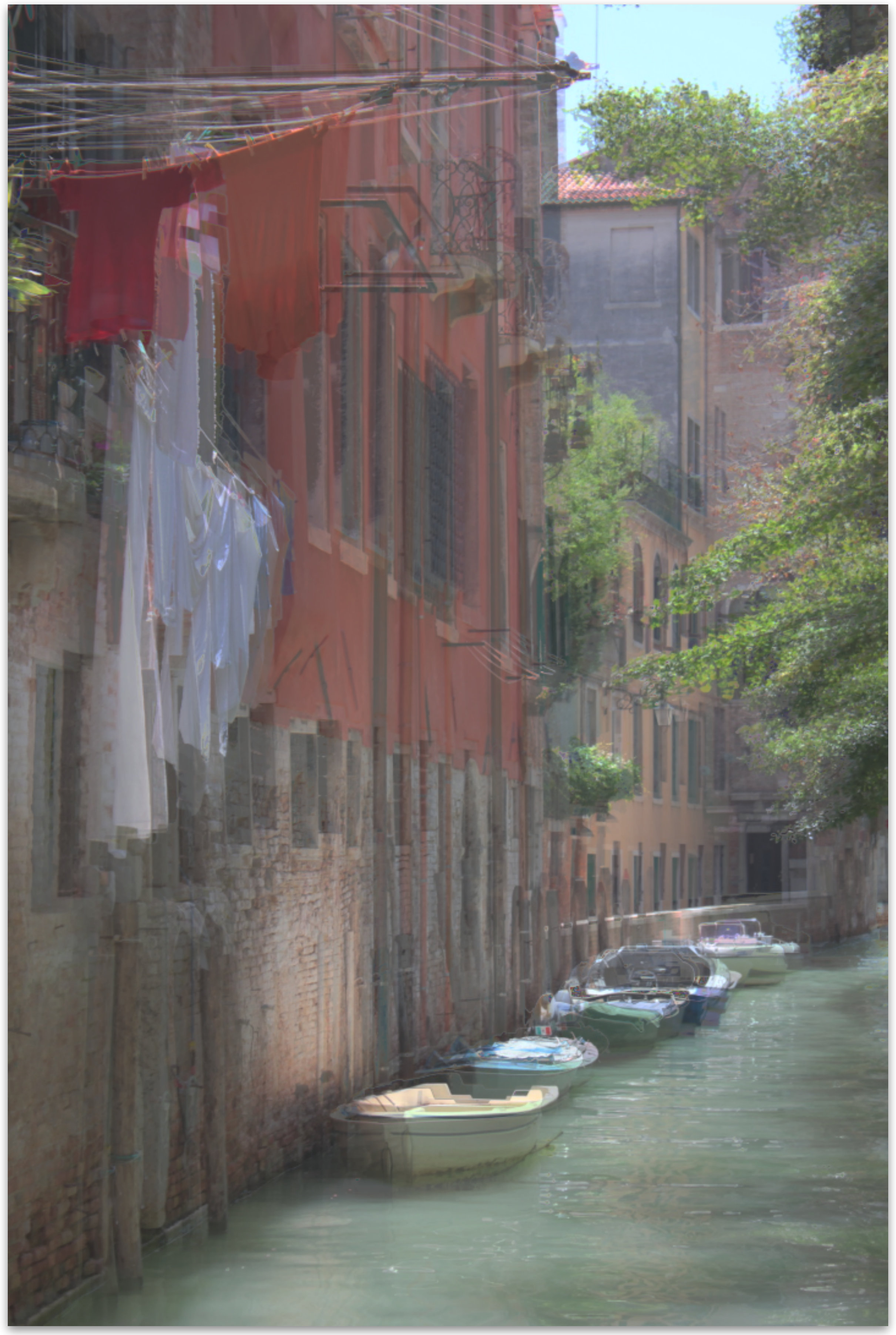
-2-stop

0-stop

+2-stop

Introduction: Camera Movement

Introduction: Camera Movement



Merged Stack and Tone Mapped

Introduction: Camera Movement

Introduction: Camera Movement

Merged Stack and Tone Mapped

Introduction: Camera Movement

Introduction: Camera Movement

- Typically, if we have **ONLY** camera movement, we can manage the merge:
  - We have only a single global movement.
- There are several robust algorithm to deal with such situations:
  - Greg Ward's MTB method.
  - Tomaszewska and Mantiuk's Homography algorithm.
  - Gallo's Multiple Homographies.

Introduction: Dynamic Scene

- What if we capture a stack of exposure images on a tripod in a dynamic scene?

-2-stop

0-stop

+2-stop

Introduction: Dynamic Scene

Introduction: Dynamic Scene

Merged Stack and Tone Mapped

Introduction: Dynamic Scene

Introduction: Dynamic Scene

Introduction: Dynamic Scene

Introduction: Dynamic Scene

Merged Stack and Tone Mapped

Introduction: Dynamic Scene

Introduction: Dynamic Scene

Introduction: Camera Movement

- Typically, if when the moving people/objects are small they can be fixed easily.
- There are several robust algorithm to deal with such situations:
  - Masks: Pece and Katuz.
  - Grandaos et al.
  - PatchMatch-based: Sen et al./Hu et al.

Datasets

Capturing Data: Kalantari’s Data

-2-stop

0-stop

+2-stop

Dynamic Stack

Video Courtesy of Jan Fröhlich - Stuttgart HDR Video Dataset

Capturing Data: Kalantari's Data

-2-stop

0-stop

+2-stop

Dynamic Stack

Static Stack

Video Courtesy of Jan Fröhlich - Stuttgart HDR Video Dataset

Capturing Data: Kalantari's Data

-2-stop

0-stop

+2-stop

Dynamic Stack | Static Stack | Training Stack

Video Courtesy of Jan Fröhlich - Stuttgart HDR Video Dataset

Images

- For each SDR image I_i, we know:
  - The CRF, $f(\cdot)$; i.e, we know its inverse $g(\cdot) = f^{-1}(\cdot)$;
  - The exposure time $t_i = \dfrac{\text{ISO}_i \cdot t'_i}{K \cdot A_i^2}$
    - t'_i: Shutter speed.
    - A_i: Aperture value.
    - ISO_i: ISO value.
    - $K \in [30.6, 13.4]$: a constant depending on the camera.

Images

- Typically, we work with "calibrated" SDR image H_i:

$$H_i = \frac{g(I_i)}{t_i}$$

- In many works, the CRF is assumed to be $f(x) = x^{\frac{1}{2.2}}$.
- Therefore, we have:

$$H_i = \frac{I_i^{2.2}}{t_i}$$

Images: Patches and Augmentations

- All methods are trained on patches of different size: 40×40, 256×256, 512×512.
- Patches may be create with or without overlap.
- We have different augmentations:
  - Rotation, Flips, etc.
  - Swapping color channels [Kalantari et al. 2017]

Preprocessing

- The problem can be “simplified” by using classic approach for a first alignment:
  - **Homography alignment** introduced by Wu et al. 2018;
  - **Optical flow alignment** introduced by Kalantari et al. 2017.
- This initial alignment reduces blur.
- Typically, it matches the background well:
  - Local mismatches are left.

HDR Image Datasets

Dataset Name	#Images	#Resolution	Calibrated	Website
Kalantari Dataset	74	1.5MPix	Uncalibrated	https://cseweb.ucsd.edu/~viscomp/projects/SIG17HDR/
Tursun Dataset	17	0.6Mpix	Uncalibrated	https://user.ceng.metu.edu.tr/~akyuz/files/eg2016/index.html

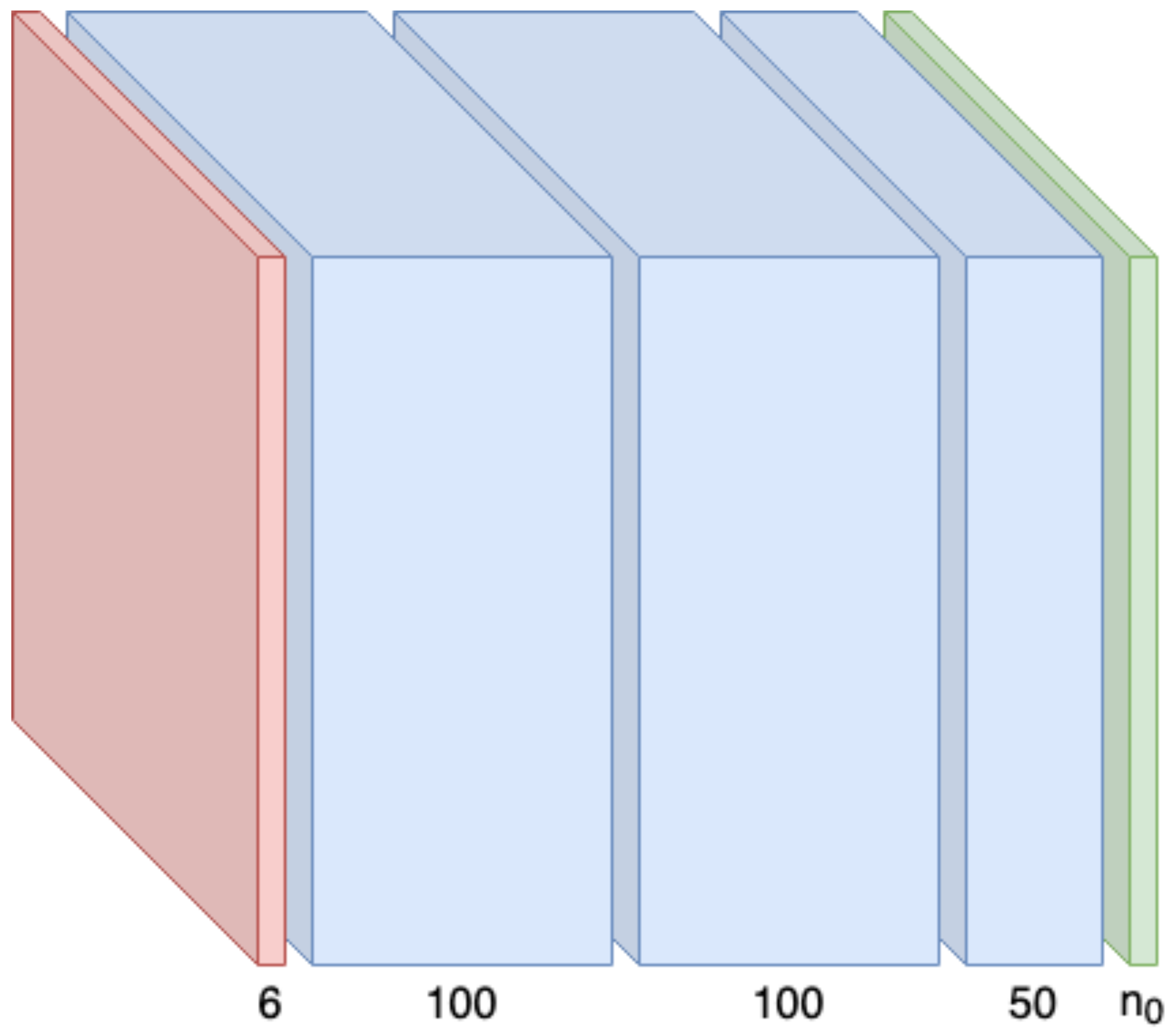
HDR Video Datasets

Dataset Name	#Videos	#Resolution	Length	FPS	Color Space	Format	Website
Stuttgart HDR Dataset	33	1920×1080	13s-100s	24/25	REC709	Floating Point	https://www.hdm-stuttgart.de/vmlab/projects/
UBC HDR Video Dataset	10	2048×1080	7s-10s	30	REC709	Floating Point	http://dml.ece.ubc.ca/data/DML-HDR/
LIVE HDR Video Quality Assessment Database	31 (310 at different bit-rates)	0.32Mpix	3s-10s	50/60	BT2020	HDR10	https://live.ece.utexas.edu/research/LIVEHDR/LIVEHDR_index.html
MPI HDR Video Dataset	2	0.3Mpix	24s-34s	24	REC709	Floating Point	https://resources.mpi-inf.mpg.de/hdr/video/
EBU HDR Video Dataset	10	3996×2160	10s-31s	50	BT2100	HLG	https://tech.ebu.ch/testsequences

End2End Architectures

Kalantari et al. 2017

- Kalantari et al. 2017 proposed a simple solution:
  - Optical Flow for the main alignment between exposures;
  - An end2end (a FCN) with ReLU in all layers except a sigmoid for the last layer:
    - Convolution varies in kernel size from large to small:
      - 7×7, 5×5, 3×3, and 1×1



Kalantari et al. 2017

- Kalantari et al. 2017 noted that the simple solution have some issues:
  - It is difficult to train; we need a huge dataset!
  - It does not fix alignment artifacts.
- The solution is to use the network to:
  - Compute Weights.
  - Refine images.

Kalantari et al. 2017

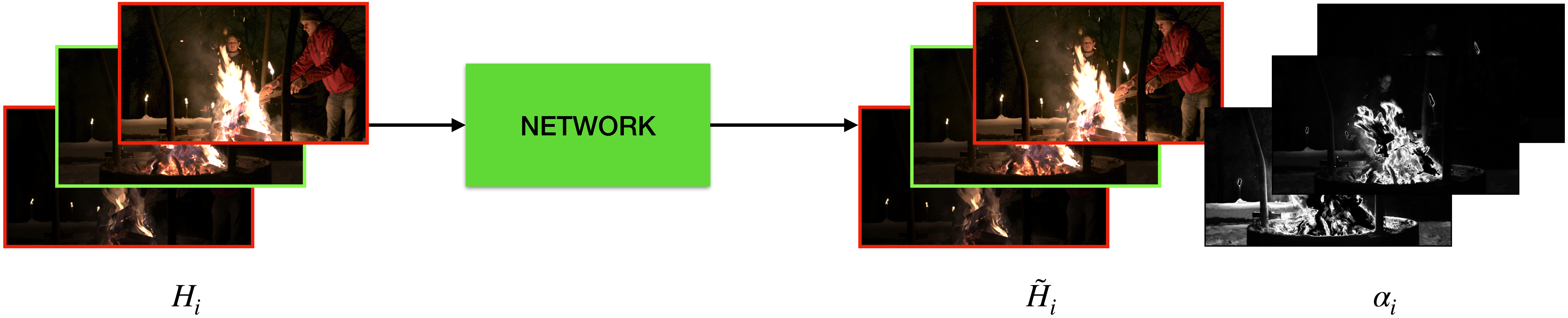
- Weight Estimator:
  - The shown architecture is used to compute the per-pixel weights, $\boldsymbol{\alpha}$, to obtain the estimated HDR image $\hat{H}$:

$$\hat{H} = \frac{\sum_i \alpha_i \cdot H_i}{\sum_i \alpha_i}$$

- Refined Images:
  - The network also refines the alignment obtaining new improved images $\tilde{H}_i$:

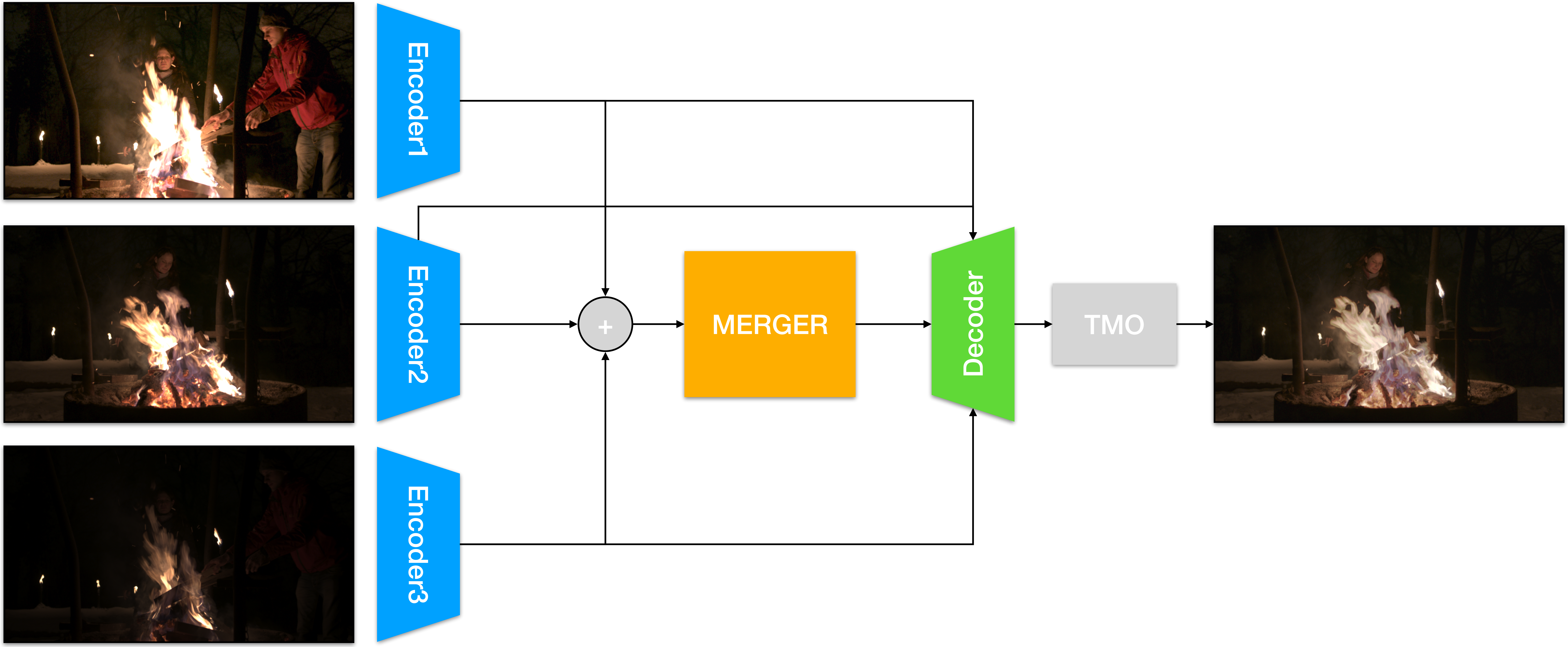
$$\hat{H} = \frac{\sum_i \alpha_i \cdot \tilde{H}_i}{\sum_i \alpha_i}$$

Kalantari et al. 2017



Video Courtesy of Jan Fröhlich - Stuttgart HDR Video Dataset

Encoder-Decoder - Wu et al. 2018



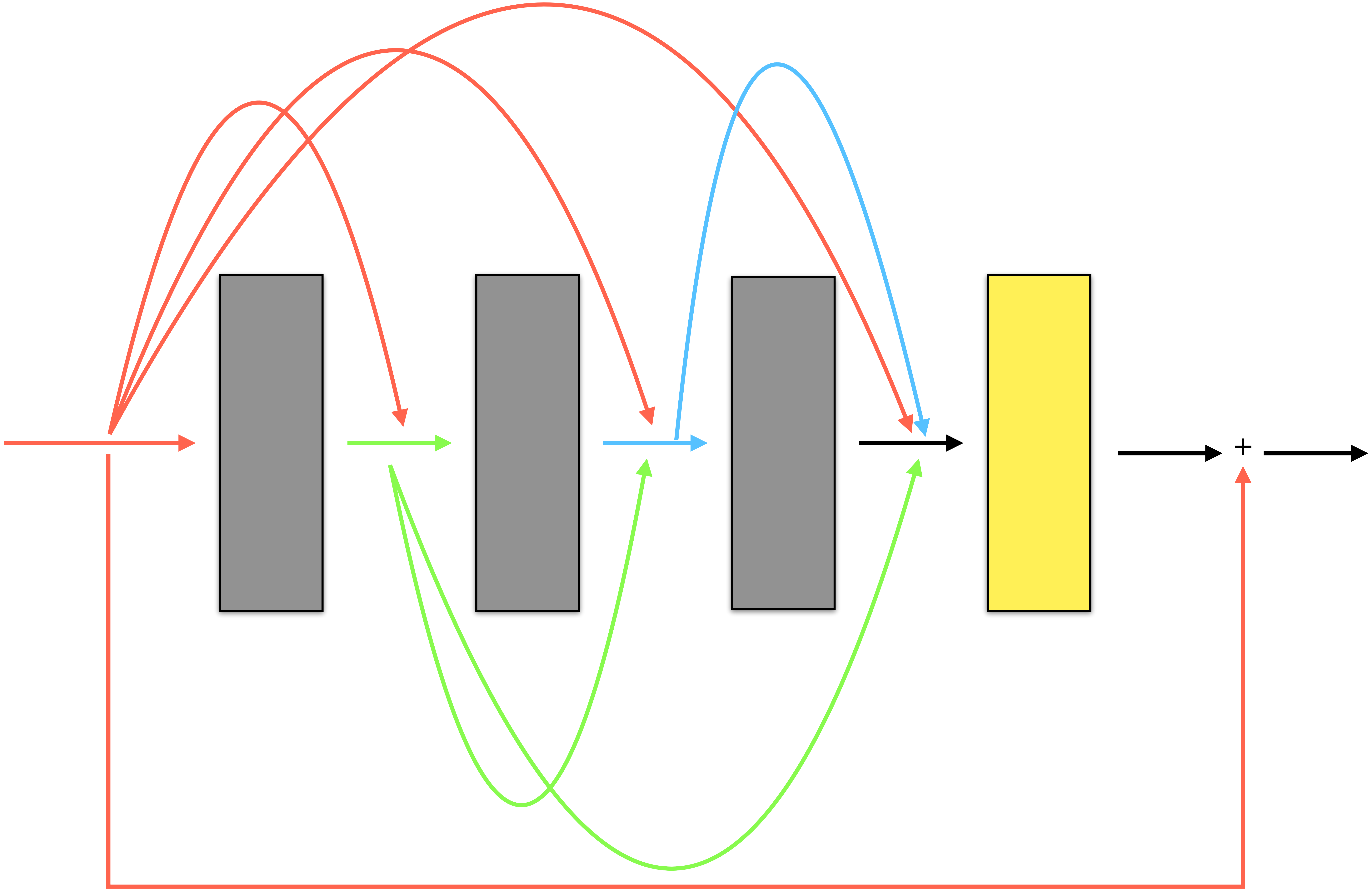
Video Courtesy of Jan Fröhlich - Stuttgart HDR Video Dataset

Attention HDR

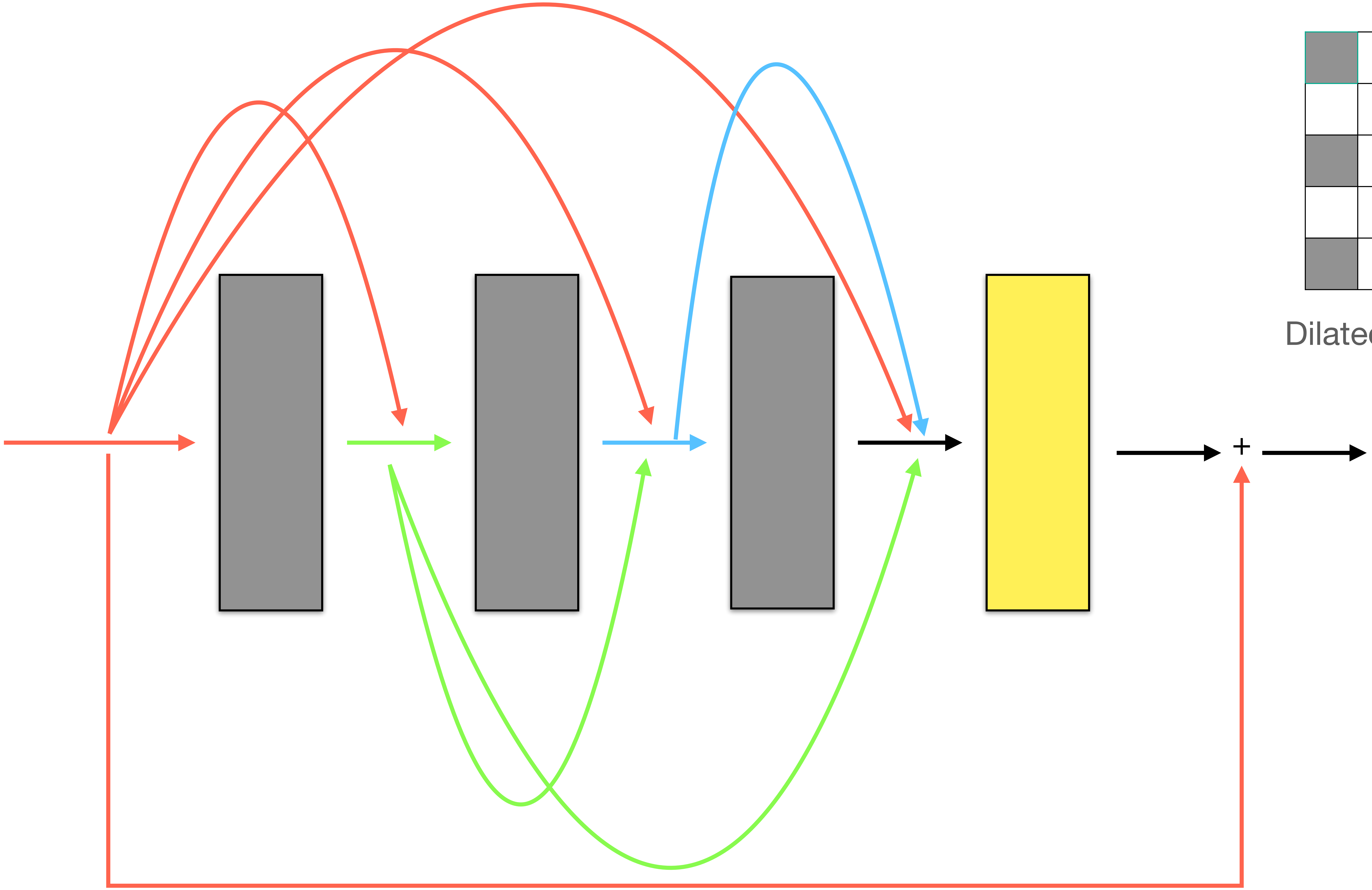
- Yan et al. 2019 introduces two blocks:
    - Attention Module:
        - The attention is computed on low level features.
        - The attention is applied to features of images that are not the reference.
    - Residual Dense Blocks [Zhang et al. 2018] with dilated convolutions to have a larger receptive field.

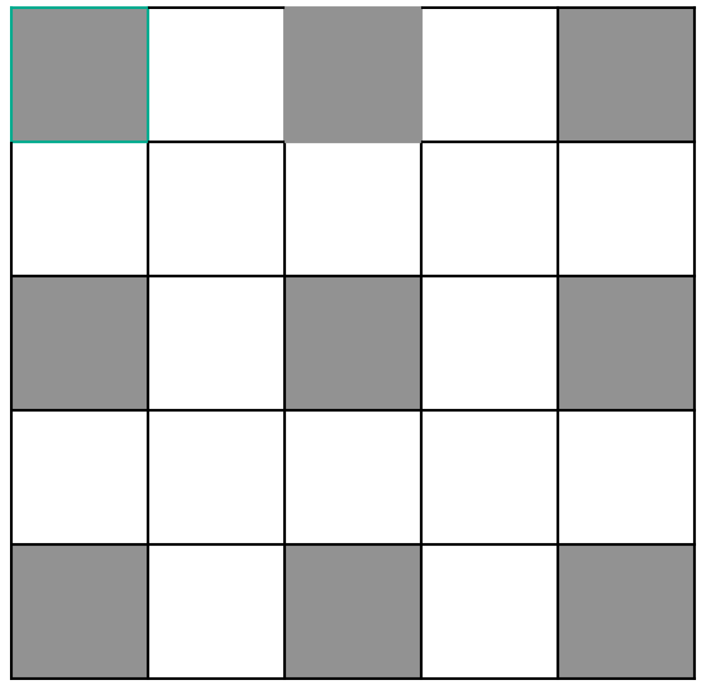
Attention HDR

Attention HDR



Attention HDR

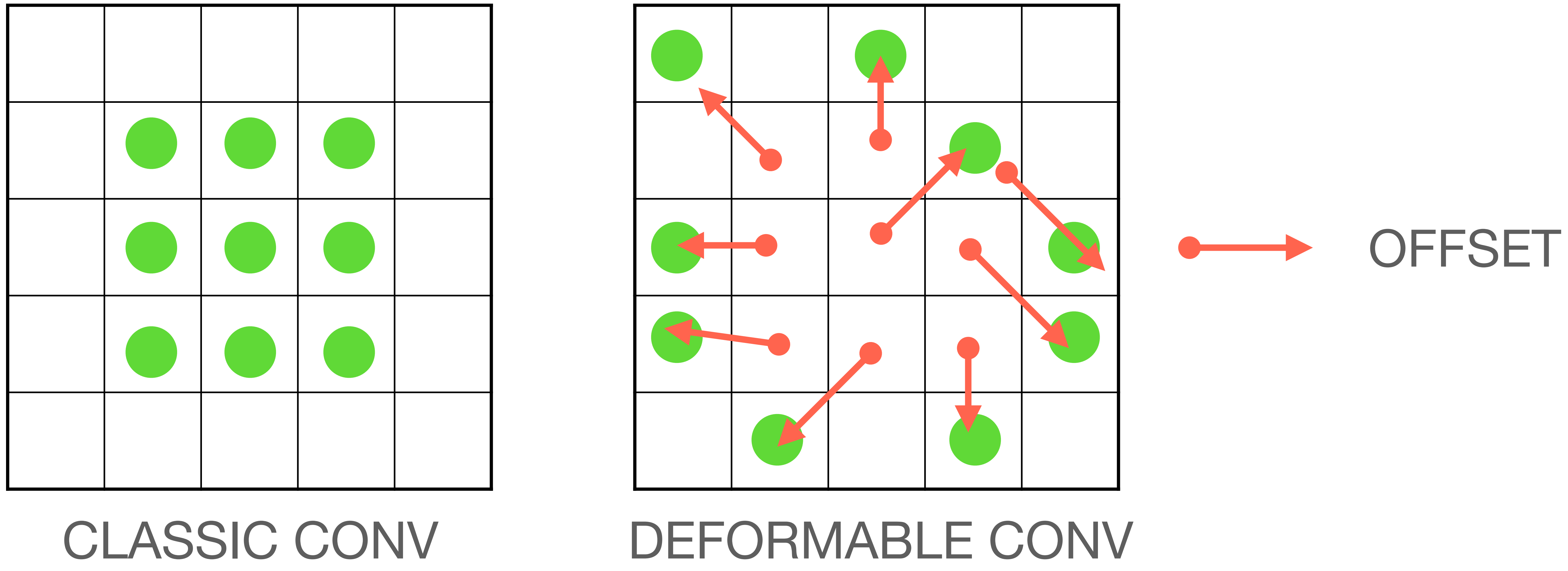




Dilated Convolutions

ADNet

- Liu et al. 2021, similarly to Pu et al. 2020, proposed for NTIRE 2021 a network based on two main blocks:
  - Attention computed using the reference, similar to Yan et al. 2019.
  - Pyramid, Cascade and Deformable (PCD) module by Wang et al. 2019:
    - PCD is applied at the feature level of the gamma-corrected images.
    - This module uses deformable convolutions

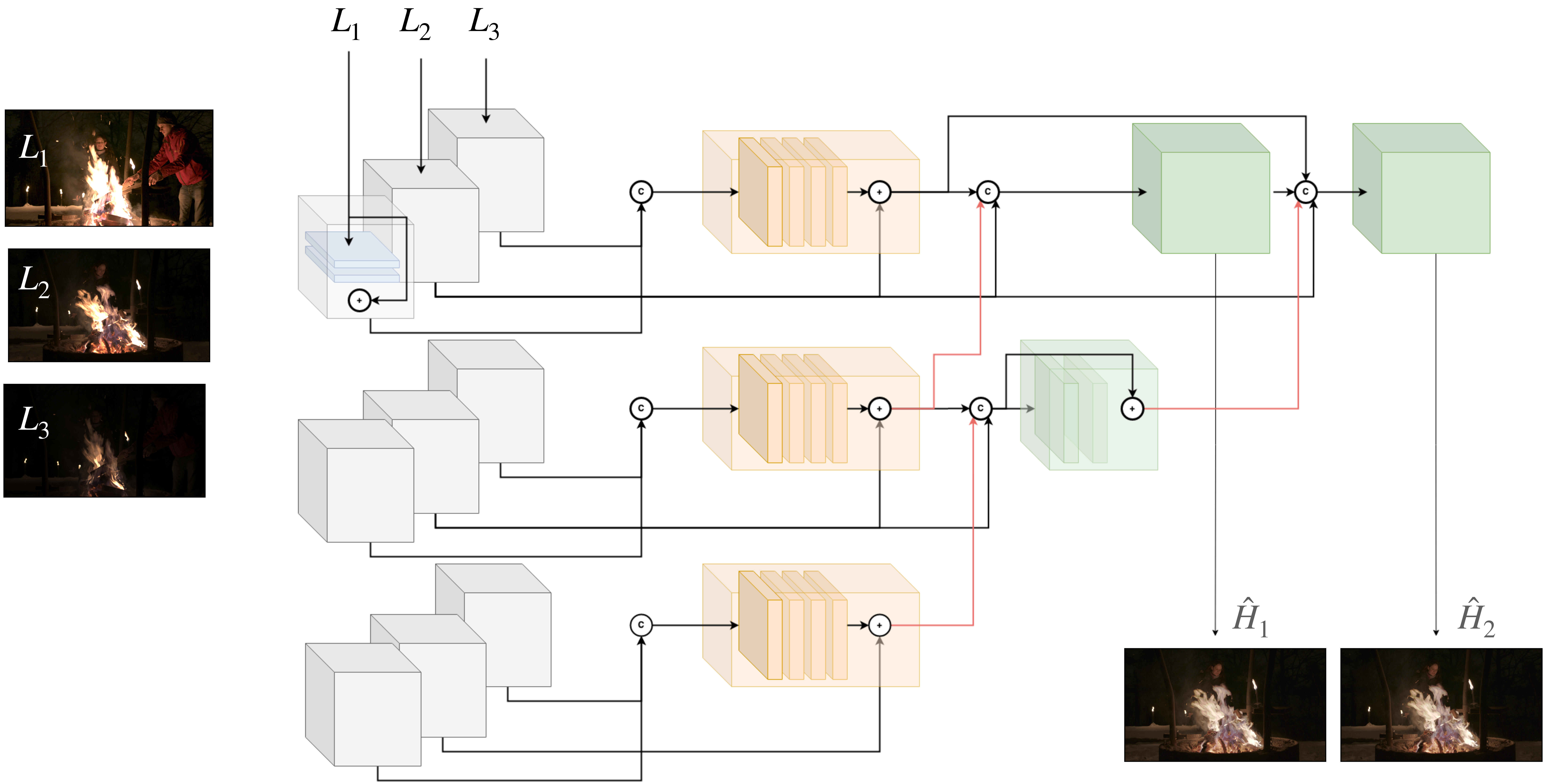


ADNet - PCD

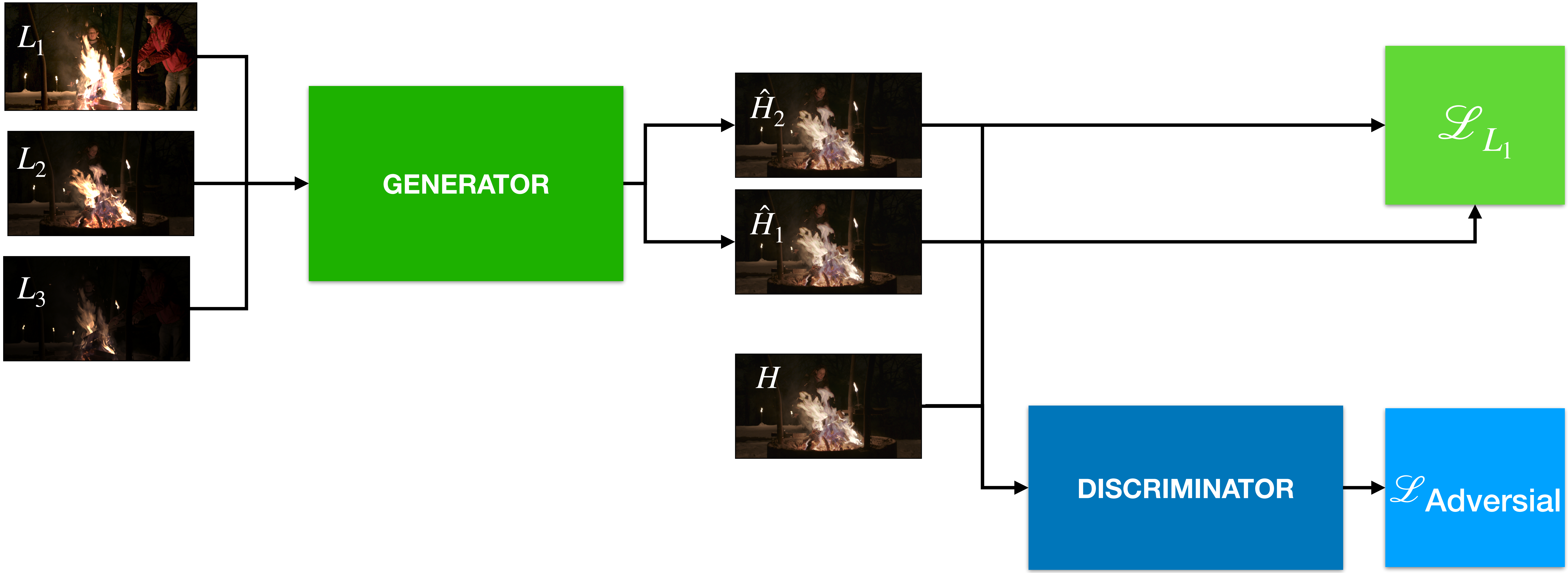

I_1
I_2
Aligned FEATURE I_2
DCONV
FEATURE I_1
FEATURE I_2
CONCATENATION
UPSAMPLE
DCONV

GAN Architectures

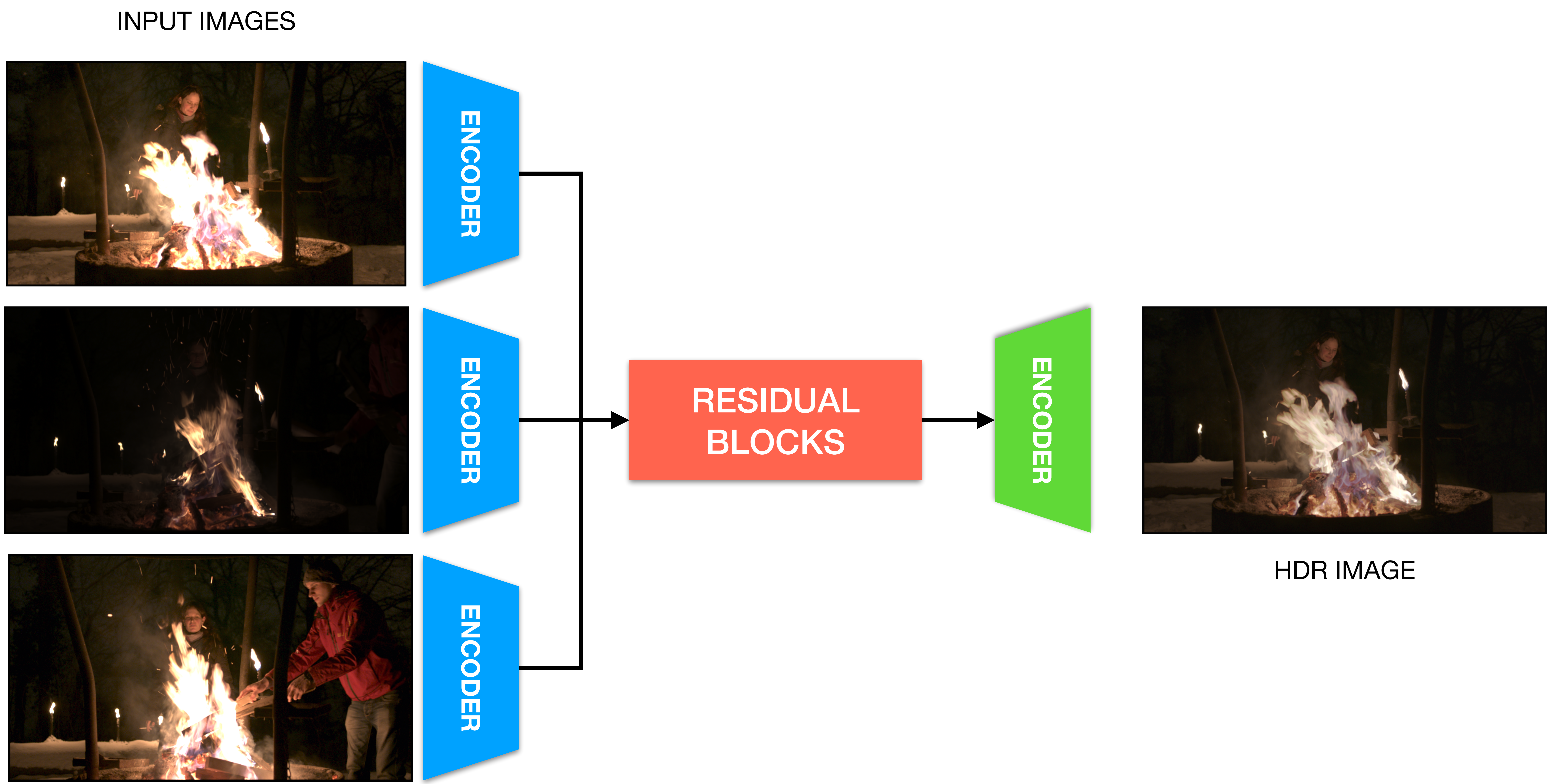
HDRGAN - Niu et al. 2021: Generator



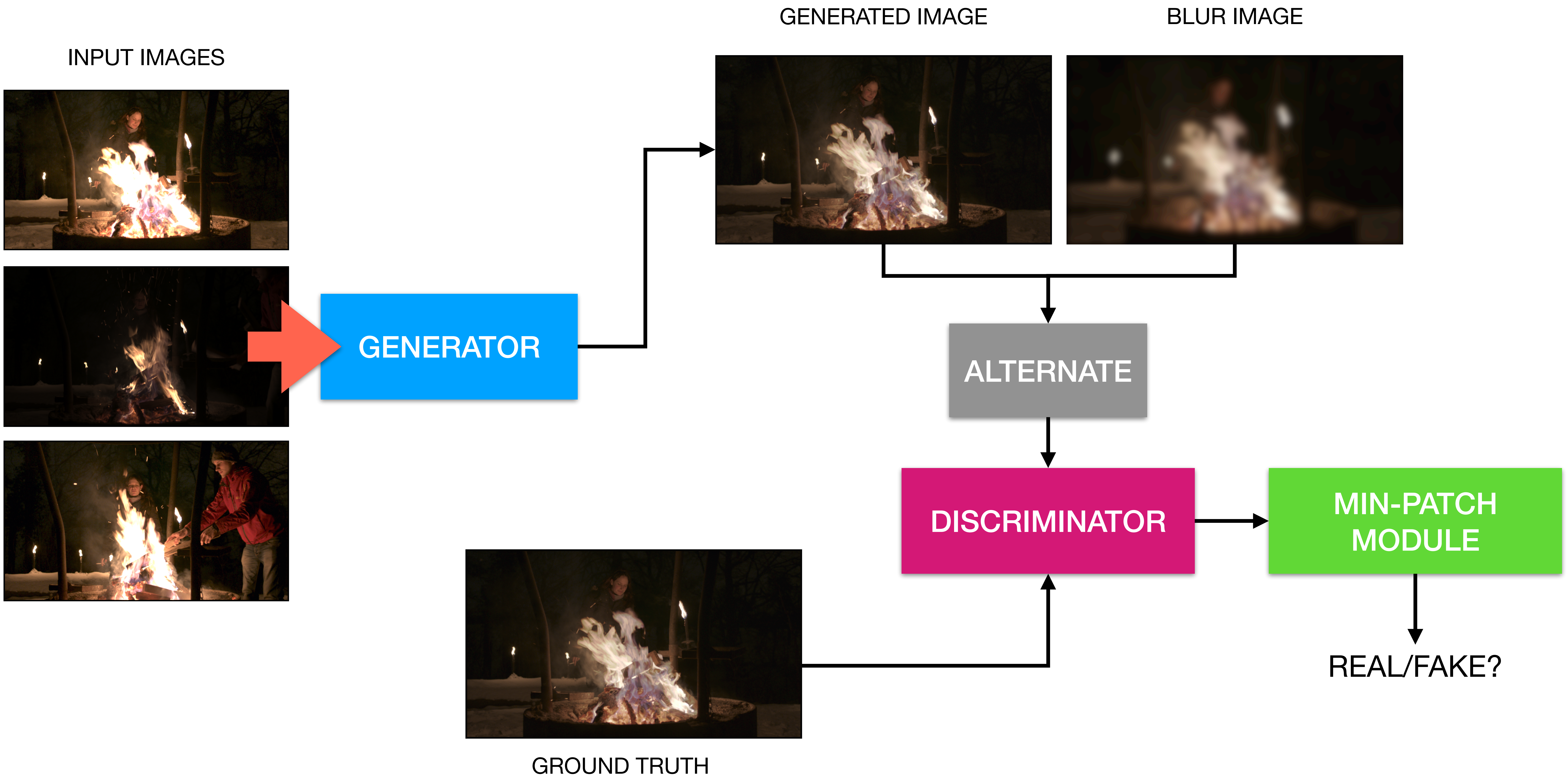
HDRGAN - Niu et al. 2021: Training



UPHDR-GAN - Li et al. 2022: Generator



UPHDR-GAN - Li et al. 2022: Training



Loss Functions

Loss Function in the μ-Law Domain

- Kalantari et al. 2017 introduced a L2 loss function in a tone-mapped domain:

$$\mathscr{L}_{\text{rec}}(\hat{I}, I) = \|\tau(I) - \tau(\hat{I})\|_2$$

where $\boldsymbol{\tau}(\cdot)$ is a differentiable tone mapping function based on the $\boldsymbol{\mu}$-law:

$$\tau(I) = \frac{\log(1 + \mu I)}{\log(1 + \mu)} \qquad \mu = 5000$$

- Note that there are variants of $\mathscr{L}_{\text{rec}}$ where we have L1 instead of L2.
- This loss function is **ubiquitous** in most HDR works for reconstruction and inverse tone mapping.

GAN Loss

- Our goal is:

$$\arg\min_{G}\max_{D}\mathscr{L}(G, D)$$

- Typically a GAN loss is defined as:

$$\mathscr{L}(G, D) = \alpha_1 \mathscr{L}_{\text{GAN}}(G, D) + \alpha_2 \mathscr{L}_{\text{rec}}(G)$$

where:

- $\mathscr{L}_{\text{GAN}}(G, D)$ is the adversial loss.
- $\mathscr{L}_{\text{rec}}(G)$ is the content/reconstruction loss.
- α_1 and α_2 are weights for balancing the two losses.

GAN Loss: HDRGAN

- Niu et al. 2021 has a GAN scheme with a content/reconstruction loss:

$$\mathscr{L}_{\text{rec}} = \min_{G} \left(\|\tau(\hat{H}_1) - \hat{H}\|_1 + \|\tau(\hat{H}_2) - \hat{H}\|_1 \right)$$

- And a GAN loss based on the sphere generative adverbial loss [Park and Kwon 2019], where the Discriminator output an $\boldsymbol{n}$-dimensional vector $\mathbf{q}$ which is projected on $\mathbf{p} \in \mathbb{S}^n$:

$$\mathscr{L}_{\text{GAN}} = \min_{G} \max_{D} \sum_{r} \mathbb{E}_{\mathbf{z}}[d_s^r(\mathbf{N}, D(\mathbf{z}))] - \sum_{r} \mathbb{E}_{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3} d_s^r(\mathbf{N}, D(G(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3))]$$

where $d_s(\mathbf{p}, \mathbf{p}')$ is the distance on the hypersphere, and $\mathbf{N} = [0,\ldots,0,1] \in \mathbb{R}^n$.

GAN Loss: UPHDR-GAN

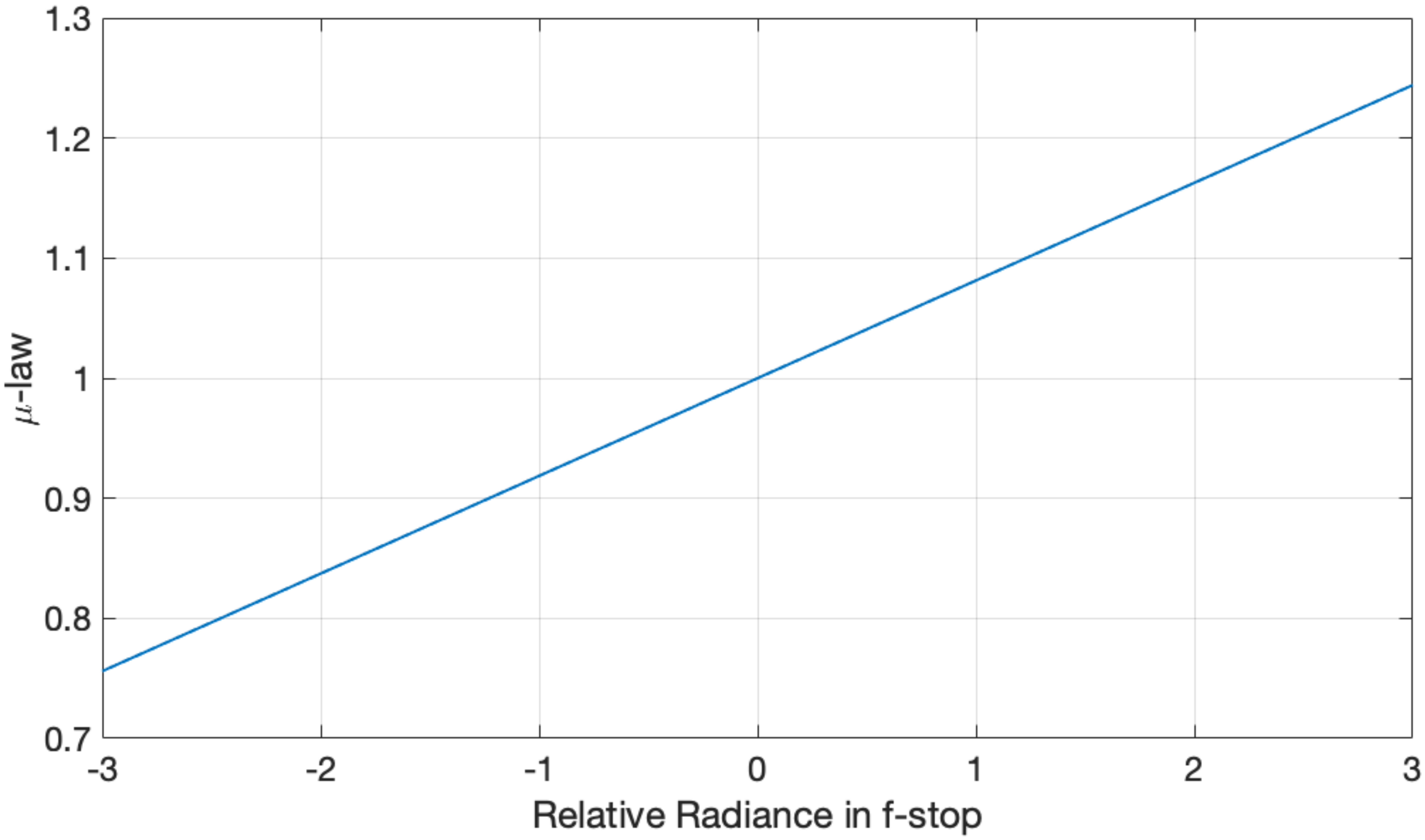
- Li et al. 2022 has a GAN scheme with a content/reconstruction loss:

$$\mathscr{L}_{\mathrm{rec}} = \mathbb{E}_{x \sim p_{\mathrm{data}}(x)}\left[\left\| VGG(G(x)) - VGG(x_2) \right\|_1\right]$$

- The GAN loss is defined as:

$$\mathscr{L}_{\mathrm{GAN}} = \mathbb{E}_{y \sim p_{\mathrm{data}}(y)}[\log D(y)] + \mathbb{E}_{x \sim p_{\mathrm{data}}(x)}[\log 1 - D(G(y))] + \mathbb{E}_{b \sim p_{\mathrm{data}}(b)}[\log(1 - D(b))]$$

Loss Function in the μ-Law Domain



Videos

HDR Videos: Temporally Varying Exposure Time

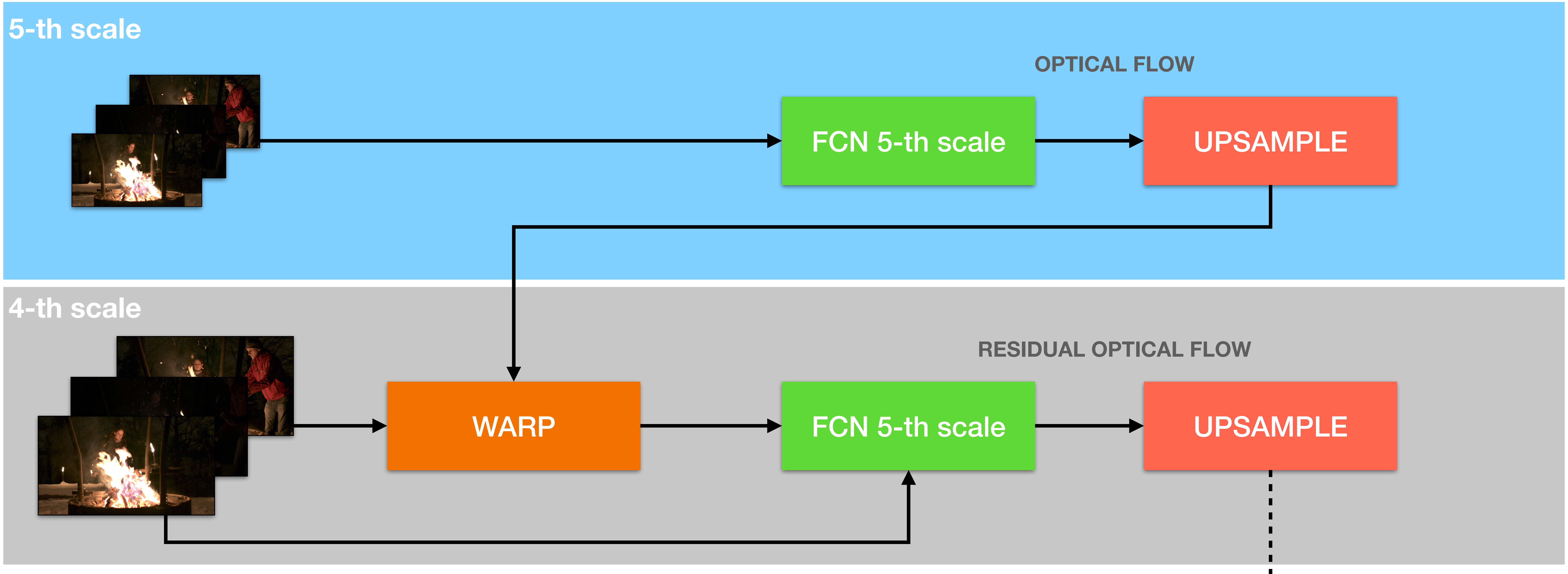
Stream

t_0 t_1 t_2

Video Courtesy of Jan Fröhlich - Stuttgart HDR Video Dataset

Video Strategies: Kalantari and Ramamoorthi 2019

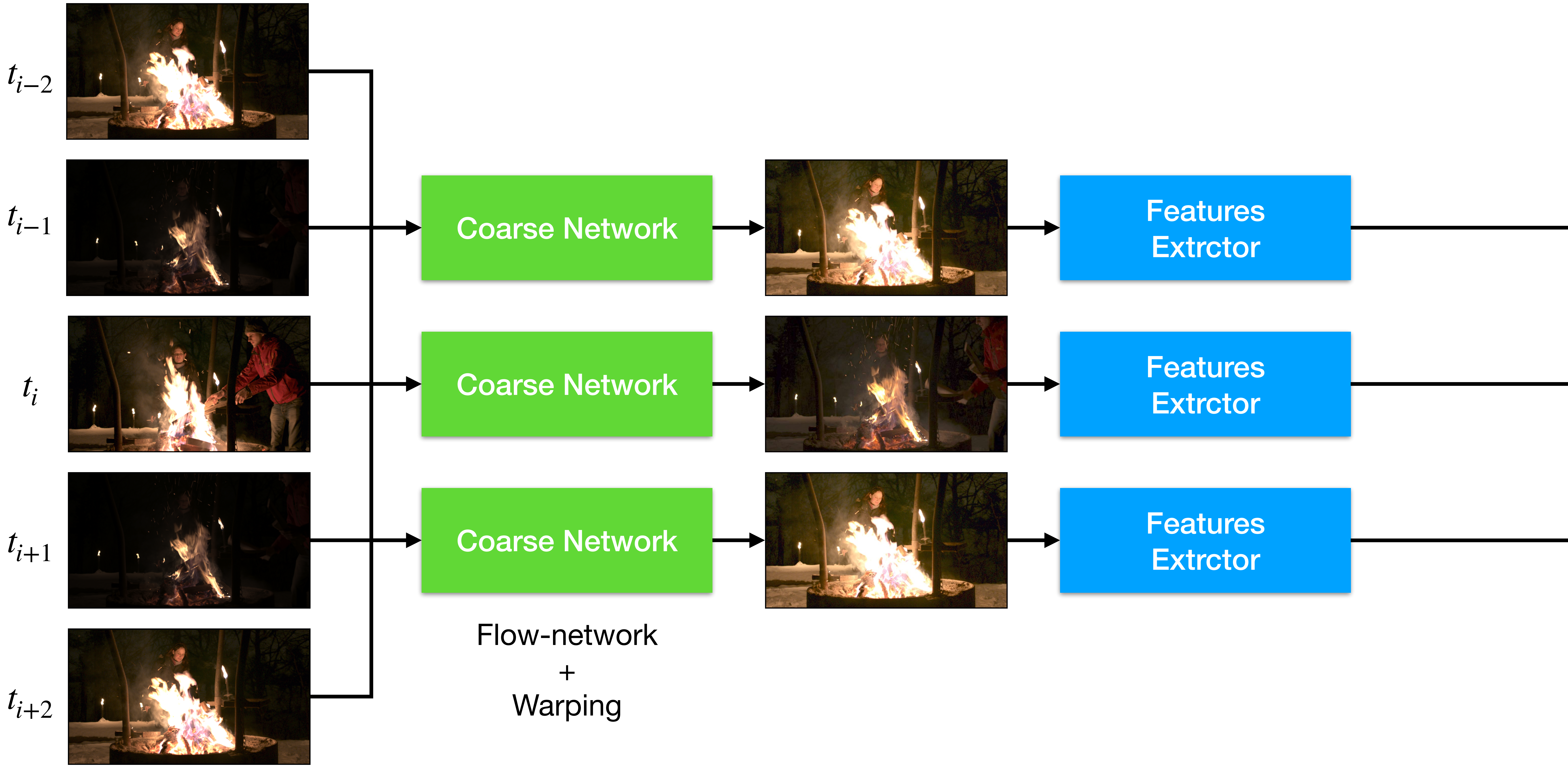
- A 5-scale pyramid for computing a multi-scale optical flow using a CNN for each scale a simple FCN:



Video Strategies: Kalantari and Ramamoorthi 2019

- Similar to the previous work by Kalantari et al. 2017, there is a merger (encoder-decoder).
- To enforce temporal coherency and reduce artifacts the merger uses neighbors frames at previous and next time.

Video Strategies: Chen et al. 2021



Video Strategies: Chen et al. 2021

Evaluation

Metrics

- Many works uses:
  - Linear domain PSNR and SSIM.
  - μ-law or Reinhard et al. 2002's TMO PSNR or SSIM
- These approaches have many issues:
  - Linear domain PSNR and SSIM are prone to outliers.
  - μ-law and Reinhard et al. 2002's TMO are empirical approaches that do not model the Human Visual System.
    - They may introduce distortions.

Metrics

- PSNR and SSIM should be computed using the PU21:
  - PU21 encodes absolute HDR linear value into approximately perceptually uniform (PU) values.
- HDR-VDP 2.2, HDR-VDP 3.0.6, and FovVideoVDP.
- Deghosting artifacts: Tursun et al. 2016.
- Note that may HDR reference images and output images are **uncalibrated**:
  - If we do not have calibration data:
    - Display-referred values.

Limitations

Limitations

- The CRF needs to be known (a partial limitation);
- Most methods are limited to merge ONLY three images:
  - There is not method addressing an arbitrary number of images or more than threes.
- The difference in f-stop has to be fixed:
  - There is no method that can merge an image at -5-stop, 0-stop, and +1-stop.

Other Problems in Reconstruction

Other Reconstruction Problems

- We have other problems for HDR reconstruction with partial real information that can be solved using deep learning:
  - Assorted pixels/rows [Choi et al. 2017, Çogolan et al. 2020, Suda et al. 2020, Xu et al. 2021, Vien et al. 2022].
  - HDR from deep optics/masks [Alghamdi et al. 2019, Metzler et al. 2020]
  - HDR reconstruction using an event camera [Wang et al. 2019, Shaw et al. 2022, Messikommer et al. 2022].
  - HDR reconstruction for quanta sensors [Gnanasambandam et al. 2020, Gao et al. 2022].

Questions?

Questions?

Modern High Dynamic Range Imaging at the Time of Deep Learning

Inverse Tone Mapping

Francesco Banterle and Alessandro Artusi

Introduction

- Acquisition is tedious:
  - Images alignment.
  - Ghosts removal.

- What can we do without bracketing or modified/expensive hardware?

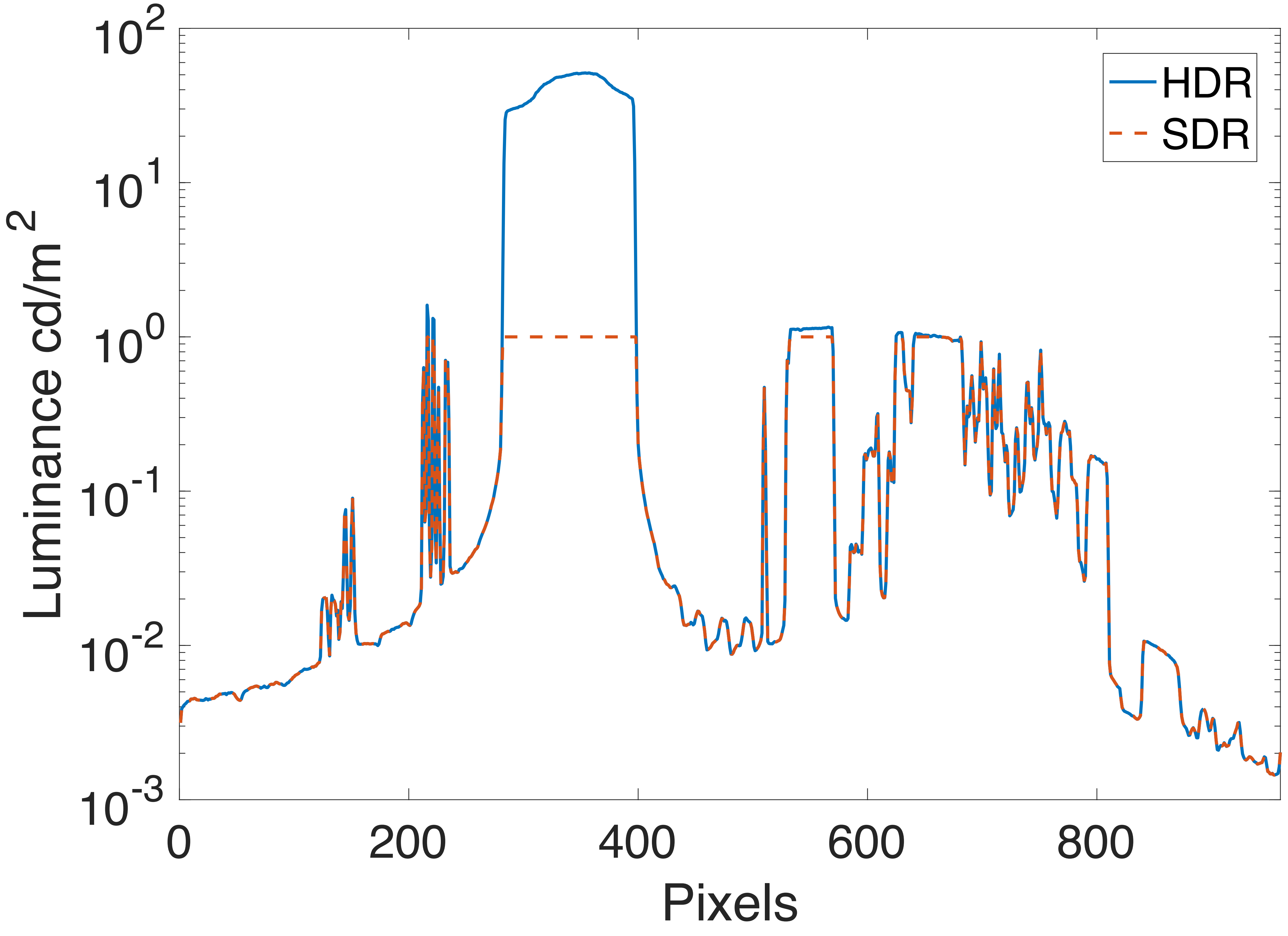
Introduction

- Acquisition is tedious:
    - Images alignment.
    - Ghosts removal.

- What can we do without bracketing or modified/expensive hardware?

The Problem

Image

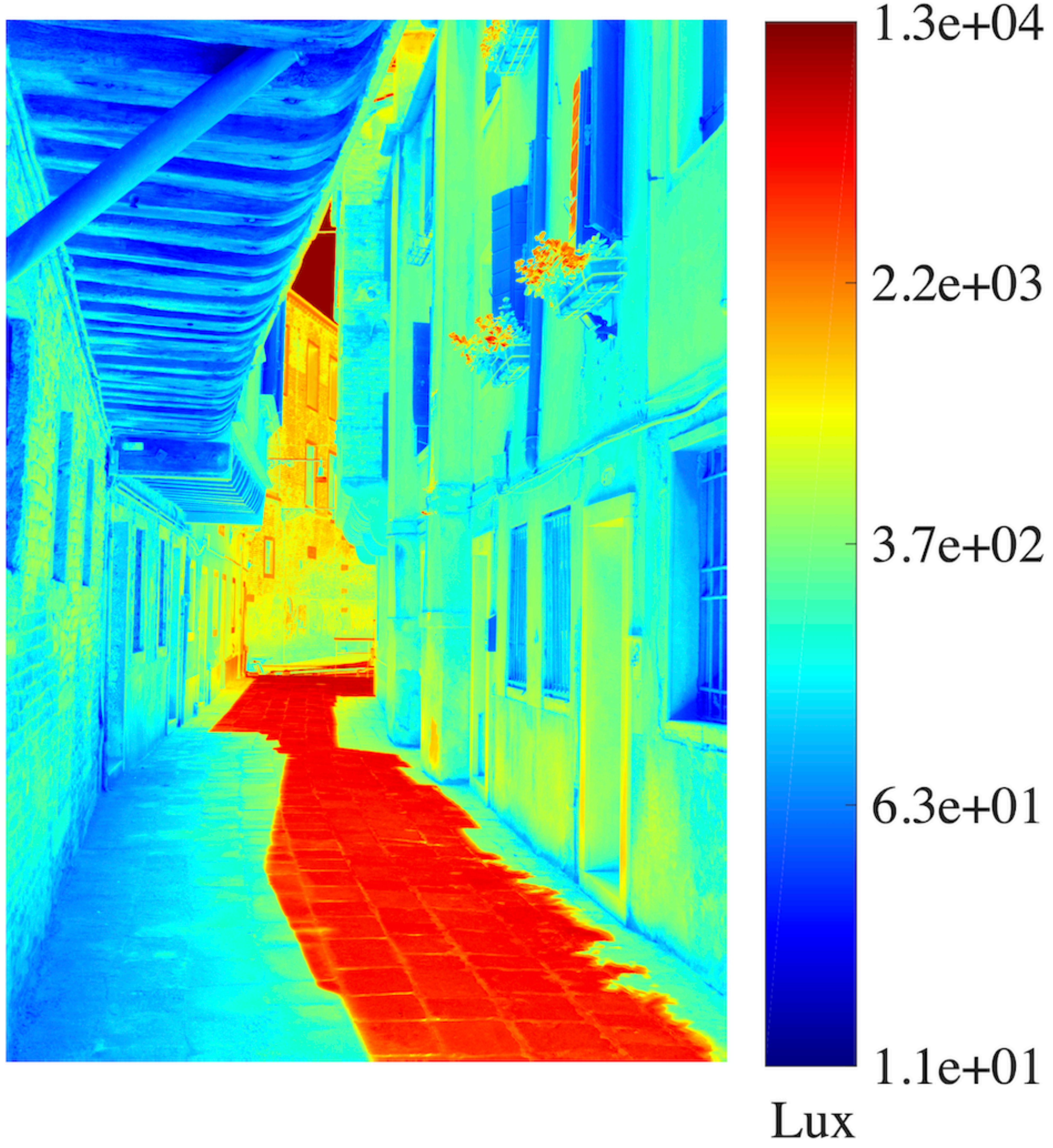


Histogram of the red dotted line

The Problem

SDR Image

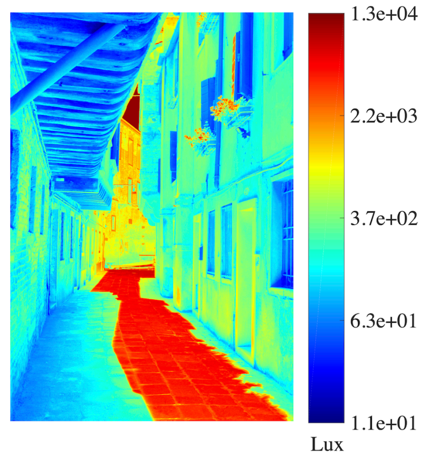
ITMO



HDR Image

The Full Pipeline

Dequantization → Linearization → Hallucination →

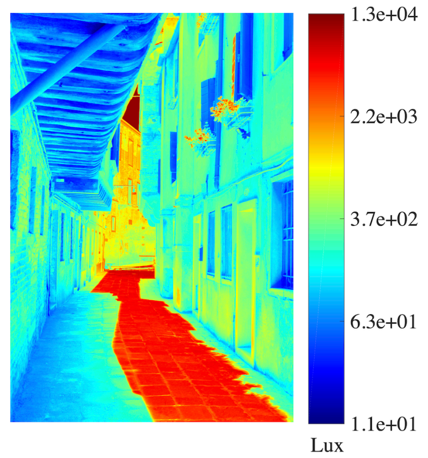


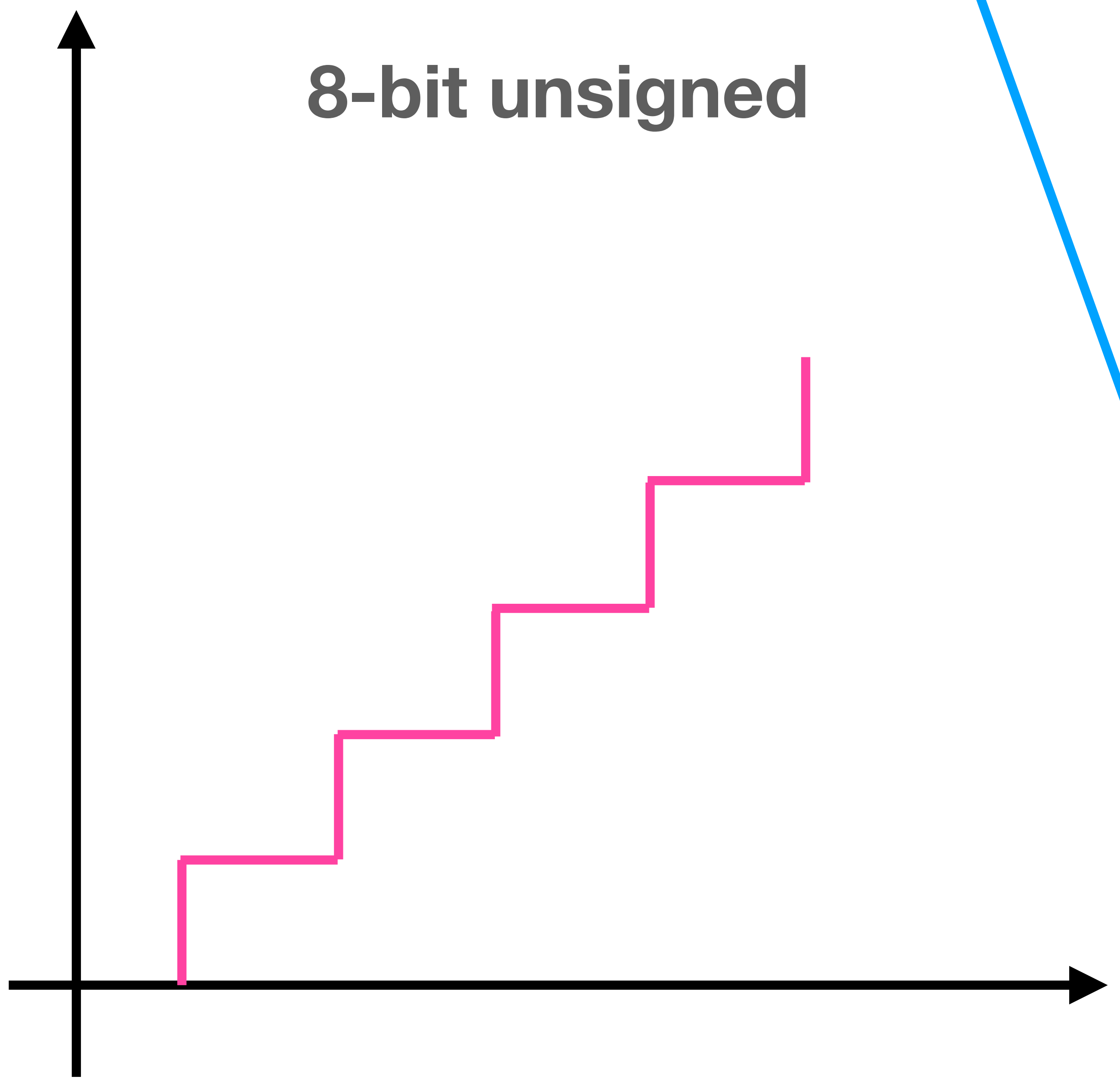
The Full Pipeline

Dequantization

Linearization

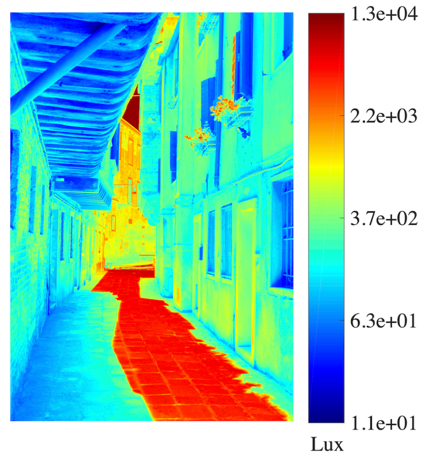
Hallucination

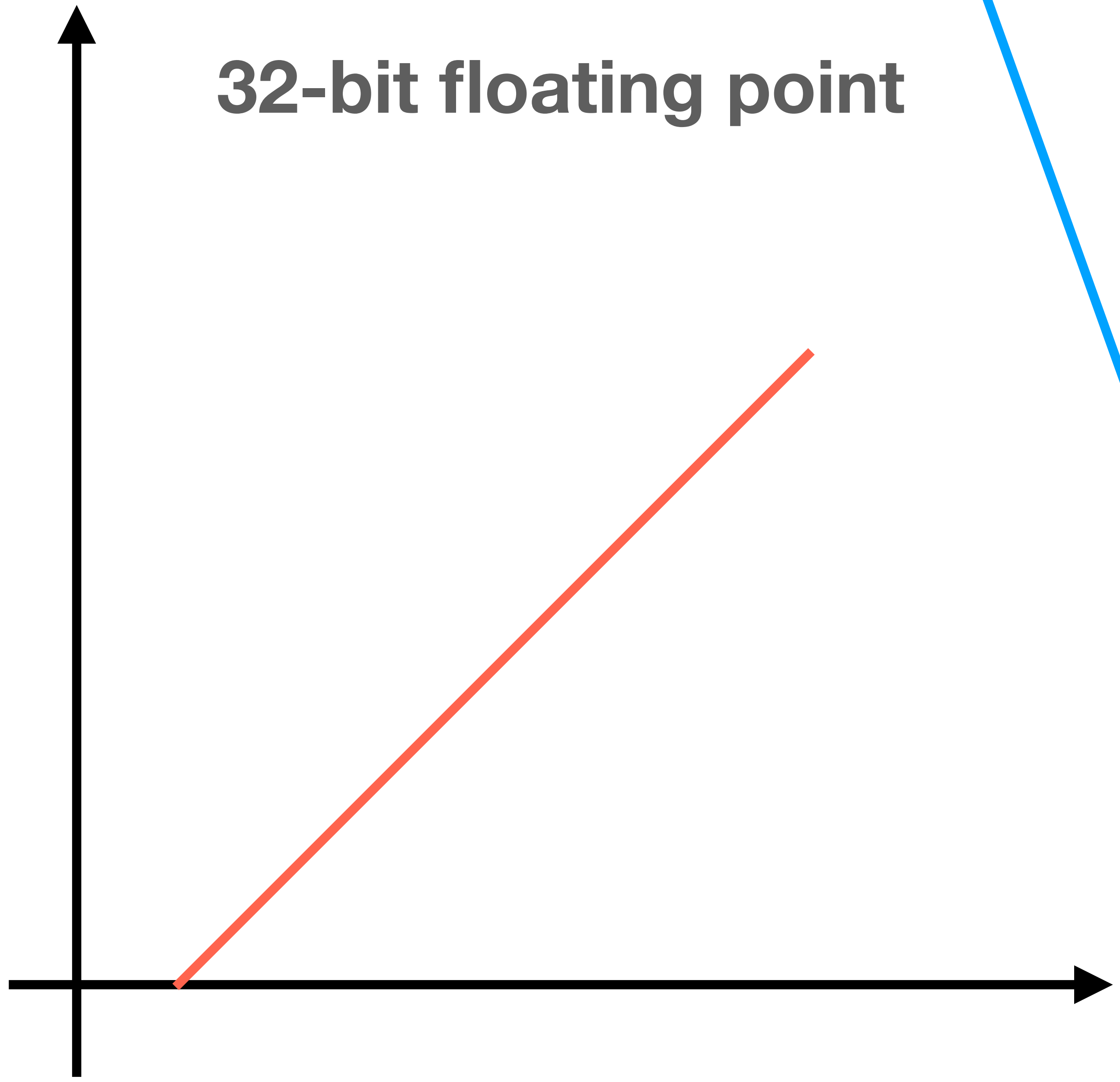




The Full Pipeline

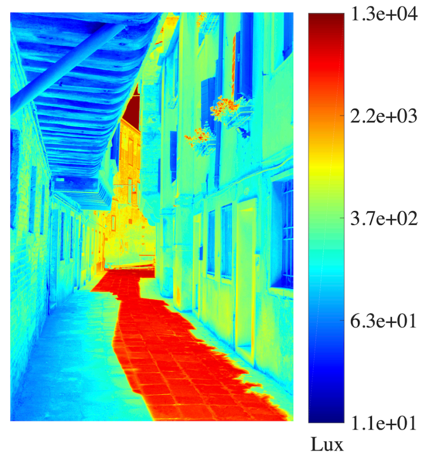
Dequantization → Linearization → Hallucination →

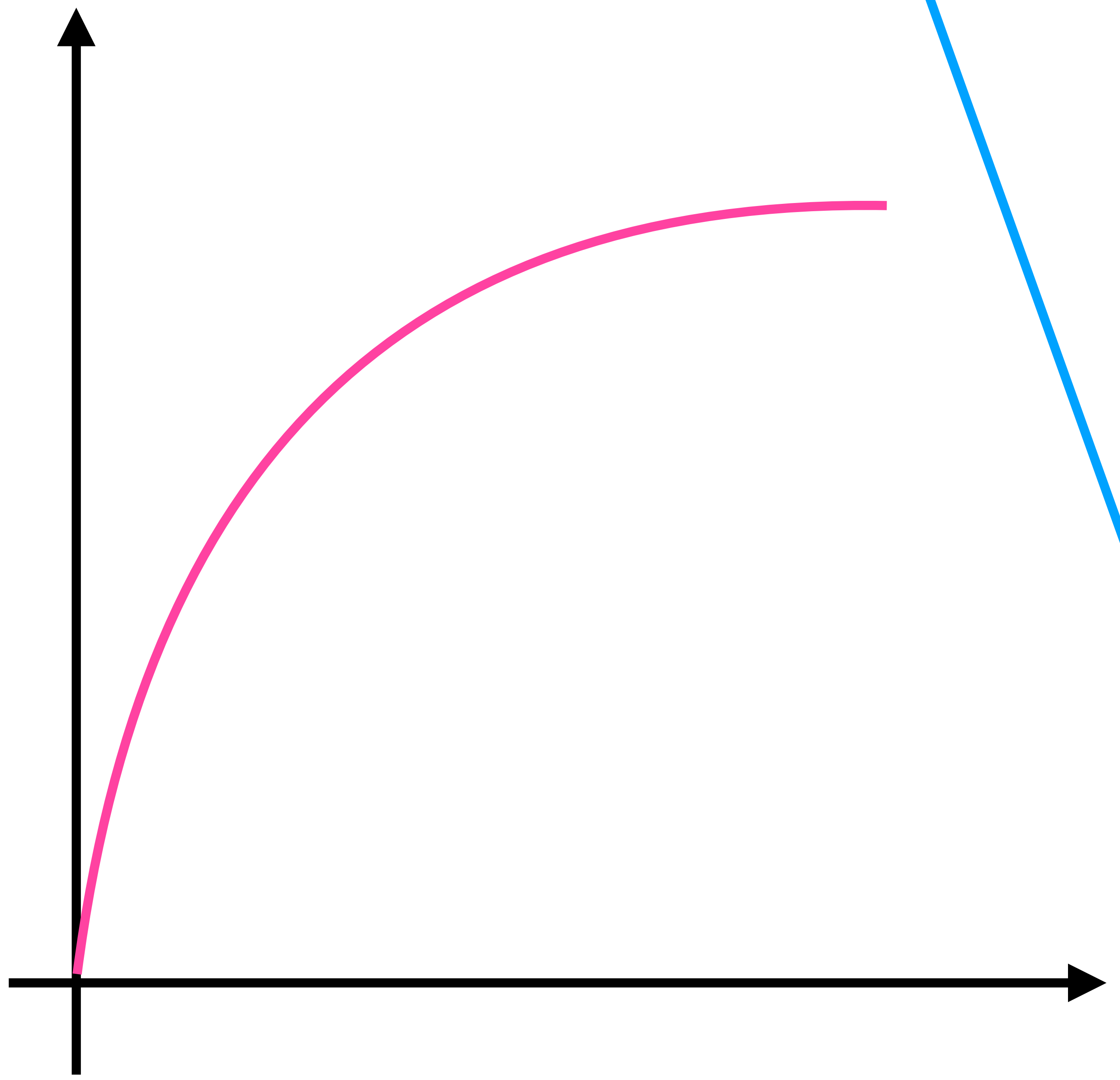




The Full Pipeline

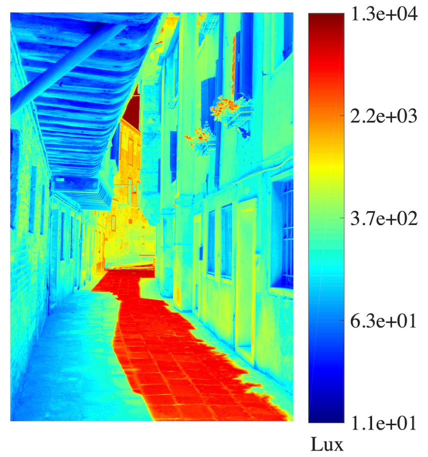
Dequantization → Linearization → Hallucination →

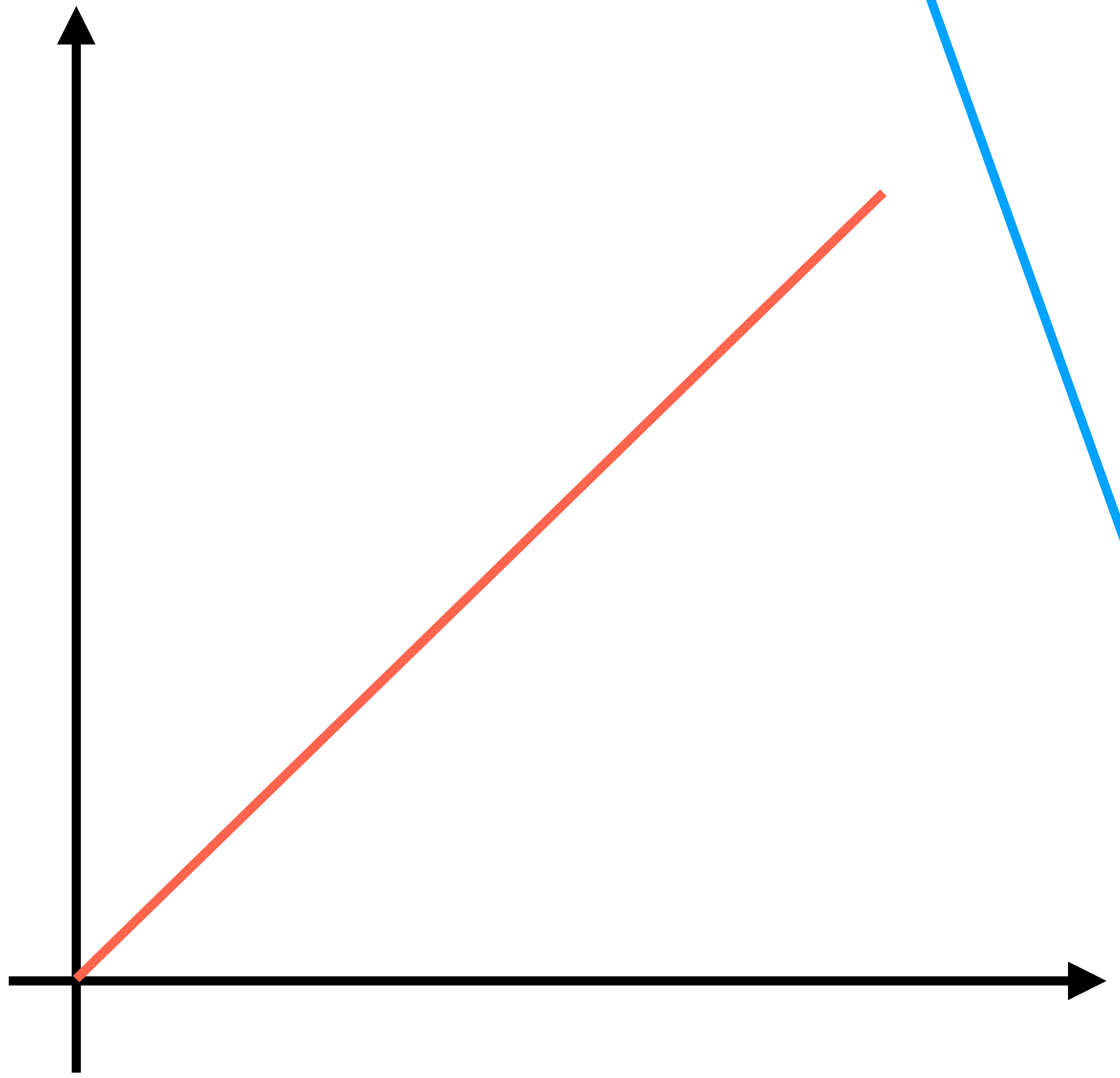




The Full Pipeline

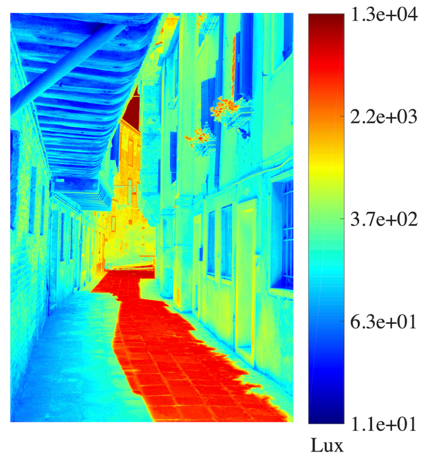
Dequantization → Linearization → Hallucination →





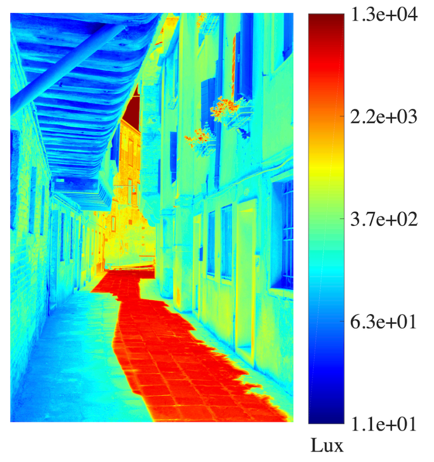
The Full Pipeline

Dequantization → Linearization → Hallucination →



The Full Pipeline

Dequantization → Linearization → Hallucination →



The Linearization Dilemma

The Linarization Dilemma

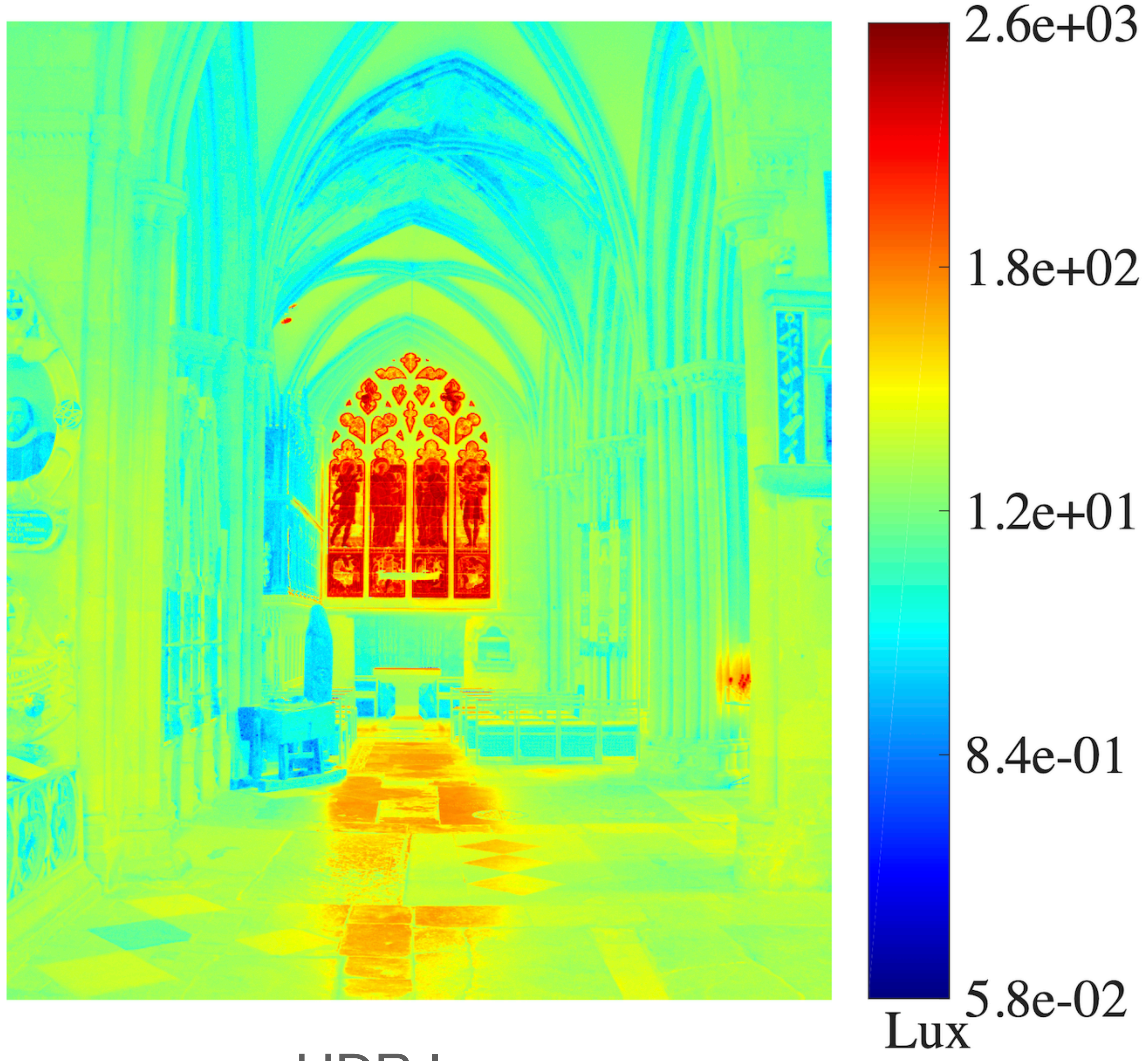
- One of the first step to decide is how we linearize the input SDR.
- Many methods uses a standard $\gamma = 2$ or $\gamma = 2.2$:
  - Eilertsen et al. 2017, Marnerides et al. 2018, etc.
- Note that many modern cameras encodes images using common CRF such as sRGB, PQ, and HLG.

Architectures

Architectures

- Here, we have **two possibilities** to solve the problem:
  - Approach 1: Given an input image, we generate directly a HDR image

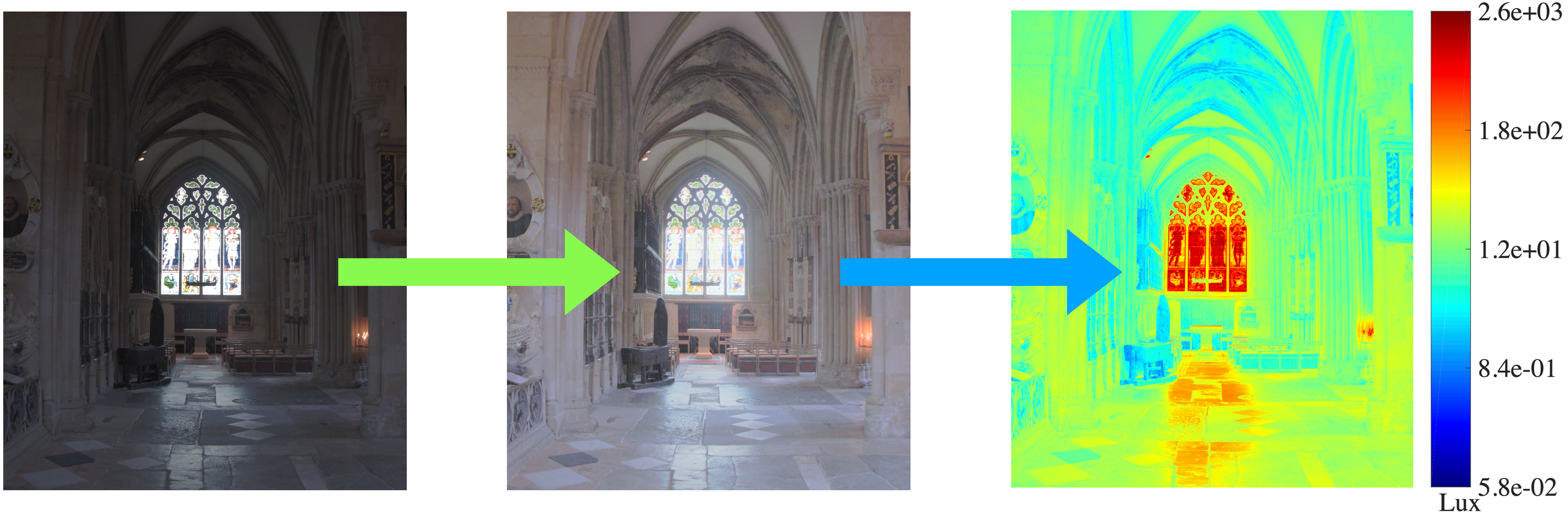
SDR Image



HDR Image

Architectures

- This approach may also compute a tone mapped version of the radiance map to recover. If the tone mapper is invertible, we can obtain a radiance map.



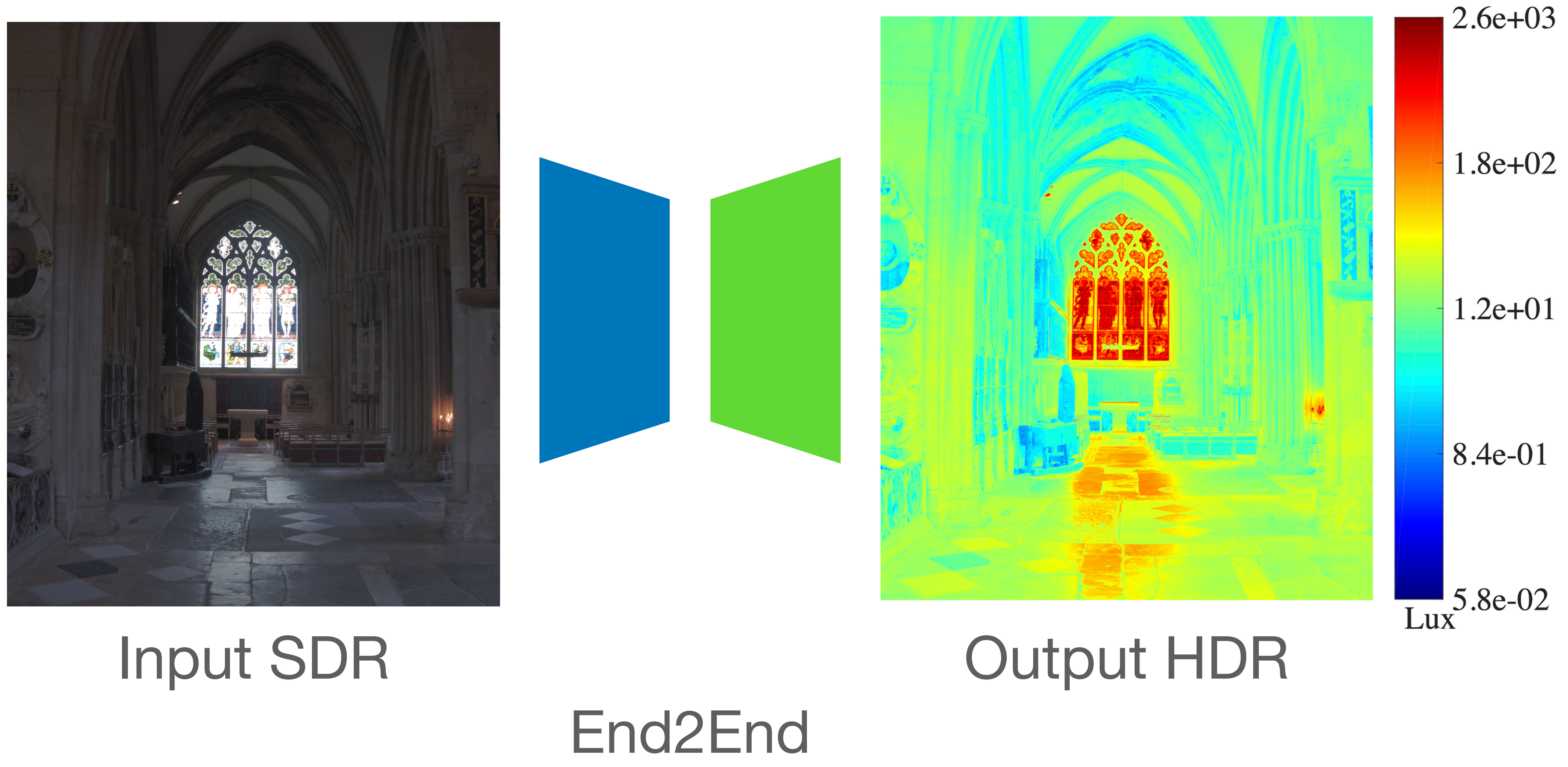
SDR Image | Tone Mapped HDR Image | HDR Image

Architectures

- Another possibility is:
  - Approach 2: Given an input SDR image, we generate a stack of n SDR images at different exposure times.

Which Architecture?

- The bread and butter of most iTMO are
  - FCN.
  - U-Net [Eilertsen et al 2017].
  - Residual Blocks [Kim et al. 2019].
- They are simple models that generally works.

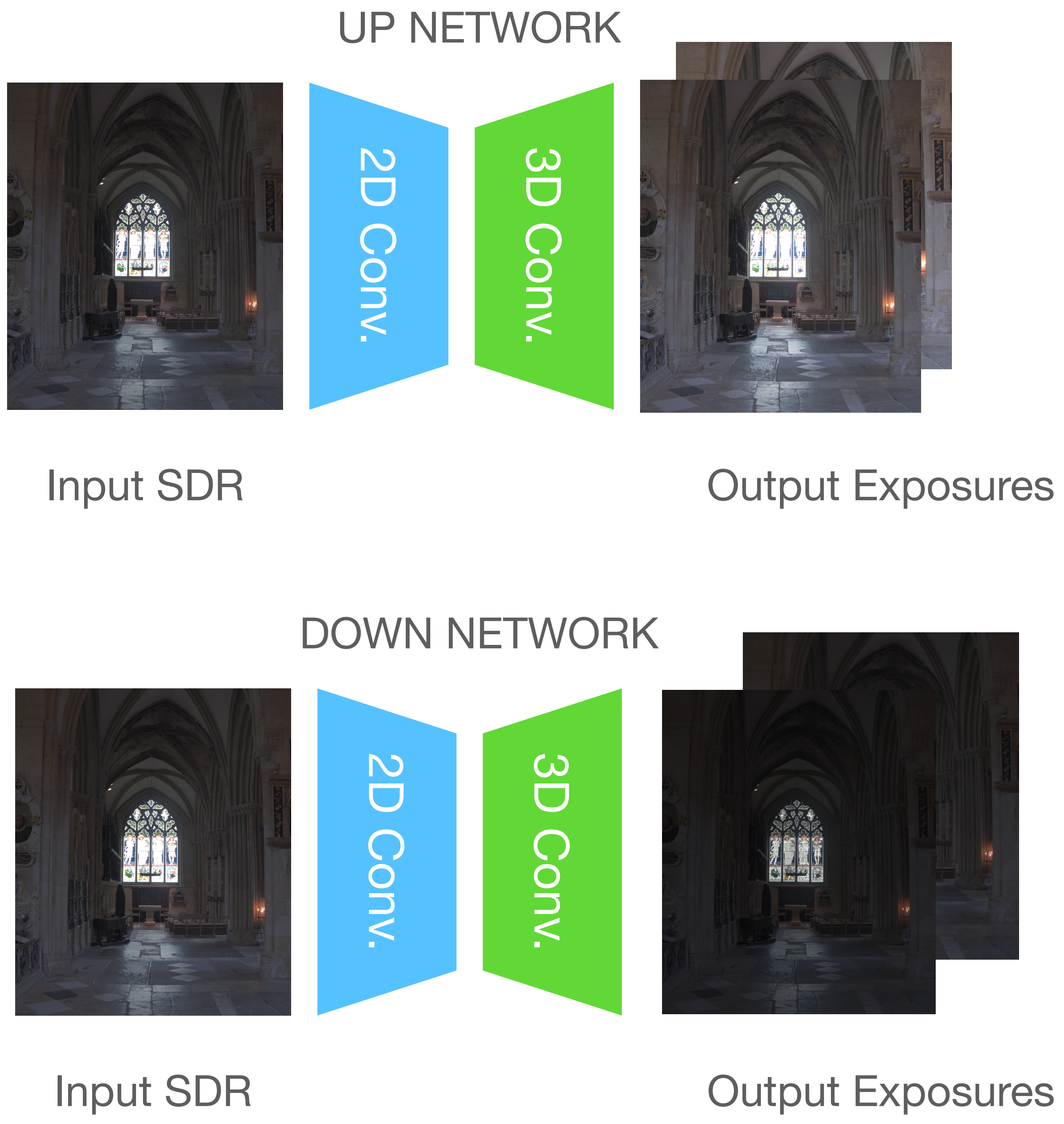


Which Architecture?

- Activation function:
  - LeakyReLU/GeLU in the encoder part.
  - ReLU in the decoder part.
  - The last layer:
    - Sigmoid: tone mapped results or single exposures.

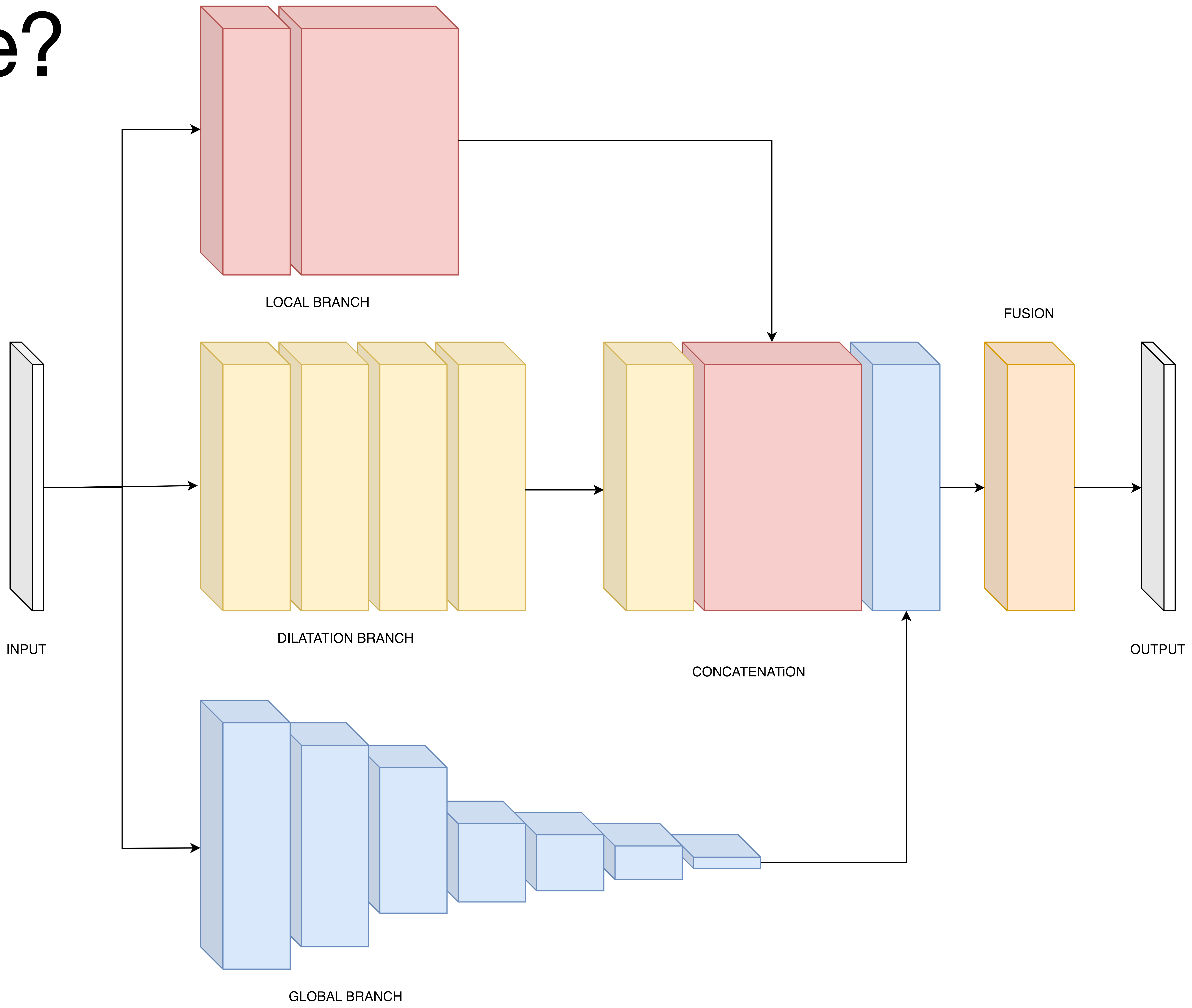
Which Architecture?

- Endo et al. 2017 employs a classic U-Net with a twist:
  - Encoder has 2D convolutions.
  - Decoders has 3D convolutions:
    - Generate in a single network all exposures.
    - Limitations: the number of exposures are limited.



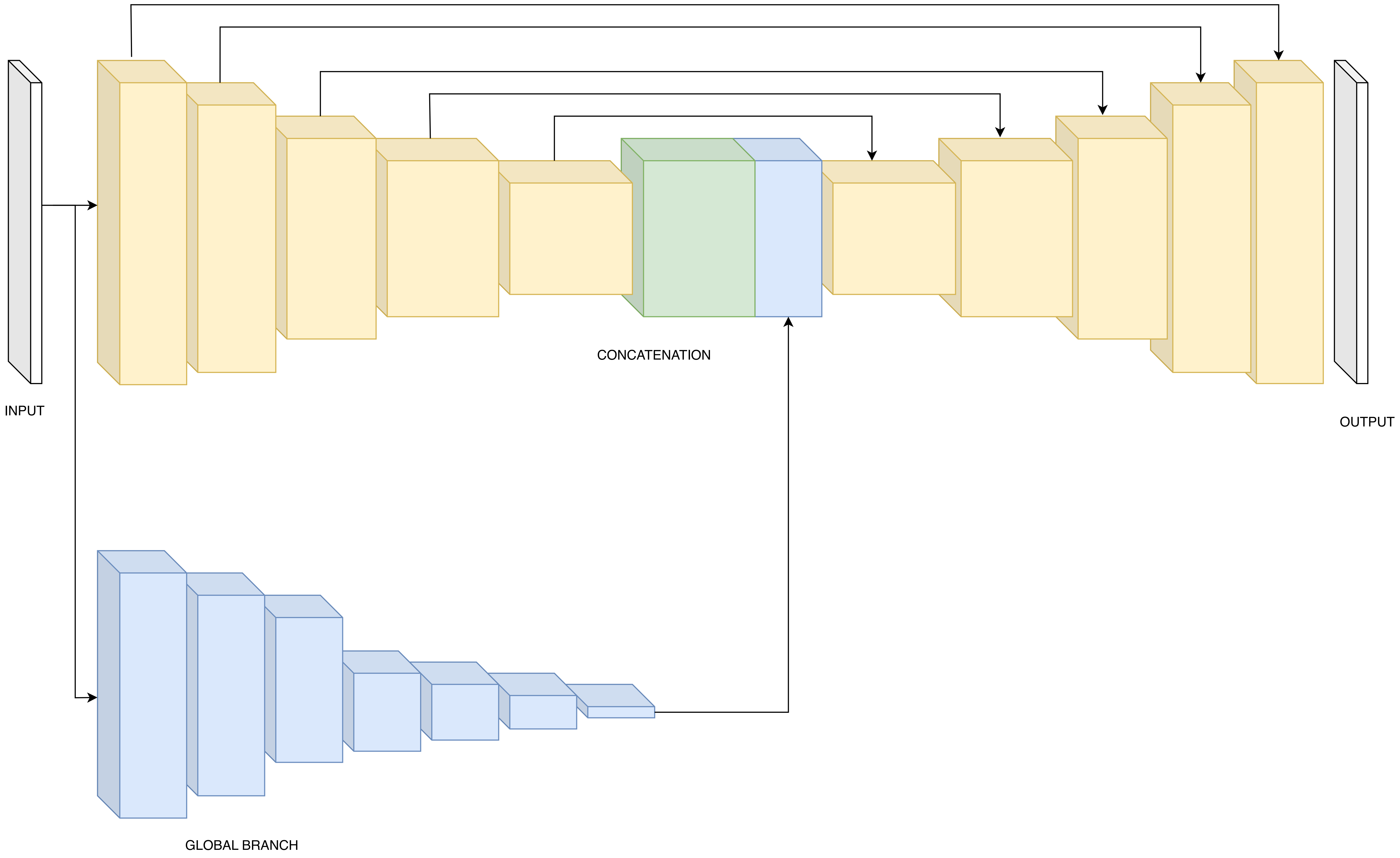
Which Architecture?

- Marnerides et al. 2018 proposed a multi-branch architecture to overcome U-Net limits:
    - Local features;
    - Medium features;
    - Global features.



Which Architecture?

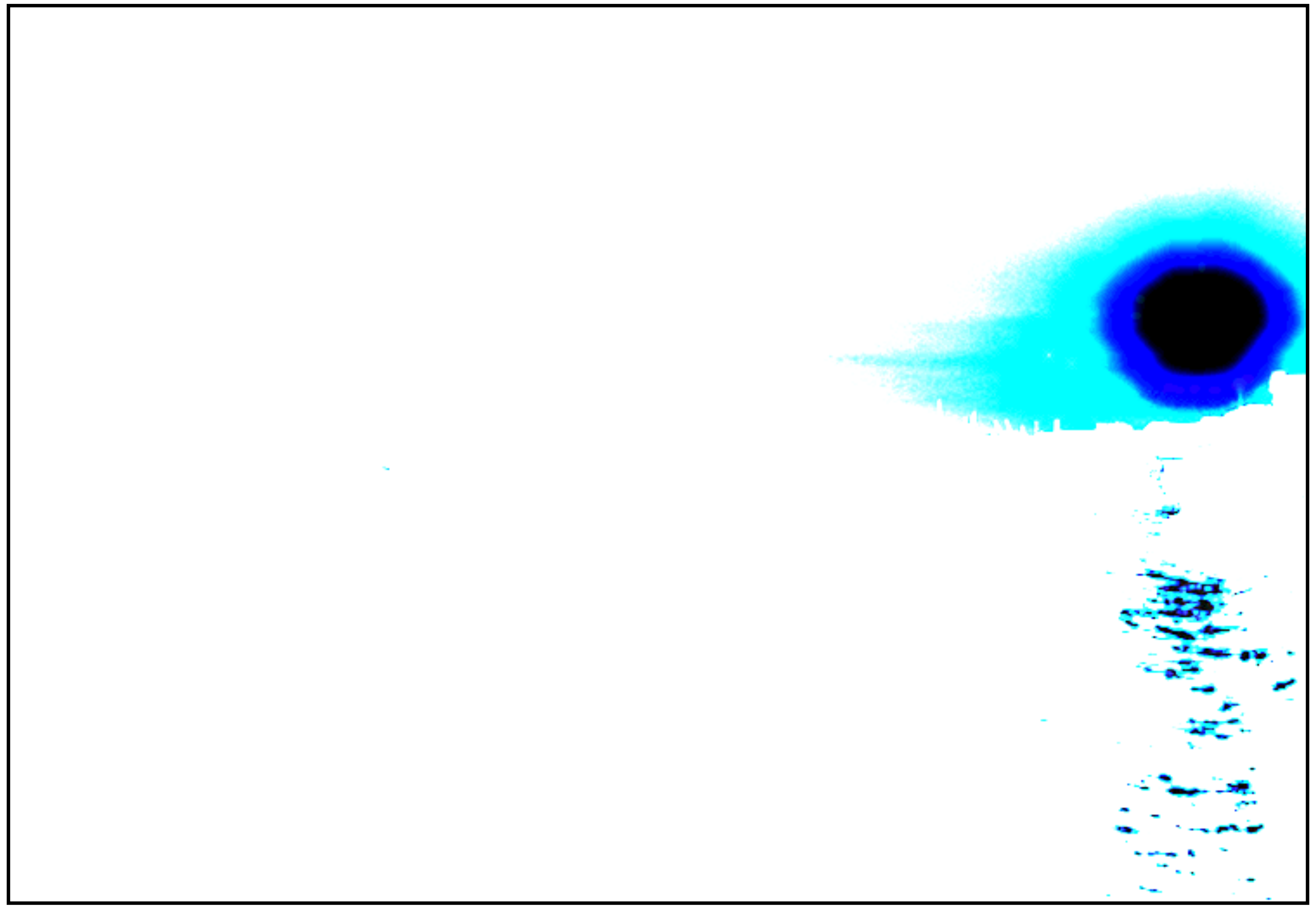
- Kinoshita and Kiya 2019 paired the global branch with U-Net to overcome some limitations of U-Net.



Which Architecture? Feature Masking

- Santos et al. 2020 introduces masking:
  - We can see inverse tone mapping as an inpainting problem, where our mask is defined using over-exposed pixels.

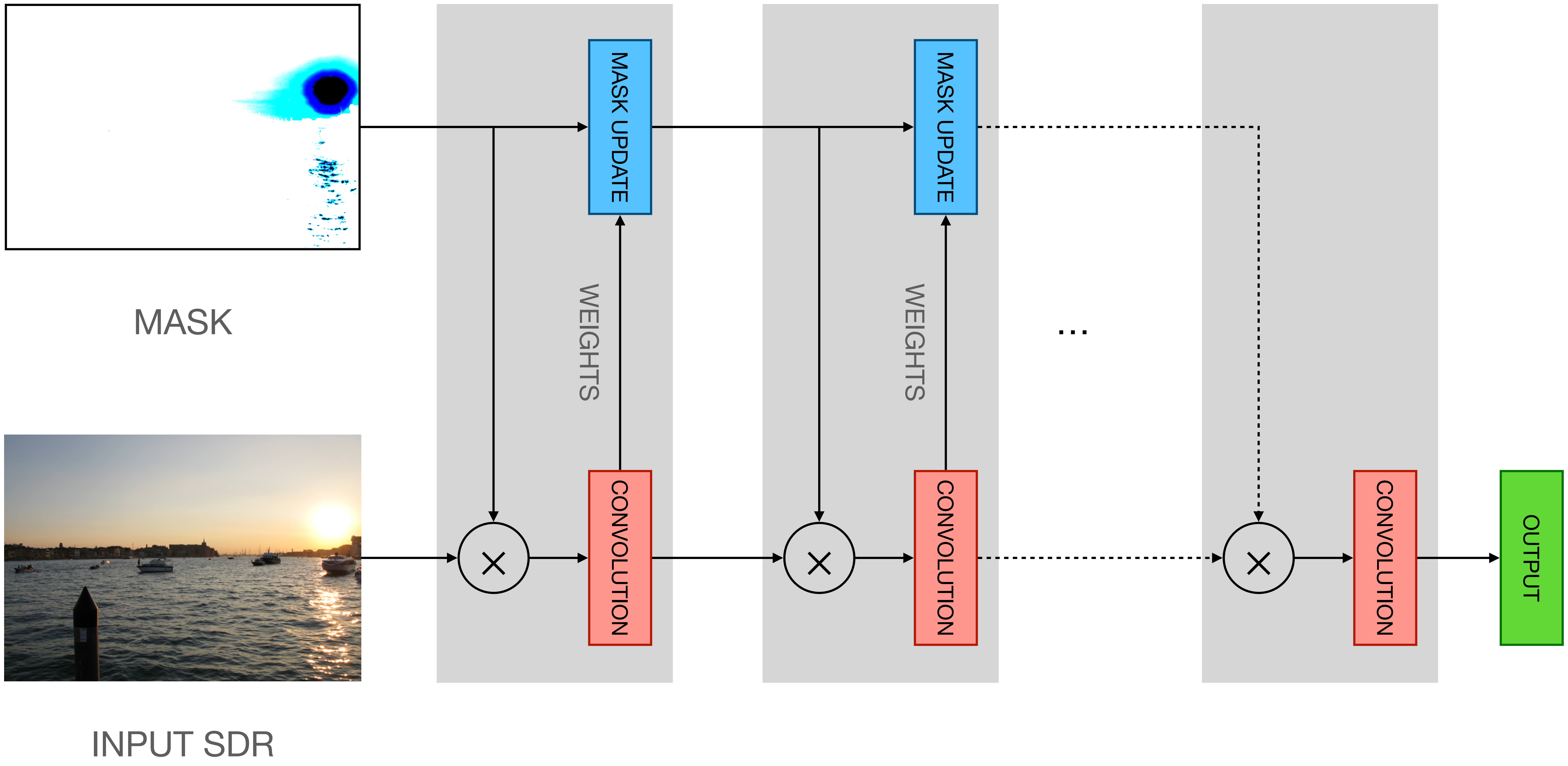
INPUT SDR



MASK

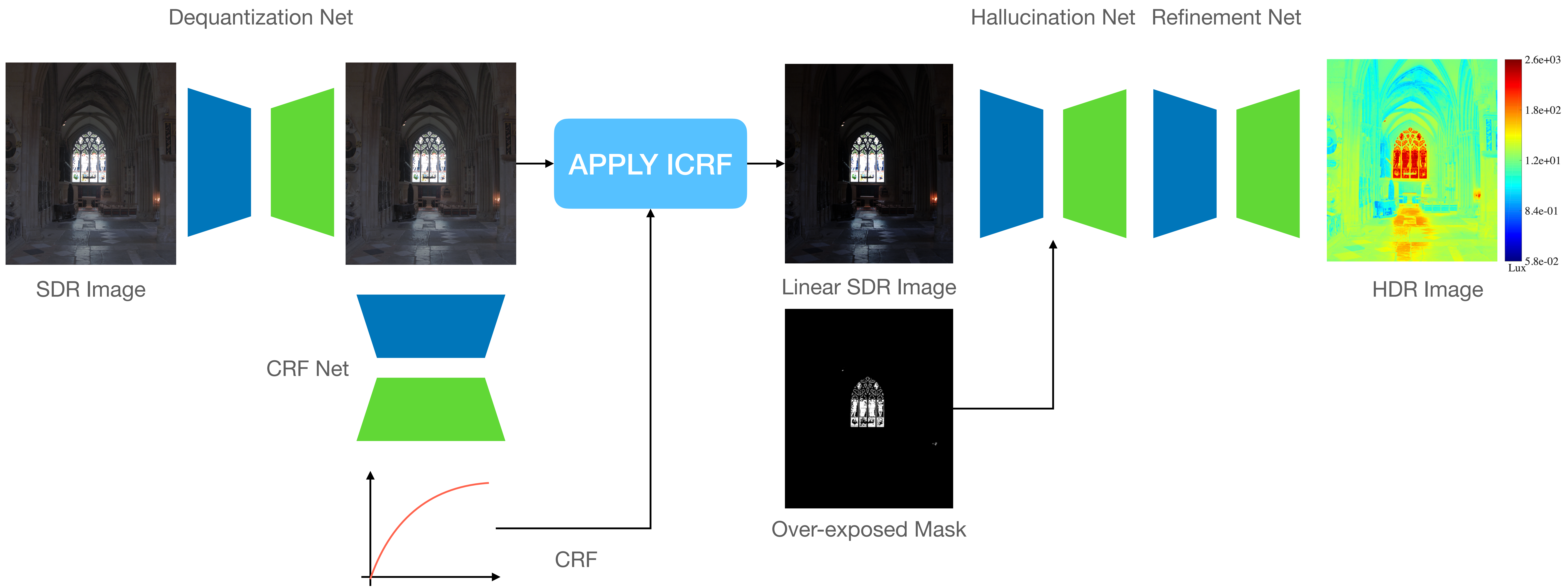
Which Architecture? Feature Masking

- Santos et al. 2020 apply the mask at each convolution step:



Which Architecture? Feature Masking

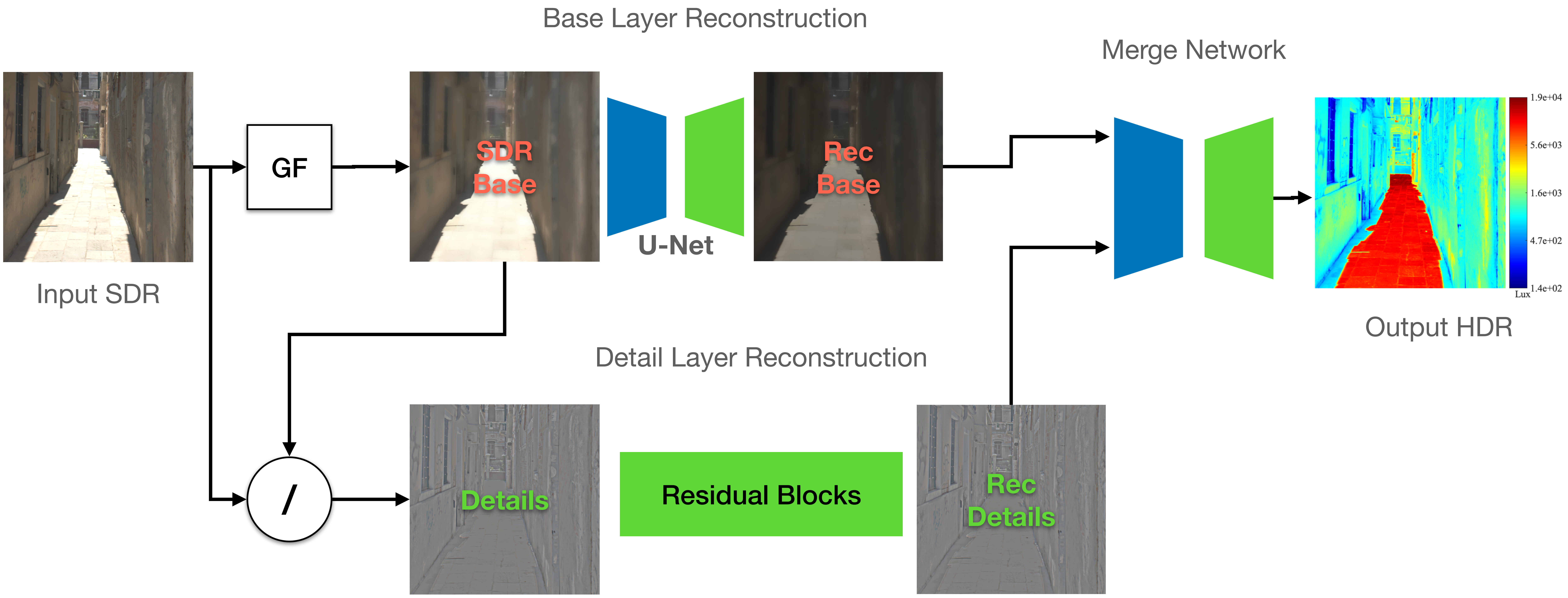
- Liu et al. 2020 has a network that recovers the inverse camera pipeline:



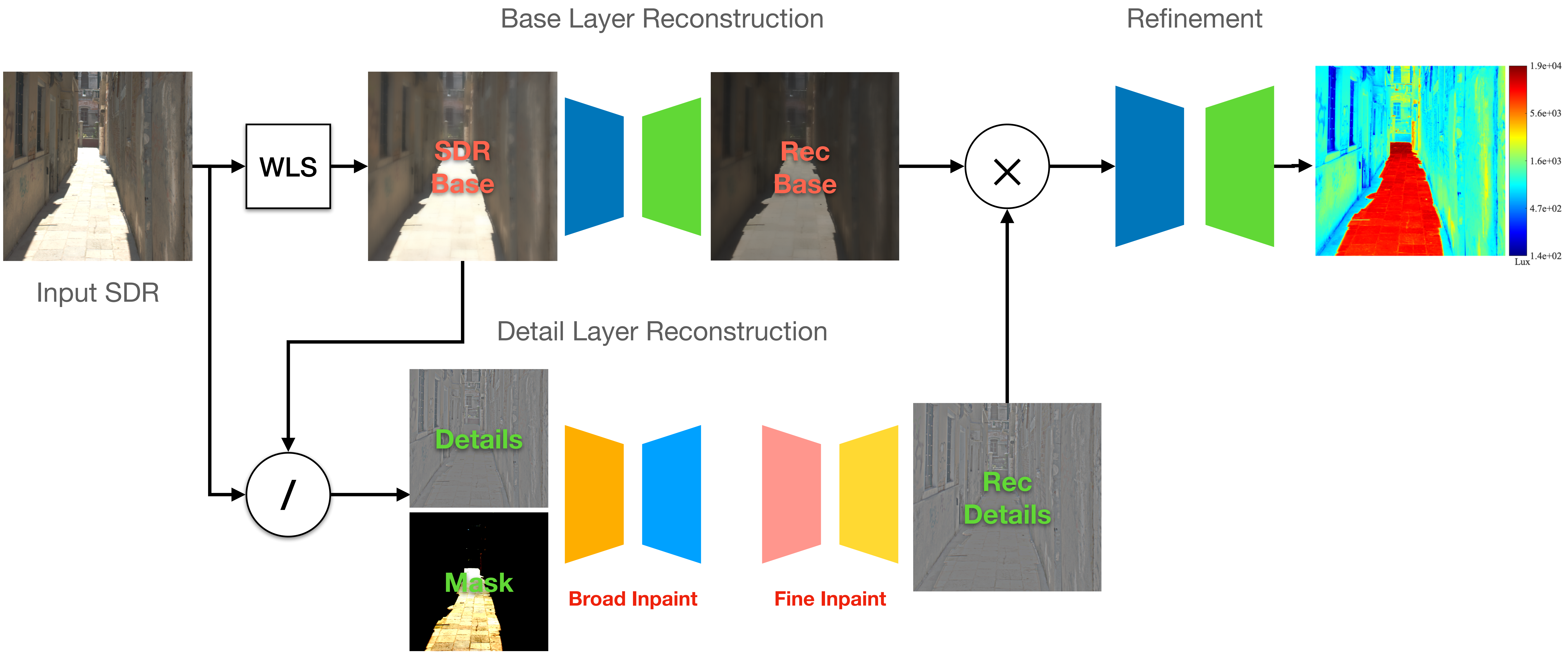
Which Architecture? Frequencies Separation

- adopted a classic end2end encoding paired with a GAN, so nothing special right now… The novelty:
  - A network for each frequency:
    - Base image or I_b: is the output of filtering the input image, I, filtered using an edge-aware filter:
      - Bilateral Filter, Guided Filter, WLS, etc.
    - Detail image or I_d: is an image encoding the high-frequency details, and it is computed as: $I_d = I/I_b$.
    - Capece et al. 2019 used a similar strategy for relighting faces.
  - A similar work with more refinement networks was proposed by Zhang and Aydın 2021 using WLS instead of the bilateral filter.

Which Architecture? Frequencies Separation - Wang et al. 2019



Which Architecture? Frequencies Separation - Zhang and Aydın 2021



Datasets

HDR Image Datasets

- Proper HDR images/videos (≥ 18-stop) are scarce on the Internet.
- There are few datasets of real HDR images.
- These datasets are typically uncalibrated:
    - This means that luminance values are relative; i.e., they do not have absolute values in cd/m^2.
    - Colors may not match the real colors.
- They are stored in different formats without the use of a standard. Typically, using the Radiance (.hdr) or OpenEXR (.exr) format files.

HDR Image Datasets

Dataset Name	#Images	#Resolution	Calibrated	Website
HDR Survey	108	5MPix	Scene-referred	http://markfairchild.org/HDR.html
HDR Eye	47	2MPix (full-HD)	Display-referred	
Stanford HDR Dataset	88	0.32Mpix	Scene-referred	https://qualinet.github.io/databases/image/high_dynamic_range_imaging_dataset_of_natural_scenes/
Laval HDR Indoor	2100	2MPix (2:1 ratio)	Relative values	http://indoor.hdrdb.com/
Laval HDR Outdoor	205	2Mpix (2:1 ratio)	Relative values	http://outdoor.hdrdb.com/
Akyuz HDR Images	10	5MPix	Relative values	https://user.ceng.metu.edu.tr/~akyuz/hdrdisp_eval/hdrdisp_project.html
Debevec HDR Images	21	0.3-2Mpix	Relative values	https://www.pauldebevec.com/
MPI HDR Images	7	3MPix	Scene-referred	https://resources.mpi-inf.mpg.de/hdr/gallery.html
Classic HDR Images	10	<1Mpix	Relative values	https://www.cs.huji.ac.il/w~danix/hdr/results.html
Funt HDR Dataset	105	3Mpix	Scene-referred	https://www2.cs.sfu.ca/~colour/data/funt_hdr/

HDR Video Datasets

Dataset Name	#Videos	#Resolution	Length	FPS	Color Space	Format	Website
Stuttgart HDR Dataset	33	1920×1080	13s-100s	24/25	REC709	Floating Point	https://www.hdm-stuttgart.de/vmlab/projects/
UBC HDR Video Dataset	10	2048×1080	7s-10s	30	REC709	Floating Point	http://dml.ece.ubc.ca/data/DML-HDR/
LIVE HDR Video Quality Assessment Database	31 (310 at different bit-rates)	0.32Mpix	3s-10s	50/60	BT2020	HDR10	https://live.ece.utexas.edu/research/LIVEHDR/LIVEHDR_index.html
MPI HDR Video Dataset	2	0.3Mpix	24s-34s	24	REC709	Floating Point	https://resources.mpi-inf.mpg.de/hdr/video/
EBU HDR Video Dataset	10	3996×2160	10s-31s	50	BT2100	HLG	https://tech.ebu.ch/testsequences

HDR Content Datasets

- Are these tables complete?
  - No, they are not.
- There are more datasets, but it can happen they may be not be available for some time. For example:
  - LiU HDR Video Dataset: high-quality dataset that is not currently available on the web.
  - MPEG HDR Video Dataset: not freely available.
  - …

Augmentation Strategies

- Classic flips and rotations;

- Cropping from high-resolution images;

- Channel swapping [Kalantari et al. 2017]:
  - RGB channels are randomly swapped;

Creating Images for Training

- The training dataset:
  - <Input SDR, Output HDR>
- How do we compute the input?

$$Z = f(E \cdot \delta t)$$

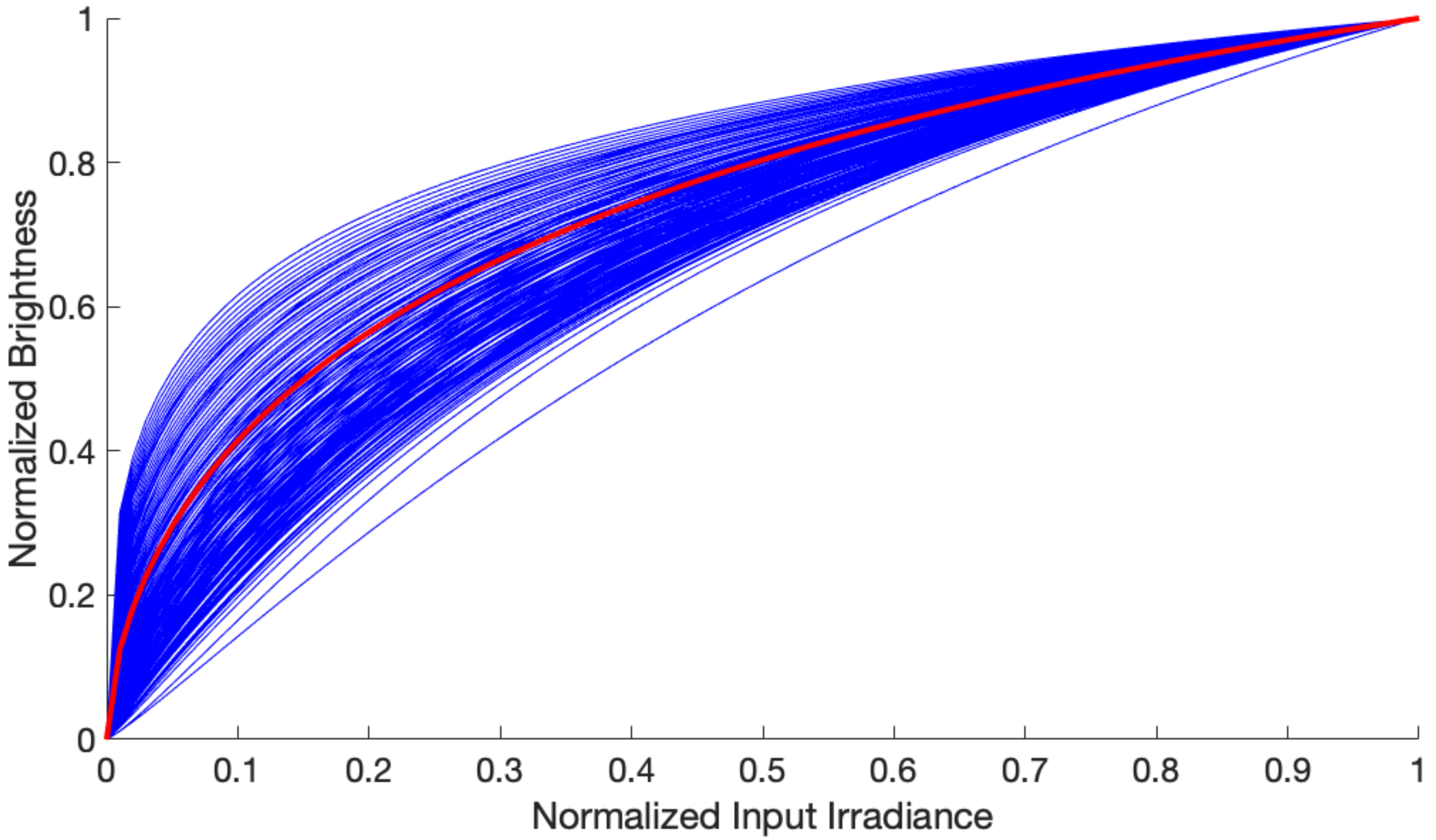
- δt is the virtual exposure value.
- $f(x)$ is the camera response function where the simplest to be used is:

$$f(x) = x^{\frac{1}{2.2}}$$

Creating Images for Training

- Many methods employs a random function from Grossberg and Nayar 2003 dataset of CRFs:
  - Eilertsen et al. 2017 showed that meaningful CRF can be modeled as:

$$f(x) = (1+\sigma)\cdot\frac{x^n}{x^n+\sigma} \qquad n \sim \mathcal{N}(0.9, 0.1) \quad \sigma \sim \mathcal{N}(0.6, 0.1)$$



Creating Images for Training

- δt is an important value to be picked up:
  - Its range is $[1/I_{\min}, 1/I_{\max}]$
- Automatic exposure:
  - $\delta t = \dfrac{1}{4 I_{\text{mean}}}$
  - We pick the δt that maximizes the well-exposed pixels in the range $[0.05, 0.95]$:
    - We do not want too dark images.

Creating Images for Training

- We may perform a random augmentation:

$$\delta t \sim [1/I_{\min}, 1/I_{\max}]$$

- In this case, we need to skip extremely bright and dark images:
  - These are difficult cases.
  - We need a minimum of well-exposed pixels in order to draw something of meaningful from our methods:
    - 50-75% of well-exposed pixels:
      - Half/Quarter of the image totally white or totally black.

Selecting Patches

- Eilertsen et al. 2017:
  - For each HDR, 10 patches are selected at 320×320 using random cropping.
    - Lee et al. uses random crops at 256×256
- Endo et al. 2017:
  - Images are downsampled at 512×512.
- Marnerides et al. 2018:
  - Random crop with Gaussian distribution (center image) at 384×384.
- Santos et al. 2020:
  - Selection of patches with texture; i.e., mean gradient of the detail layer over 0.85 (bilateral separation).

Training

The Loss Function

- Eilertsen et al. 2017:
  - MSE in the log domain.
  - We have a loss function for the luminance and the reflectance component:
    - Equal weight in the paper for both losses.
- Marnerides et al. 2018:
  - L1 + Cosine Loss (for colors in under-exposed areas):

$$\mathscr{L}_{\cos}(\hat{I}, I) = 1 - \frac{1}{N} \sum_{i,j} \frac{\hat{I}(i,j) \cdot I(i,j)}{\|\hat{I}(i,j)\|_2 \cdot \|I(i,j)\|_2} \,,$$

where I is the reference image and $\hat{I}$ is the results of the network.

The Loss Function

- Lee et al. 2018 employs as content loss L_1 and classic GAN loss:

$$\mathcal{L}_{\text{GAN}}(D) = \frac{1}{2}\mathbb{E}_{x,y}[(D(y,x)-1)^2] + \frac{1}{2}\mathbb{E}_{x,z}[(D(G(y,z),x))^2]$$

$$\mathcal{L}_{\text{GAN}}(G) = \mathbb{E}_{x,z}[(D(G(y,z),x)-1)^2]$$

$$\mathcal{L}_{L_1}(G) = \mathbb{E}_{x,y,z}[\|y-G(x,z)\|_1]$$

- Wang et al. 2019, Santos et al. 2020, Liu et al. 2020 uses a perceptual loss (VGG network) together with L_1:

$$\mathcal{L}_P(I,\hat{I}) = \|\psi(I)-\psi(\hat{I})\|_2$$

- Liu et al. 2020 has a complex loss where the main contribution is the reconstruction loss (L_1) TV loss and a CRF loss (MSE)

Videos

What's about video?

- There are many papers treating videos:
  - In many cases, these works on a single frame:
    - There is no temporal coherence mechanisms in place:
      - **Not working on multiple frames at the same time;**
      - **No temporal loss;**
  - Why are these considered videos methods?
    - They use HDR10/HDR10+ video datasets with wide gamut (e.g., RECO2020 or REC2100 color space).
    - They output directly PQ/HLG values.
    - They work on YUV input values.

What's about video? Video Stabilization

- Eilertsen et al. 2019 showed how to make imaging method temporal coherent: colorization, inverse tone mapping etc.

- The key is the introduction of a new loss:

$$\mathscr{L}(I, \hat{I}) = \mathscr{L}_{\text{rec}}(I, \hat{I}) \cdot (1 - \alpha) + \alpha \mathscr{L}_{\text{reg}}(I, \hat{I})$$

where $\alpha \in [0.85, 0.95]$.

- Given that it is difficult to have good video dataset, the idea is to approximate a "video movement" by a small Euclidian Transformation T, which can be: a translation, a rotation, and a scaling.

What's about video?

- If our network is $f(\cdot)$ and its input I_{in} we can define the regularization as:

$$\mathcal{L}_{\text{reg}}(I,\hat{I}) = \mathcal{L}_{\text{reg}}(I, f(I_{in})) = \left\| \boxed{\Big(f(T(I_{in})) - T(I)\Big)} - \boxed{\Big(f(I_{in}) - I\Big)} \right\|_2$$

The difference between ground-truth and the network results after T; i.e., the "next frame"

The difference between ground-truth and the network results.

- $T(\cdot)$ is a random transformation:
  - Translation $[-2,2]^2$ pixels;
  - Rotation $\pm 1°$;
  - Scaling $[0.97,1.03]$;

Evaluation

Evaluation

- Main metrics recommended for evaluations are [Hanji et al. 2022]:
  - If we have a reference:
    - HDR-VDP 2.2, HDR-VDP 3.0.6, and PU21-PSNR.
  - If we do not have a reference:
    - PU21-PIQE, and PU-VSI.
- To focus evaluation on the generated content, we should remove influence of the CRF. A possibility is to estimate the CRF using the reference (if available).

Future Directions

The Status

- Currently, 2-3 new methods appears every month on arXiv!
- Many works just get old or new datasets and they train the latest architecture on them:
  - Diffusion networks;
  - Transformers;
  - etc.

Promising Approaches

- The main limitations of doing HDR and especially inverse tone mapping is that datasets are very small:
  - There are a small amount of images achieving 20-stops.
  - The few datasets may disappear due to maintenance!
- On the other hand there are large datasets available online of SDR image that could be used to copy well-exposed data in over-exposed areas:
  - Banterle et al. 2021: unsupervised generation of HDR videos from SDR videos.
  - Wang et al. 2022: unsupervised generation of HDR images from SDR images.

Questions?

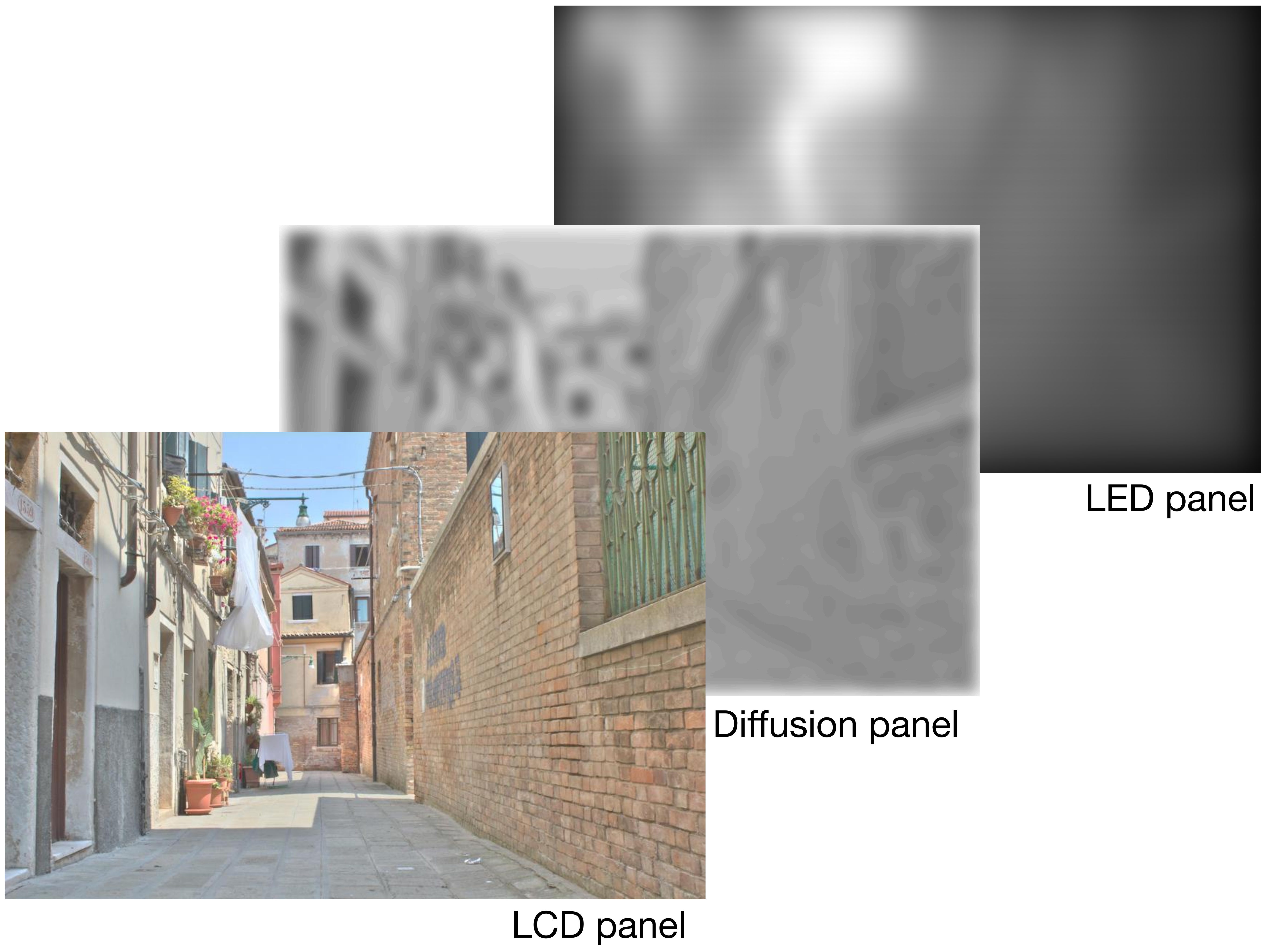
Modern High Dynamic Range Imaging at the Time of Deep Learning

Visualisation

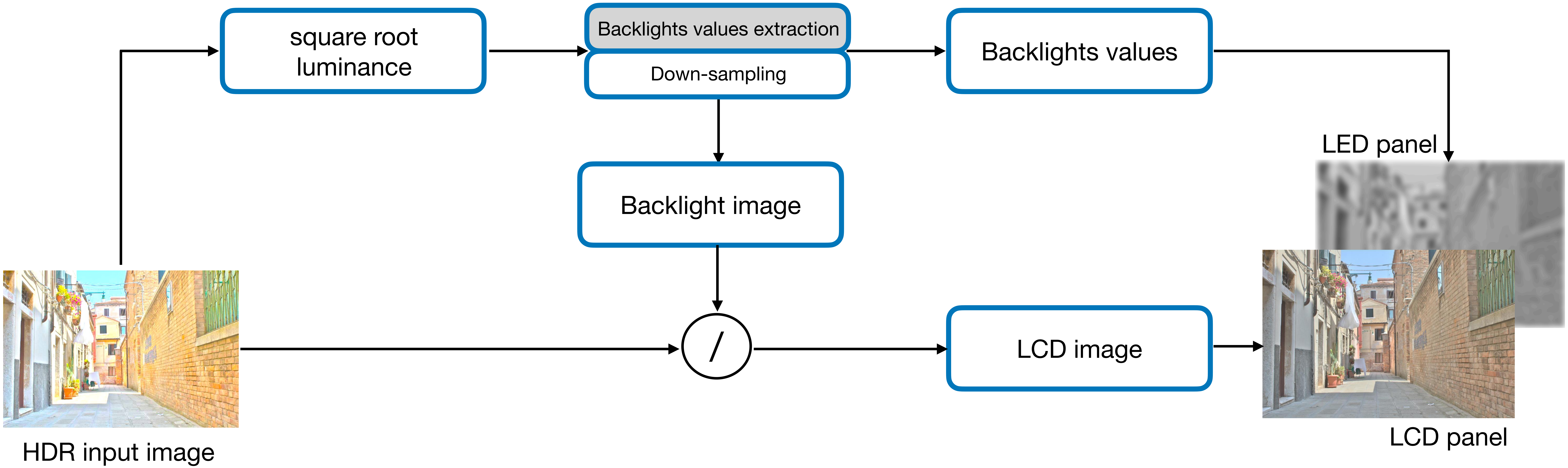
Francesco Banterle and Alessandro Artusi

HDR Direct Visualization:
HDR Displays

HDR Display: Modulating Backlight

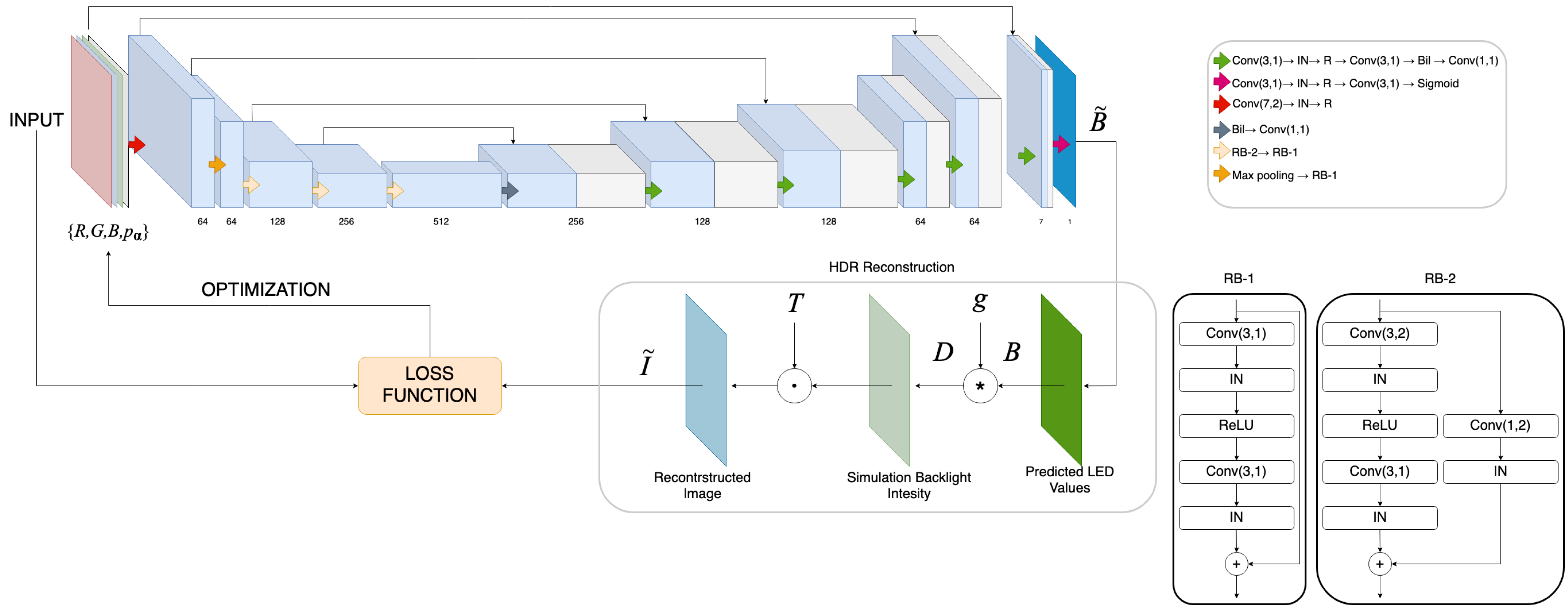


Baseline Method for Backlight Display



L. Duan , K. Debattista, Z. Lei and A. Chalmers, “Subjective and Objective Evaluation of Local Dimming Algorithms for HDR Images”, IEEE ACCESS, VOL. 8, MARCH 2020

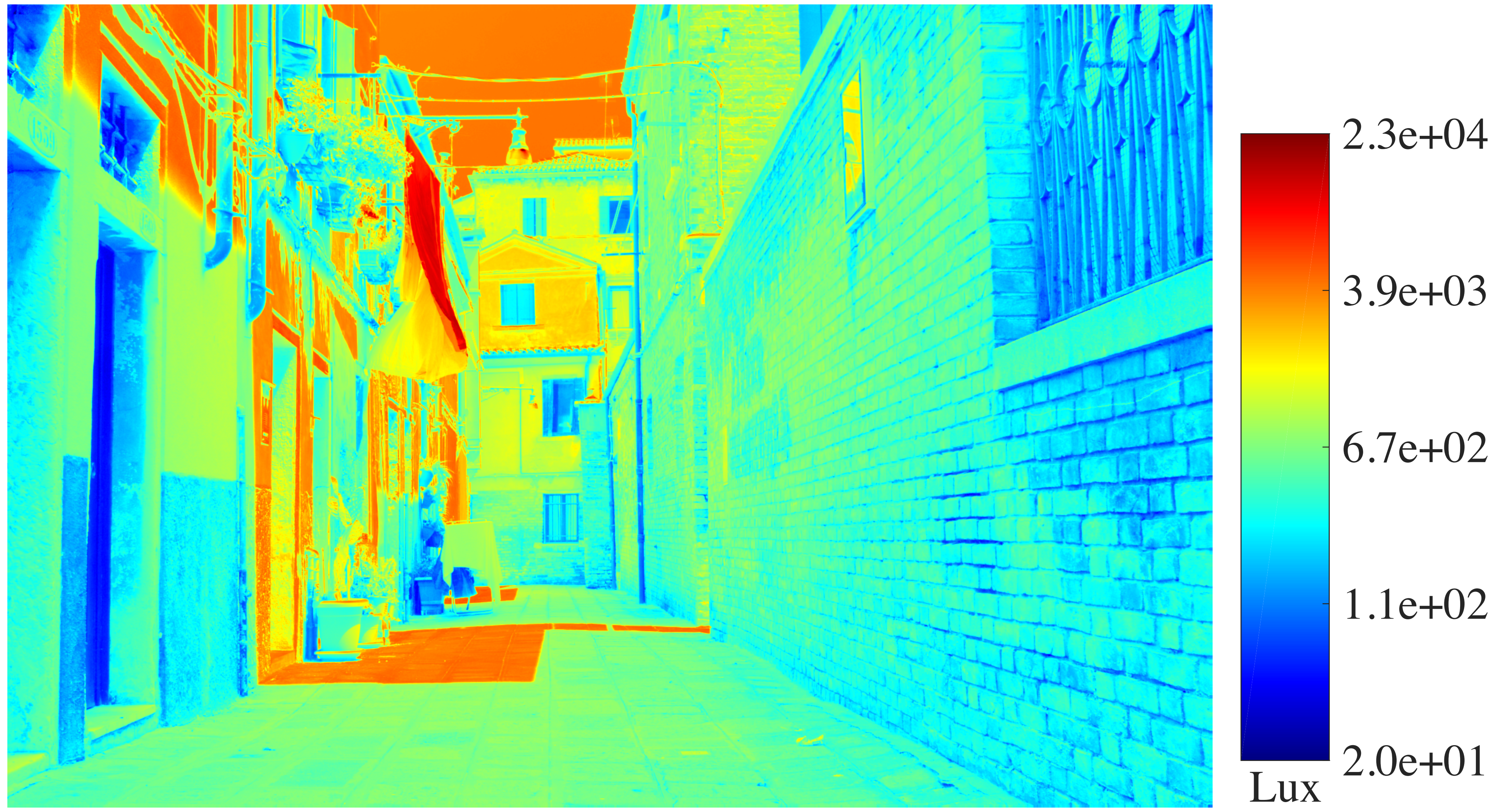
Deep-learning Approach for BLD



Source: L. Duan , D. Marnerides , A. Chalmers , Z. Lei , and K. Debattista, “Deep Controllable Backlight Dimming for HDR Displays”, IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, VOL. 68, NO. 3, AUGUST 2022

HDR Conversion to SDR Content: Tone Mapping

Tone Mapping



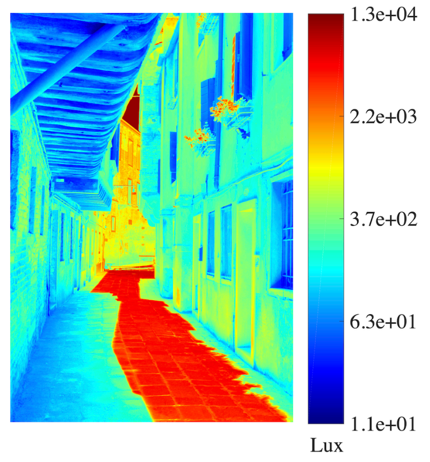
32-bit Scene-referred HDR image

TMO

8-bit Tone Mapped Image

The Full Pipeline

The Full Pipeline



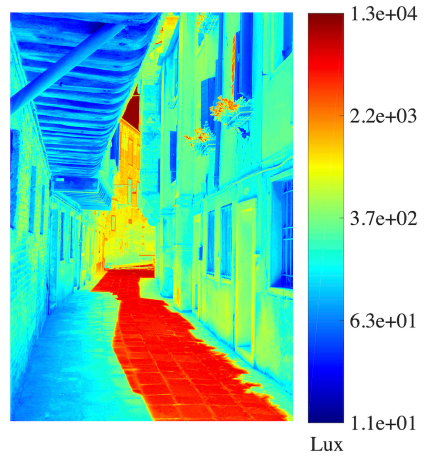
RGB → Y

Luminance mapping

Color mapping

$$Y_{HDR} = w_1 R + w_2 G + w_3 B$$

The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

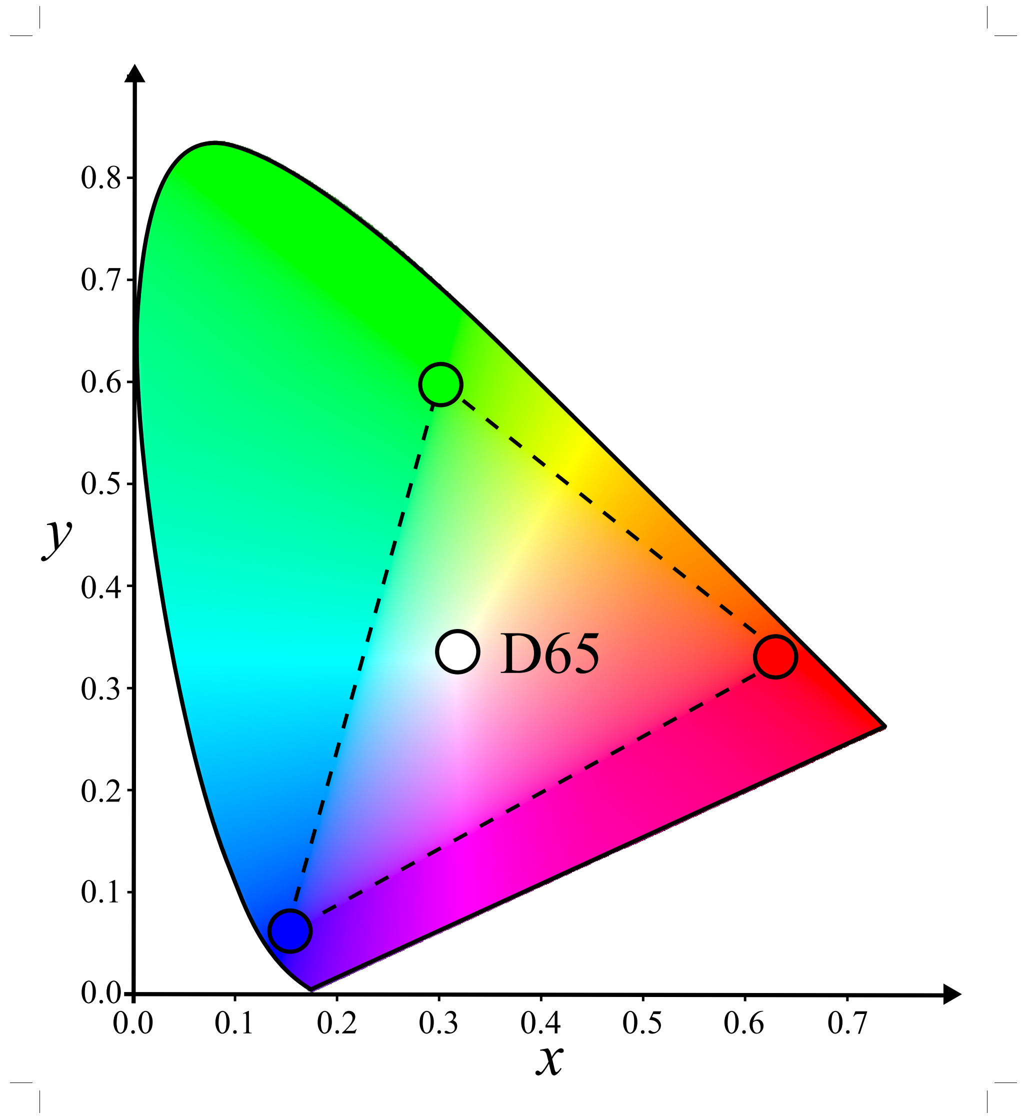
$$Y_{HDR} = w_1 R + w_2 G + w_3 B$$

$$sRGB:$$

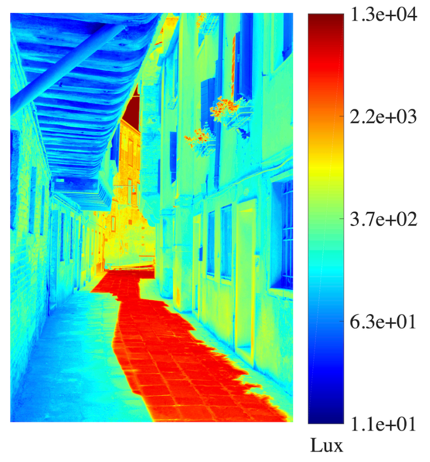
$$w_1 = 0.2126$$

$$w_2 = 0.7152$$

$$w_3 = 0.0722$$



The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

$$Y_{HDR} = w_1 R + w_2 G + w_3 B$$

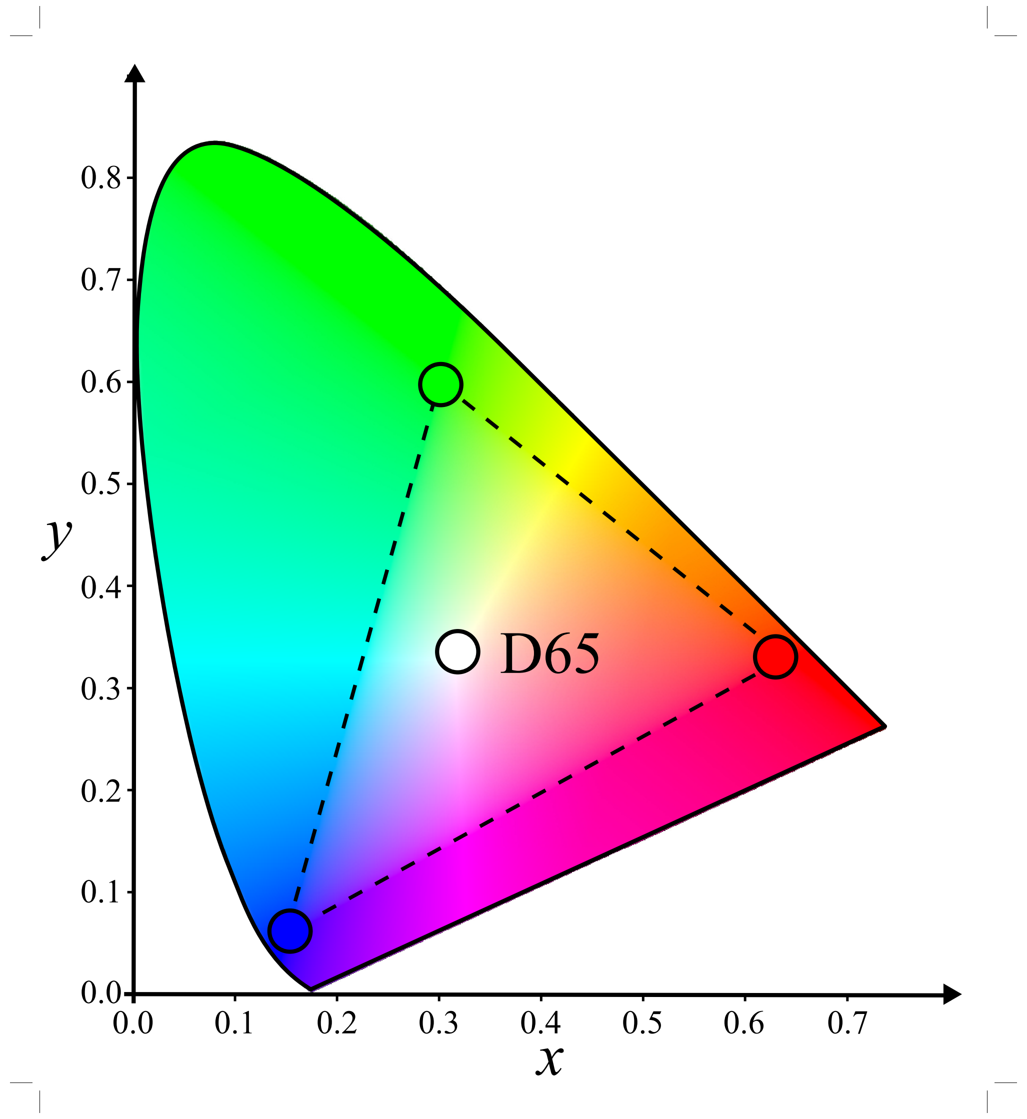
$$Y_{SDR} = F(mY^{\gamma}_{HDR})$$

$sRGB:$

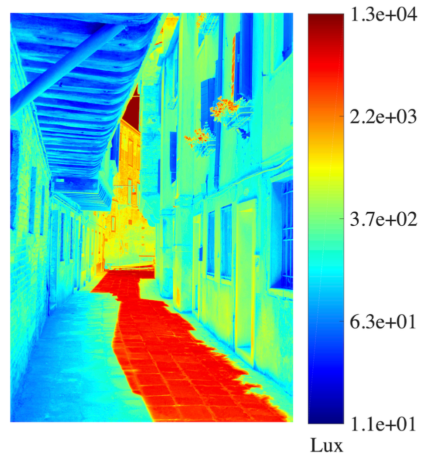
$w_1 = 0.2126$

$w_2 = 0.7152$

$w_3 = 0.0722$



The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

$$Y_{HDR} = w_1 R + w_2 G + w_3 B$$

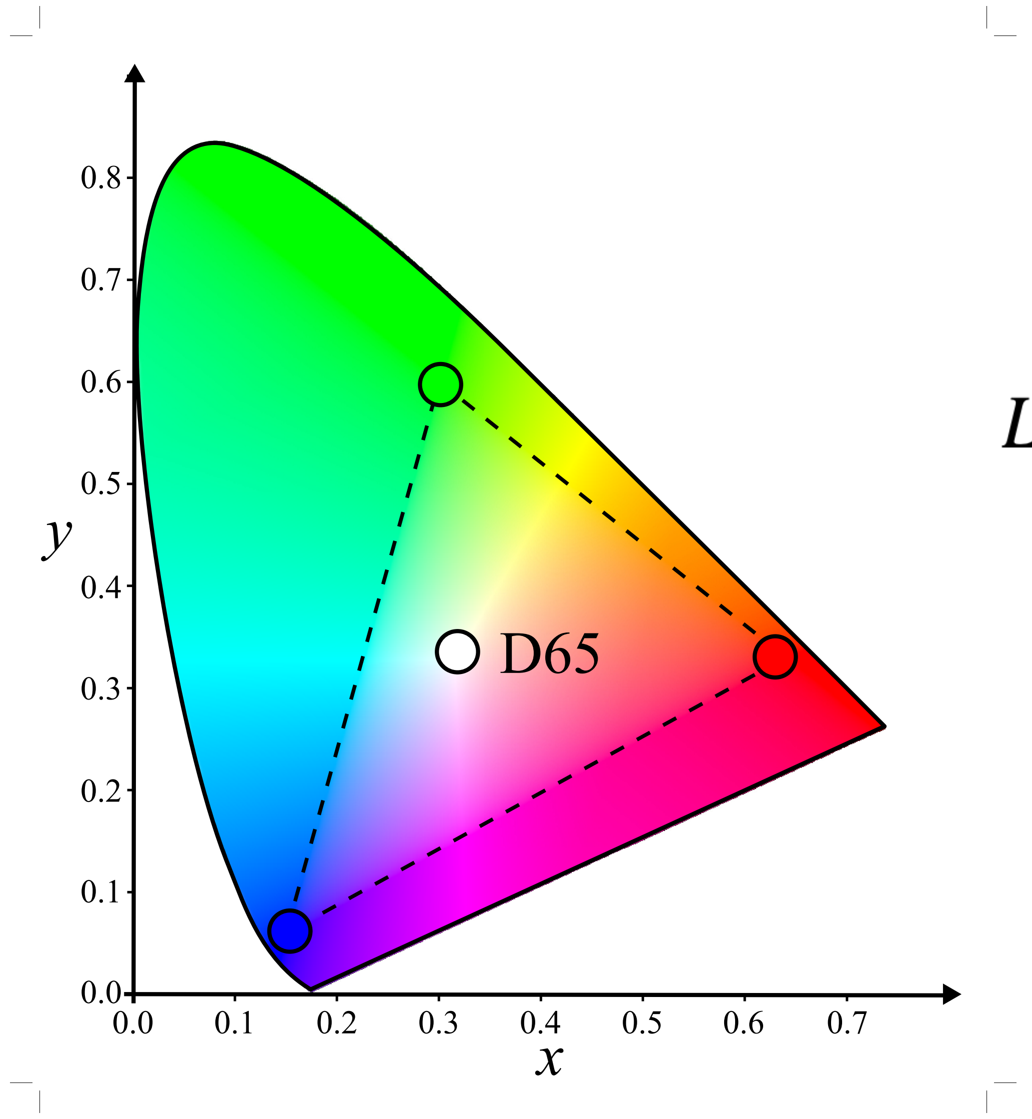
$$Y_{SDR} = F(mY^{\gamma}_{HDR})$$

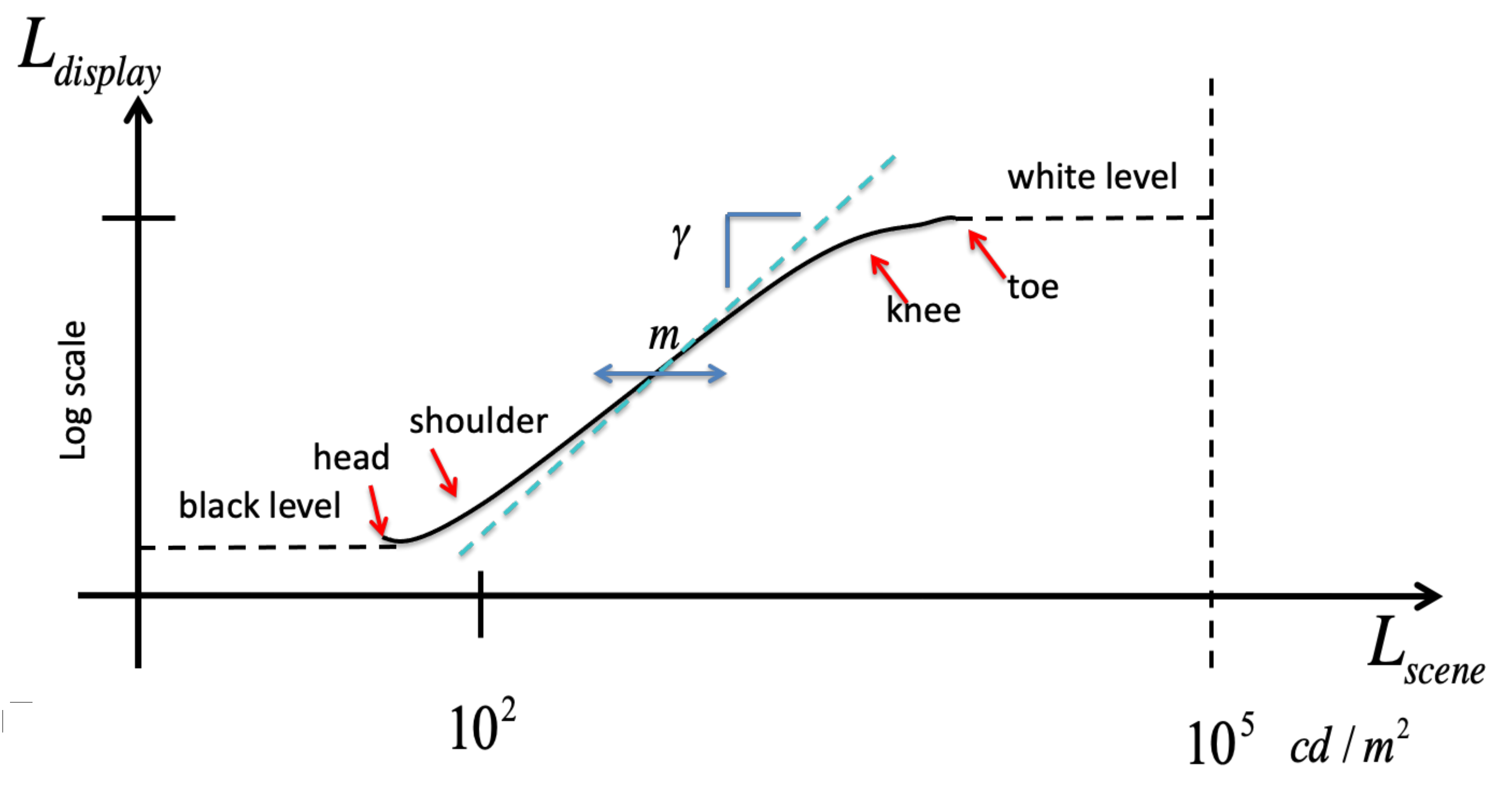
$sRGB:$

$w_1 = 0.2126$

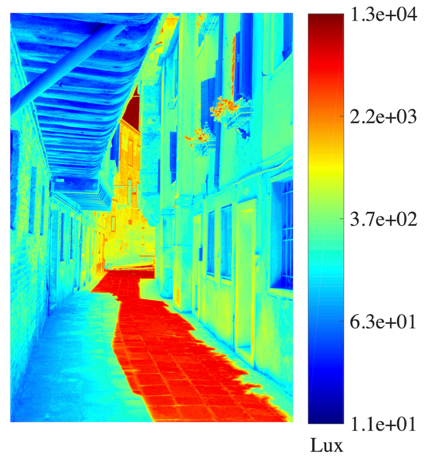
$w_2 = 0.7152$

$w_3 = 0.0722$





The Full Pipeline



RGB → Y → Luminance mapping → Color mapping →

$$Y_{HDR} = w_1 R + w_2 G + w_3 B$$

$$Y_{SDR} = F(mY_a^{\gamma})$$

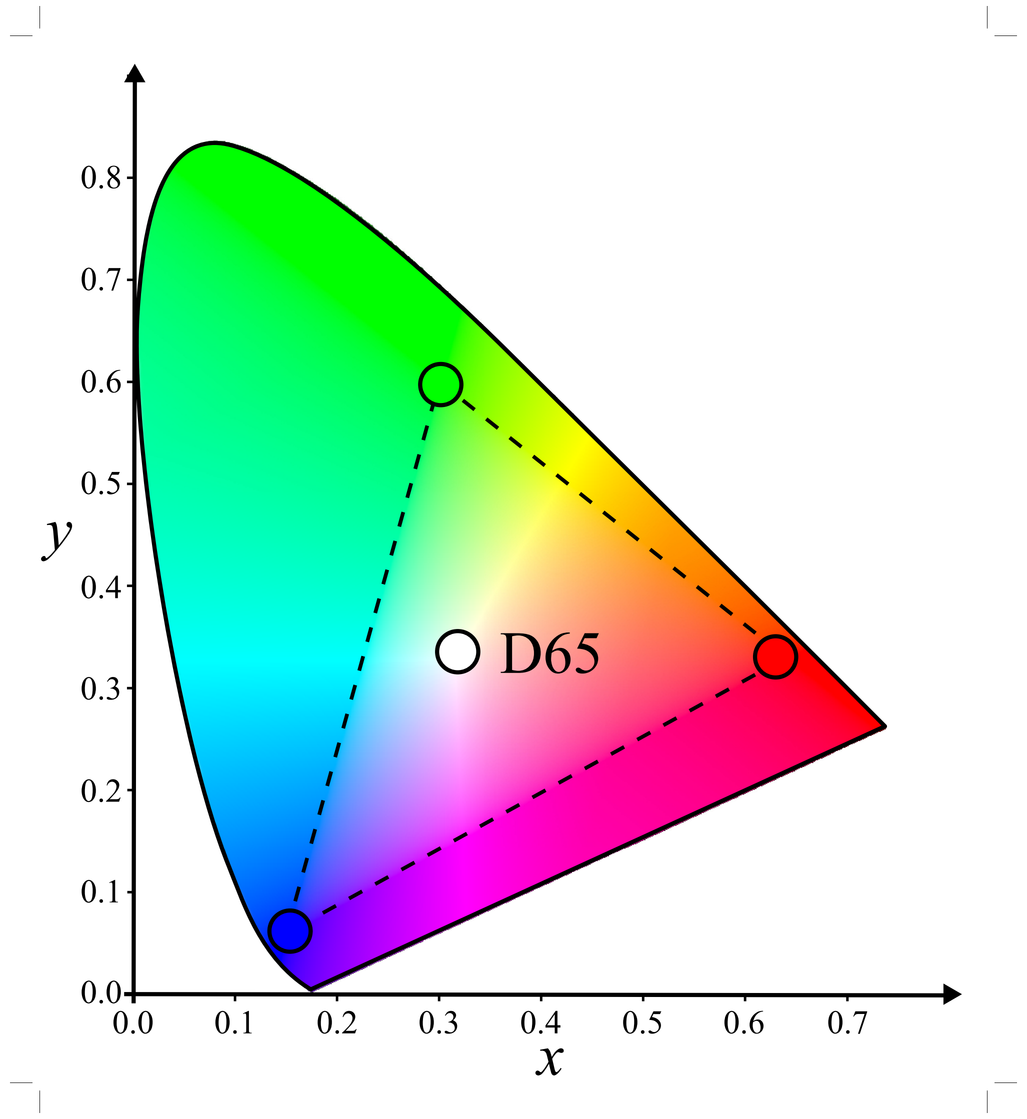
$$Y_a = G_s(Y_{HDR})$$

$sRGB:$

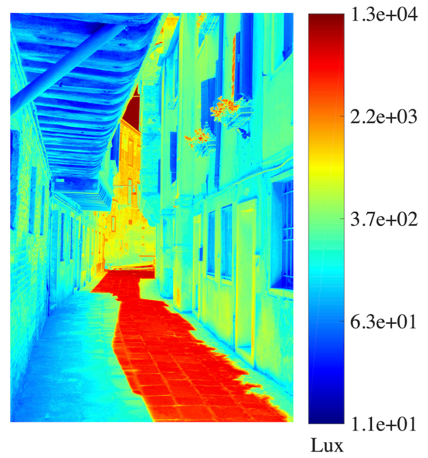
$w_1 = 0.2126$

$w_2 = 0.7152$

$w_3 = 0.0722$



The Full Pipeline



RGB → Y → Luminance mapping → Color mapping →

$$Y_{HDR} = w_1 R + w_2 G + w_3 B$$

$$Y_{SDR} = F(mY_a^\gamma)$$

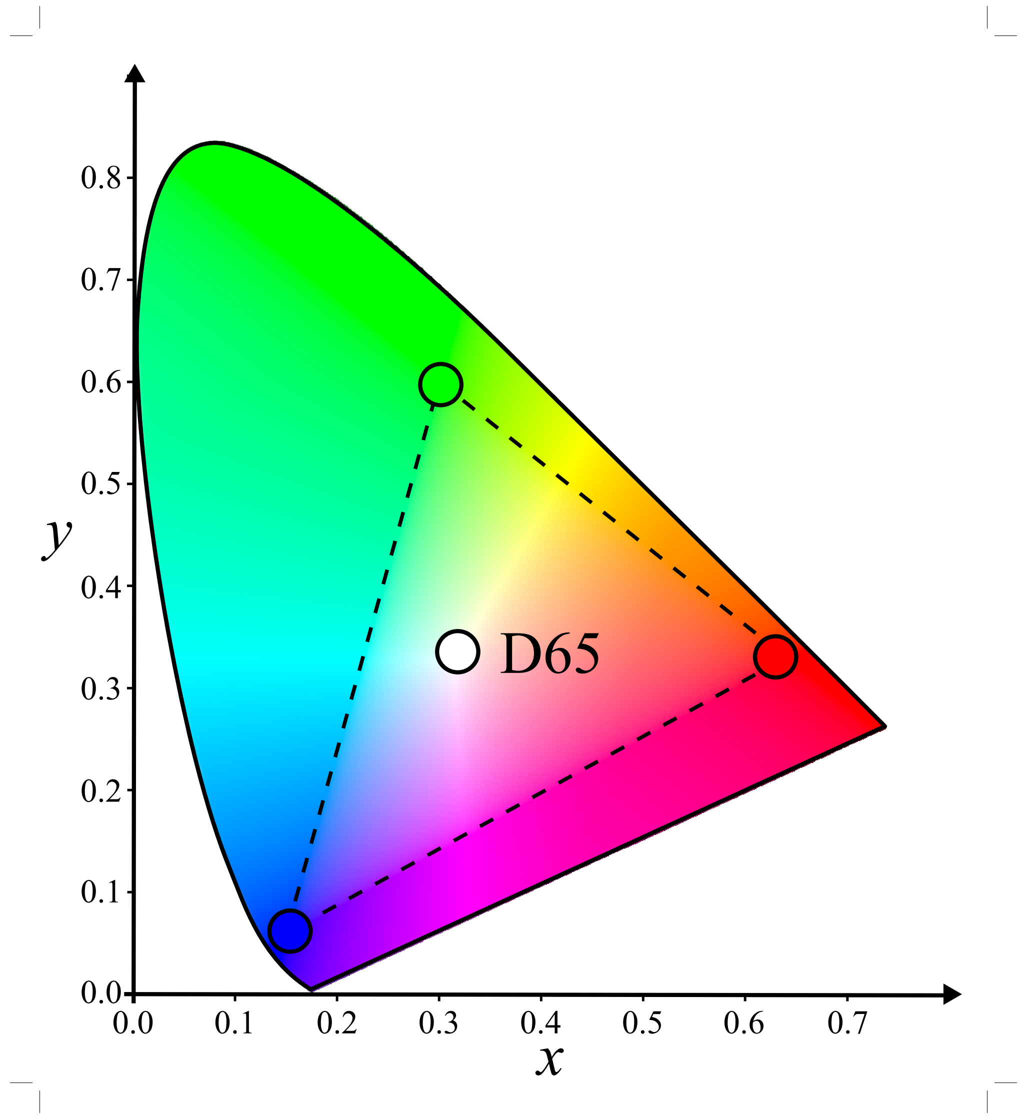
$$Y_a = G_s(Y_{HDR})$$

$sRGB:$

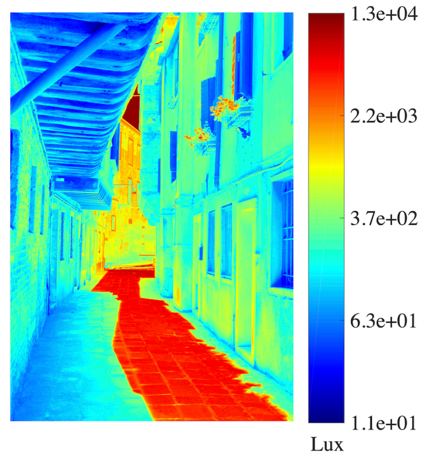
$w_1 = 0.2126$

$w_2 = 0.7152$

$w_3 = 0.0722$



The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

$$Y_{HDR} = w_1 R + w_2 G + w_3 B$$

$$Y_{SDR} = F(mY_a^{\gamma})$$

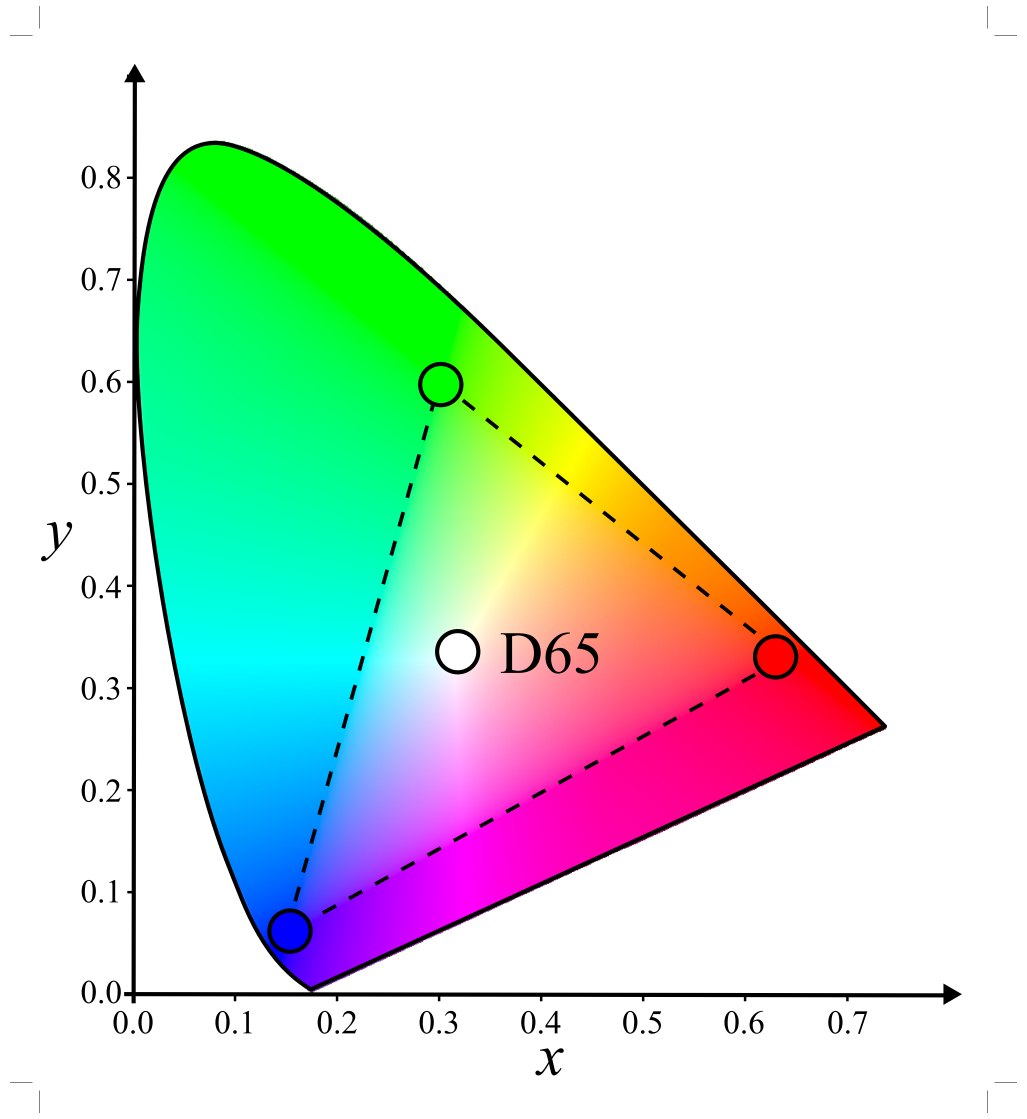
$$Y_a = G_s(Y_{HDR})$$

$sRGB:$

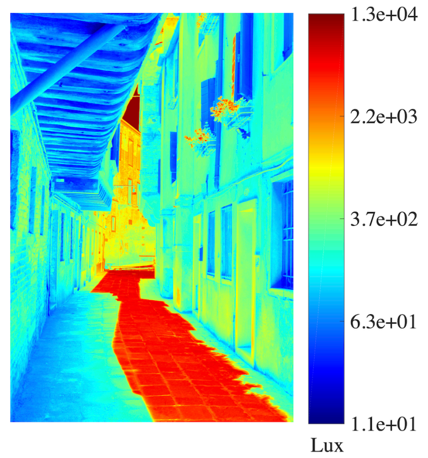
$$w_1 = 0.2126$$

$$w_2 = 0.7152$$

$$w_3 = 0.0722$$



The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

$$Y_{HDR} = w_1 R + w_2 G + w_3 B$$

$$Y_{SDR} = F(mY_a^{\gamma})$$

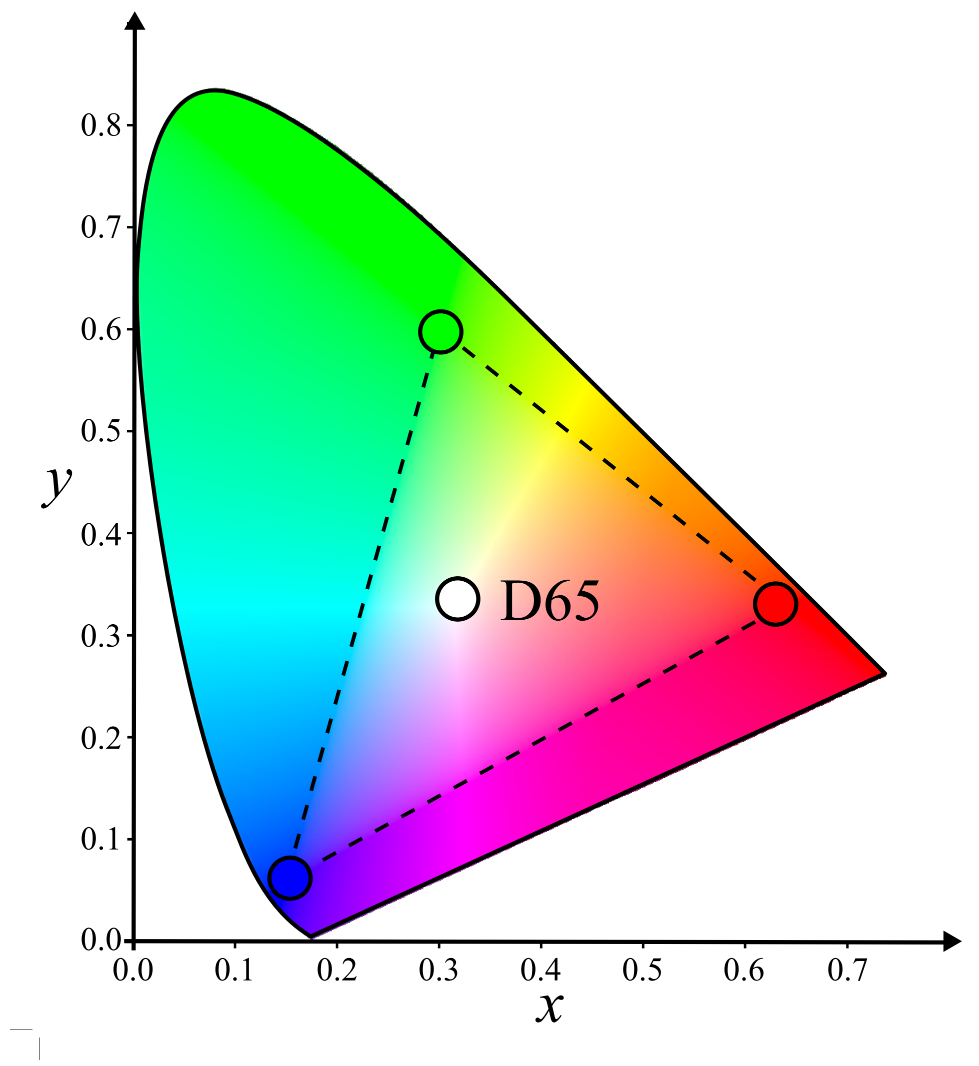
$$Y_a = G_s(Y_{HDR})$$

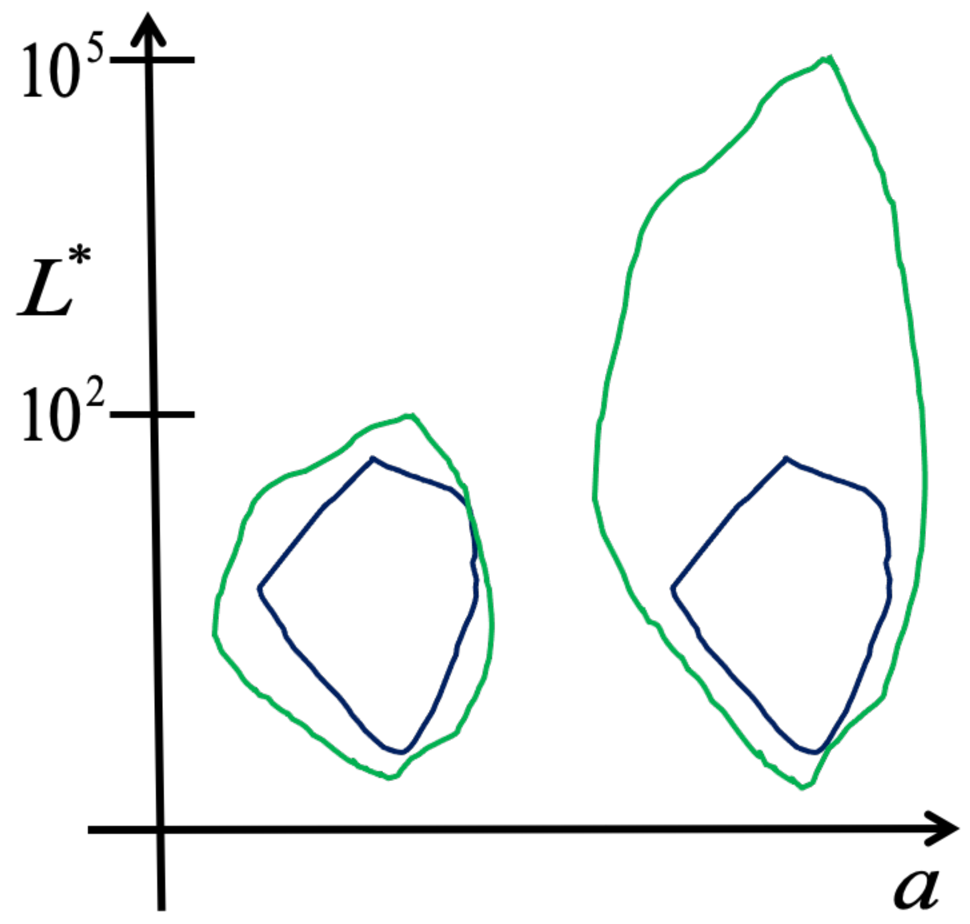
$sRGB:$

$w_1 = 0.2126$

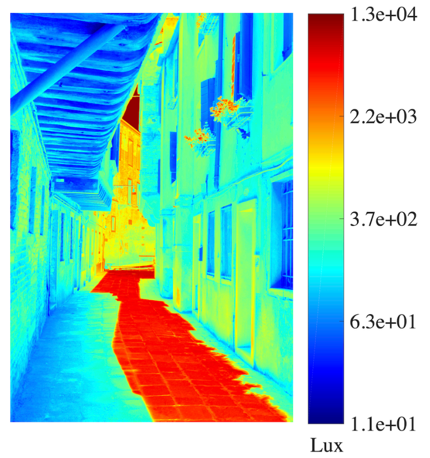
$w_2 = 0.7152$

$w_3 = 0.0722$





The Full Pipeline



RGB → Y → **Luminance mapping** → **Color mapping**

$$Y_{HDR} = w_1 R + w_2 G + w_3 B$$

$$Y_{SDR} = F(mY_a^{\gamma})$$

$$Y_a = G_s(Y_{HDR})$$

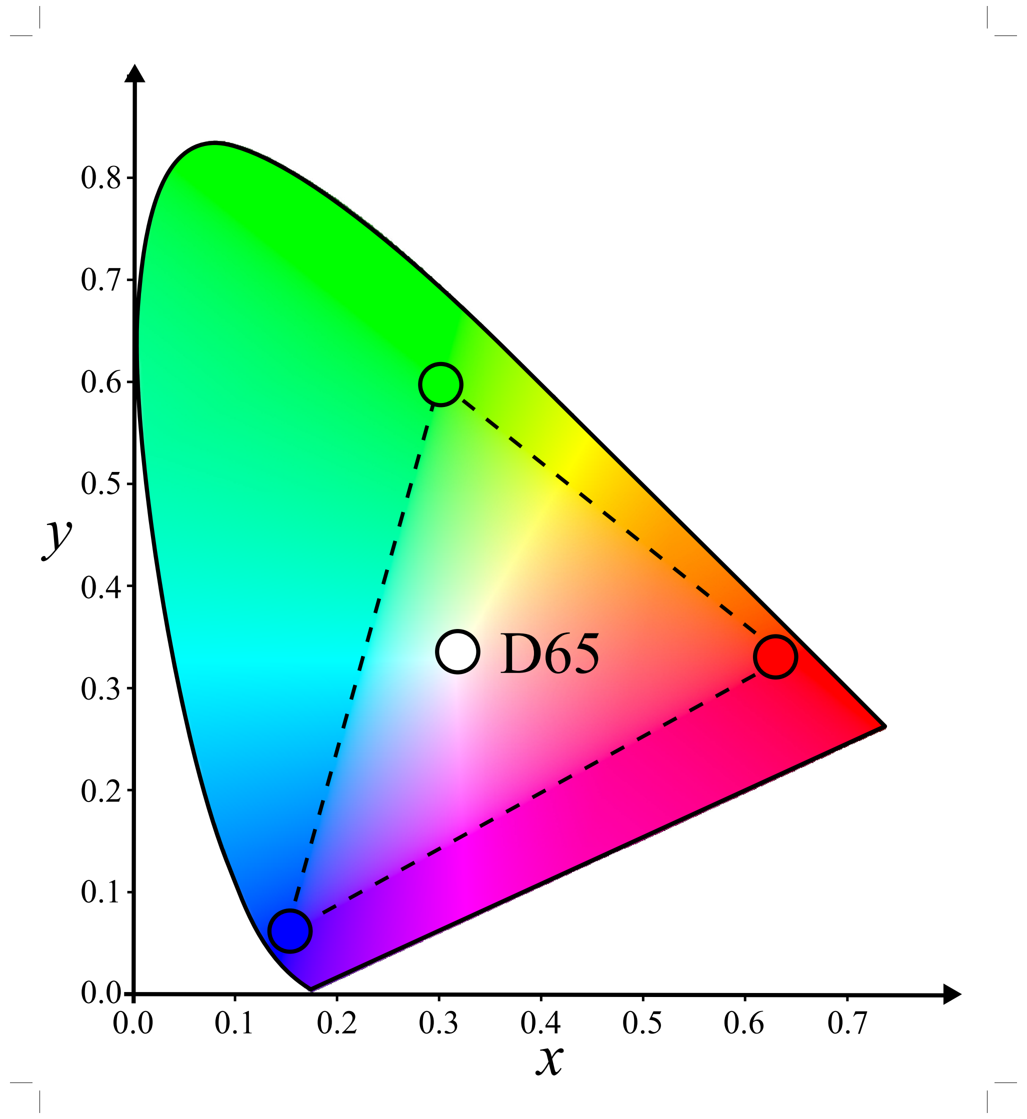
$$RGB_{SDR} = \left(\frac{RGB_{HDR}}{Y_{HDR}}\right)^s Y_{SDR}$$

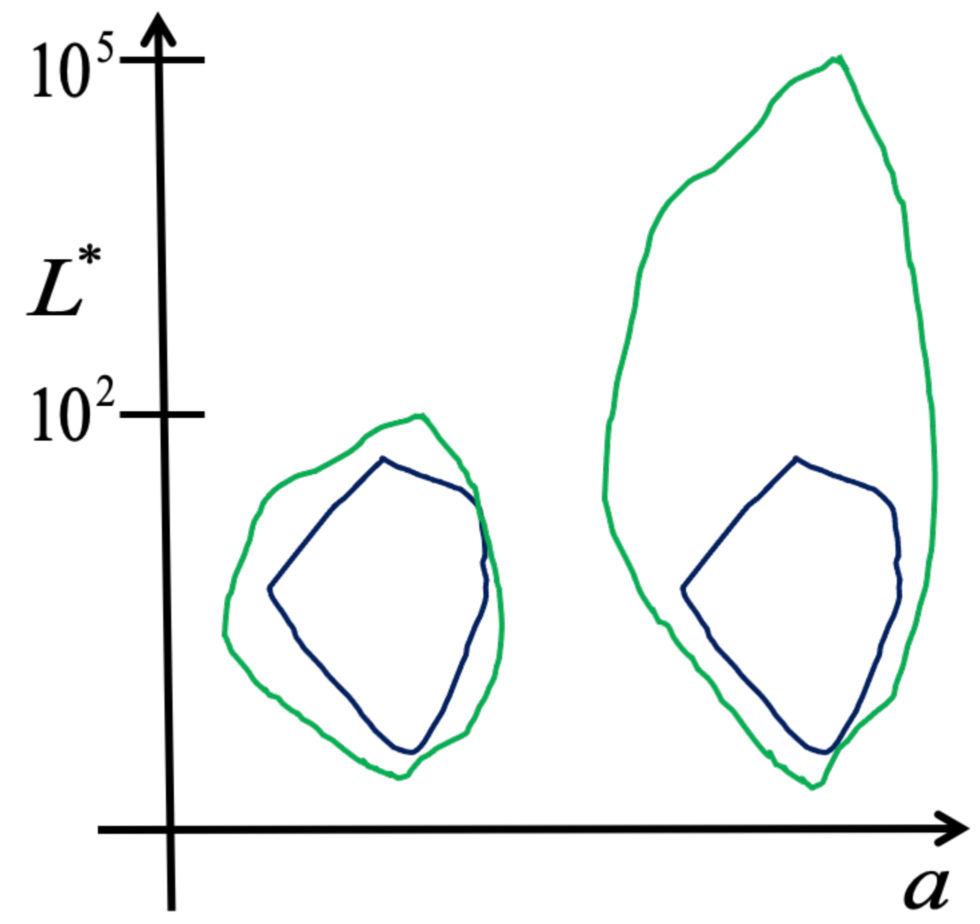
$sRGB:$

$w_1 = 0.2126$

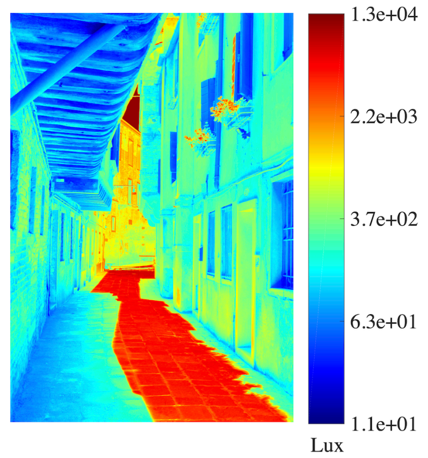
$w_2 = 0.7152$

$w_3 = 0.0722$





The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

$$Y_{HDR} = w_1 R + w_2 G + w_3 B$$

$$Y_{SDR} = F(mY_a^\gamma)$$

$$Y_a = G_s(Y_{HDR})$$

$$RGB_{SDR} = \left(\frac{RGB_{HDR}}{Y_{HDR}}\right)^s Y_{SDR}$$

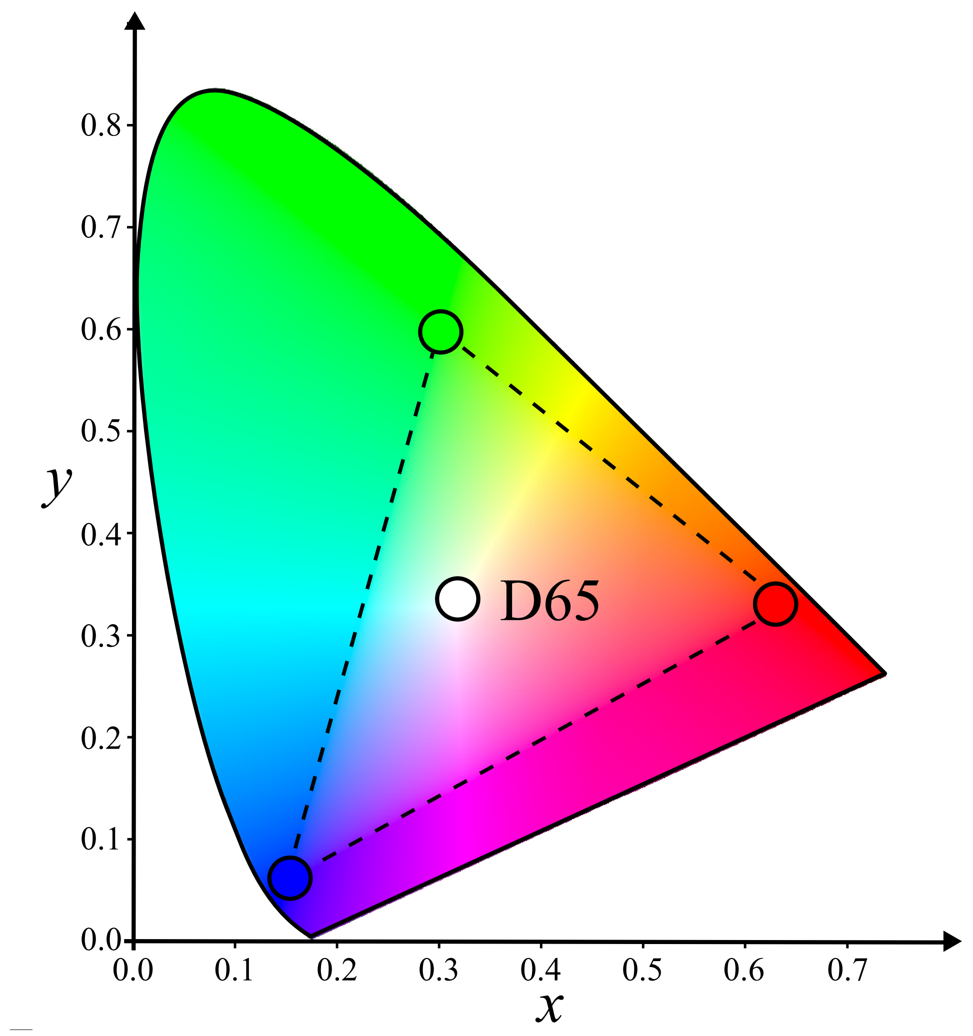
$$RGB_{SDR} = \left(\left(\frac{RGB_{HDR}}{Y_{HDR}} - 1\right)s + 1\right)^s Y_{SDR}$$

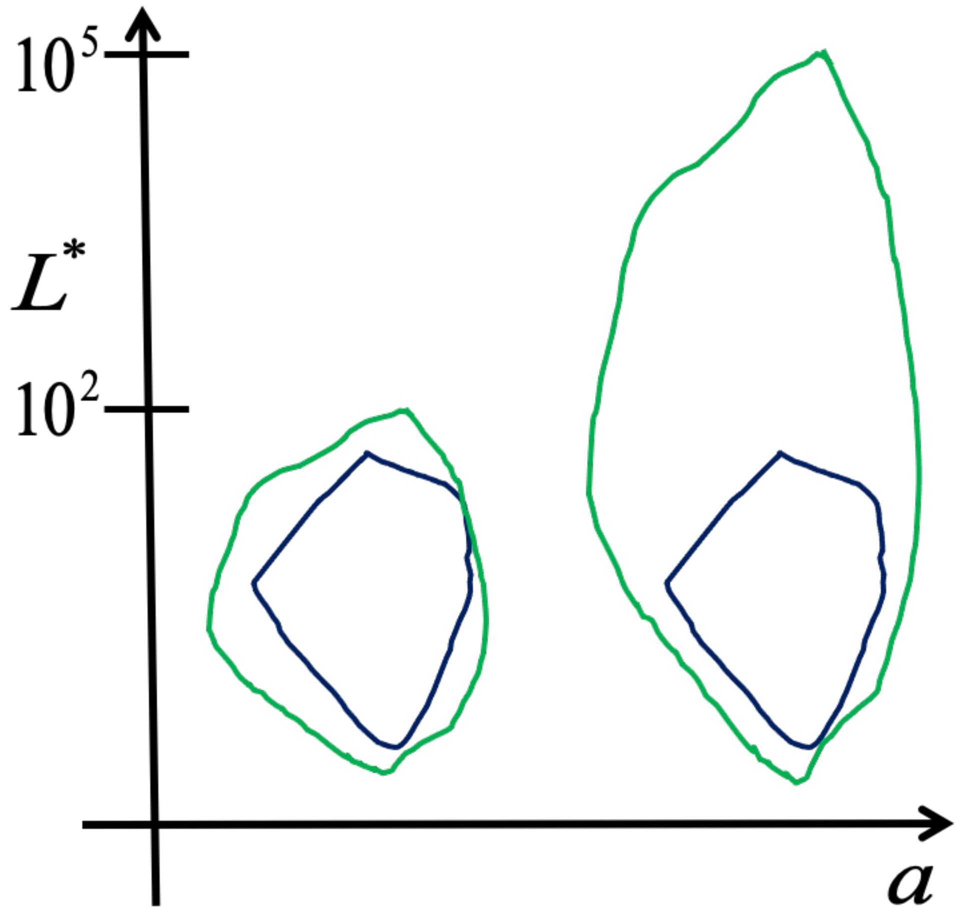
$sRGB:$

$w_1 = 0.2126$

$w_2 = 0.7152$

$w_3 = 0.0722$





Aims/Goals

Aims/Goals

- **Quality optimisztion**
  - To best reproduce the characteristics of the LDR image (Cite:VHF 2021)
  - To mimic the original HDR content under a limited range [0-255] (DeepTMO Cite:RSV 2020)
  - Learning-based self-supervised TMO (Cite:WSC 2022)
  - Fusing stack of n differently exposed LDR images (DeepFuse Cite:DF2017)
  - Optimising color mapping using HSV (TMNet Cite:ZWZW 2020)
- **Performances optimisation**
  - Parameters free TMO (TMO-net Cite:PKO 2021)
  - Real-time DL based TMO (Cite:ZZWW 2022)

Architectures

Architectures - Generative Adversarial Network

Legend:

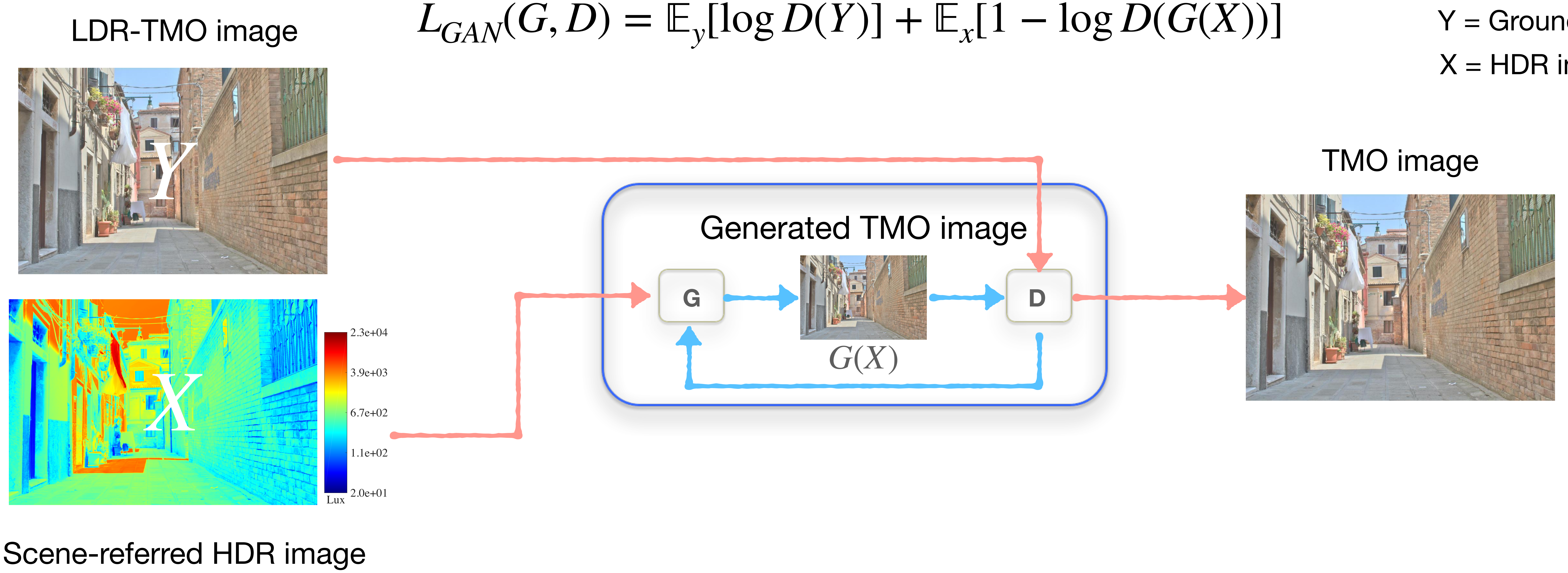
G = Generator

D = Discriminator

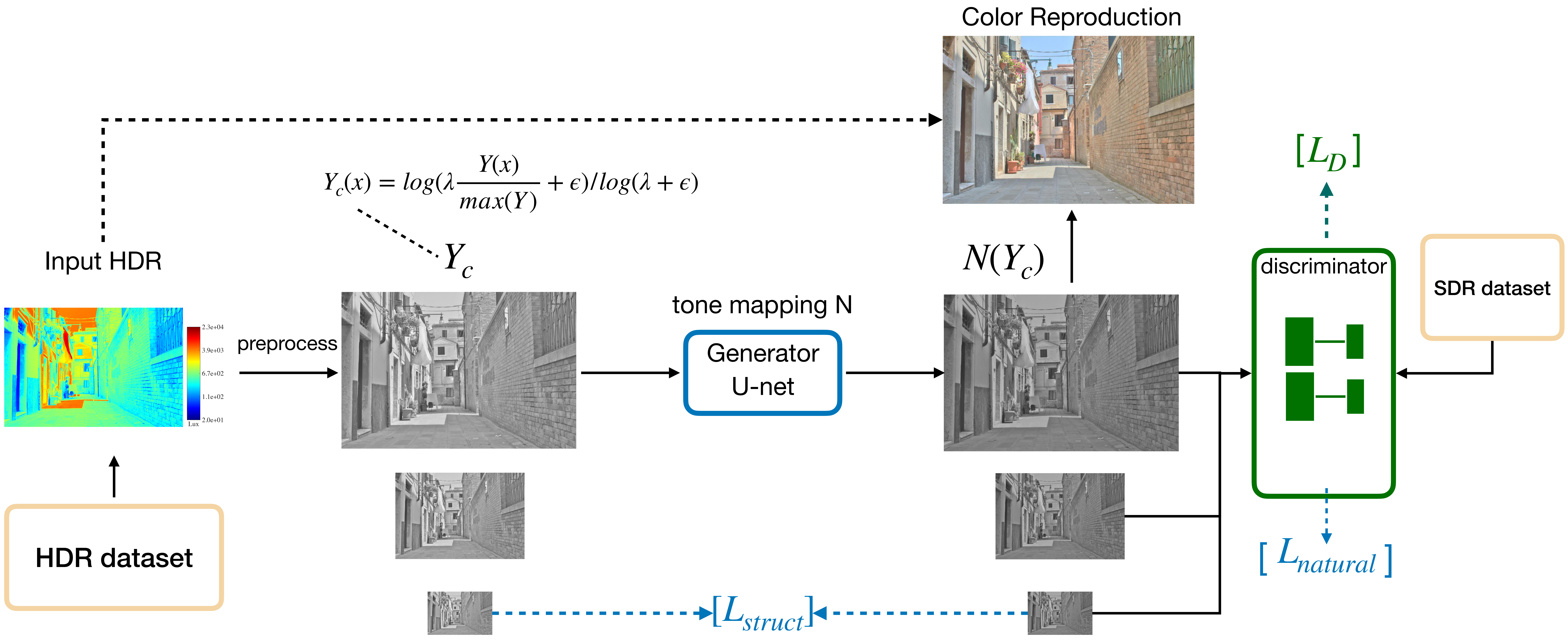
Y = Ground truth SDR

X = HDR input

$$L_{GAN}(G, D) = \mathbb{E}_y[\log D(Y)] + \mathbb{E}_x[1 - \log D(G(X))]$$



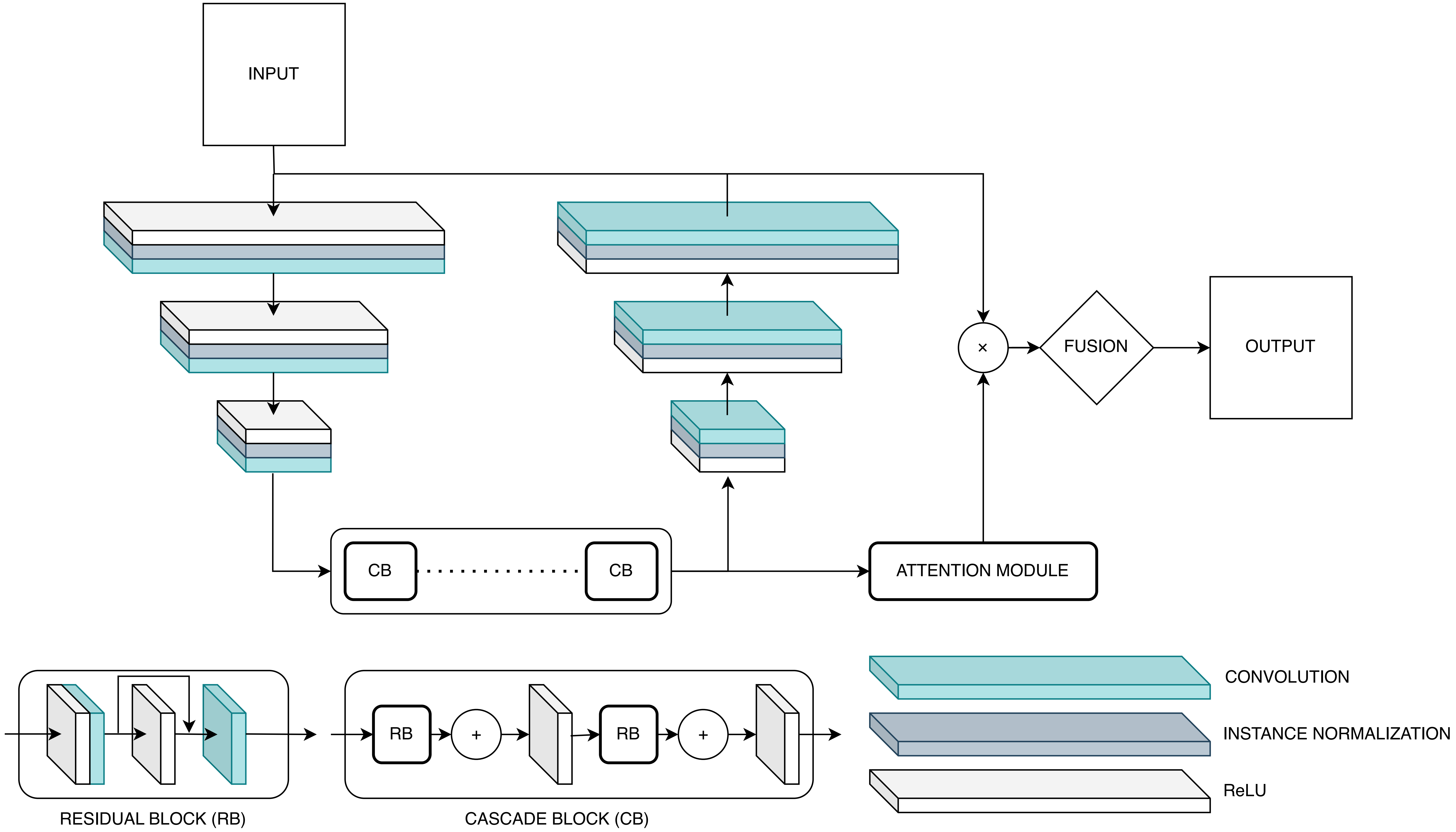
Architectures - Generative Adversarial Network Ref: VHF-2021



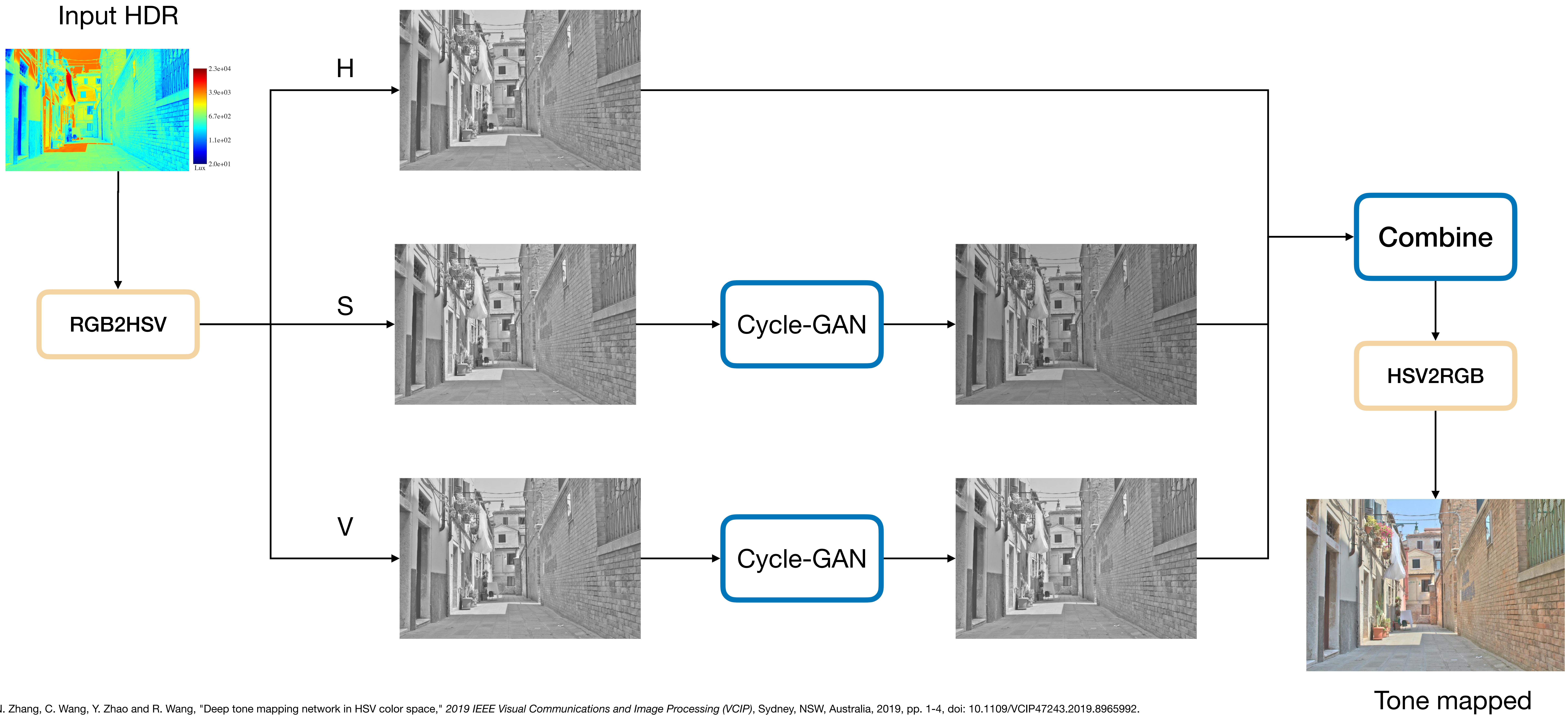
VINKER Y., HUBERMAN-SPIEGELGLAS I., FATTAL R.: Unpaired learning for high dynamic range image tone mapping. In 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021 (2021), IEEE, pp. 14637–14646. URL: https://doi.org/10.1109/ICCV48922.2021.01439.

Architectures - Generator U-net modified Ref: PKO-2021

- Attention module
  - Channel and pixels-wise;
  - Ensuring that the generator
    - Global/local contrast;
    - Color consistency;
    - Eliminate under/over-exposure (image synthesis)



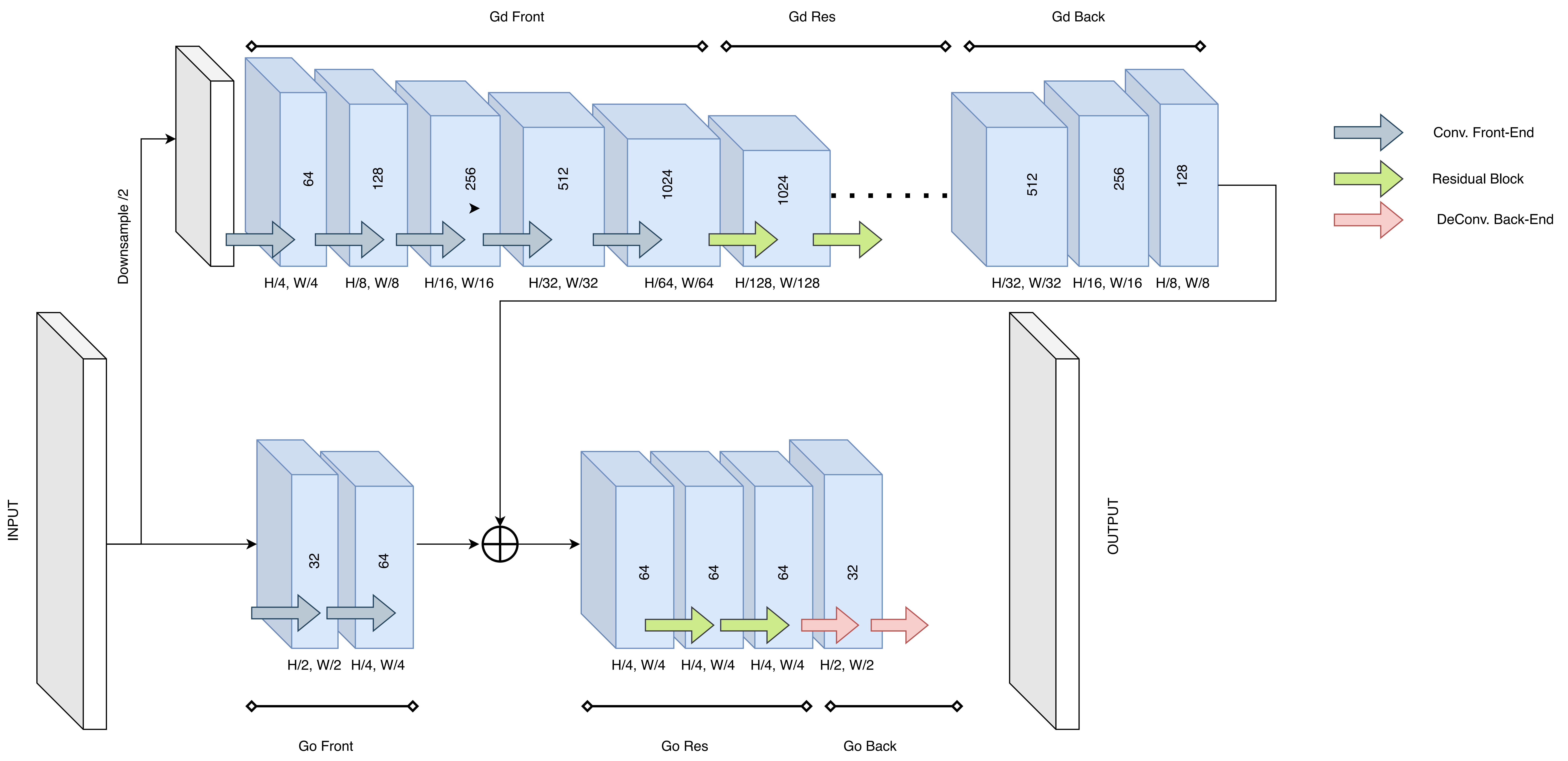
Architectures - Cycle-GAN Approach Ref: ZZWW-2020-2022



N. Zhang, C. Wang, Y. Zhao and R. Wang, "Deep tone mapping network in HSV color space," *2019 IEEE Visual Communications and Image Processing (VCIP)*, Sydney, NSW, Australia, 2019, pp. 1-4, doi: 10.1109/VCIP47243.2019.8965992.

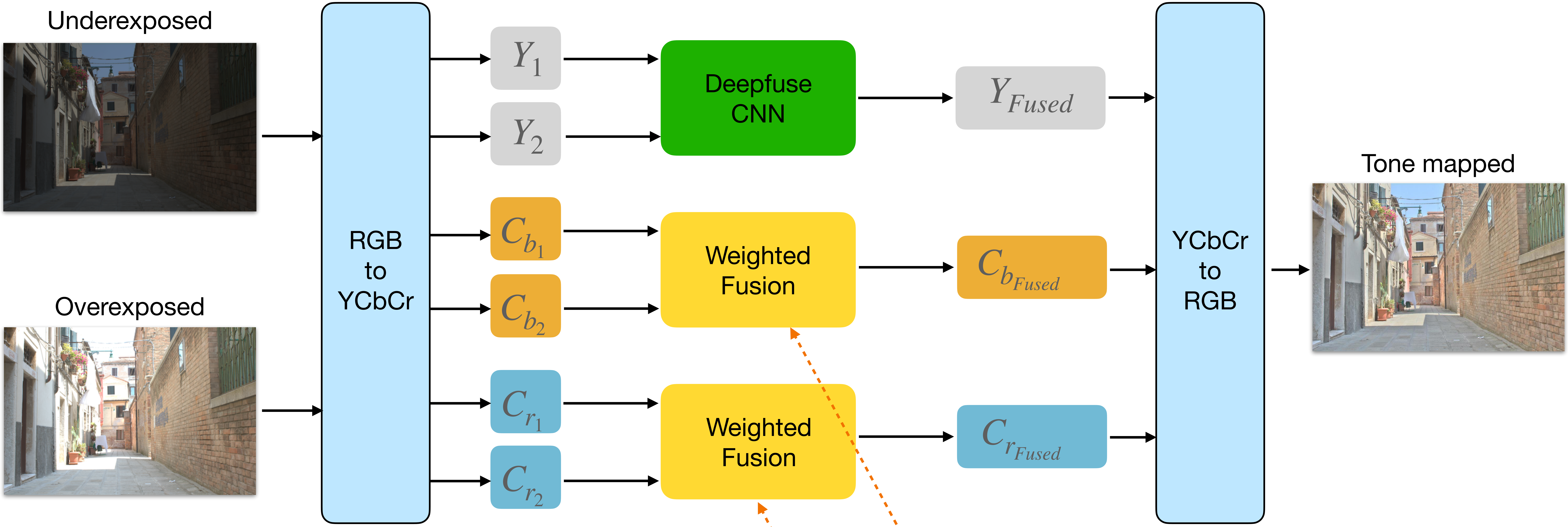
ZHANG N., ZHAO Y., WANG C., WANG R.: A real-time semi-supervised deep tone mapping network. IEEE Trans. Multim. 24 (2022), 2815–2827. URL: https://doi.org/10.1109/TMM.2021.3089019, doi:10.1109/TMM.2021.3089019. 2

Architectures - Multi-Scale Generator Ref: RSV-2020



A. Rana, P. Singh, G. Valenzise, F. Dufaux, N. Komodakis and A. Smolic, "Deep Tone Mapping Operator for High Dynamic Range Images," in *IEEE Transactions on Image Processing*, vol. 29, pp. 1285-1298, 2020, doi: 10.1109/TIP.2019.2936649.

Architectures - Convolutional Neural Network Ref: DF-2017



$x_i = C_{b_i}, C_{r_i}$

$x = C_{b_{Fused}}, C_{r_{Fused}}$

$$x = \frac{x_1\left(|x_1 - \tau|\right) + x_2\left(|x_2 - \tau|\right)}{|x_1 - \tau| + |x_2 - \tau|}$$

K. R. Prabhakar, V. S. Srikar and R. V. Babu, "DeepFuse: A Deep Unsupervised Approach for Exposure Fusion with Extreme Exposure Image Pairs," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 4724-4732

Architectures - DeepFuse CNN Ref: DF-2017

K. R. Prabhakar, V. S. Srikar and R. V. Babu, "DeepFuse: A Deep Unsupervised Approach for Exposure Fusion with Extreme Exposure Image Pairs," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 4724-4732, doi: 10.1109/ICCV.2017.505.

Architectures - Autoencoder Ref: WCS-2022

Training

Generative Adversarial Approach

The Loss Function- General Approach

$$Loss = L_{adv} + \sum_{i=1,..n} L_i$$

Adversarial Loss

- Image appearance matching

Other Loss functions

- To preserve the content and structure
- pixel-wise loss.
- perceptual loss, features matching, gradient

The Loss Function Ref: VHF-2021

- Vinker et al. 2021:
  - Not training the generator (N) to conceive new images from scratch
    - Removing biases in Y_c with respect to regular LDR images
  - Discriminator, 3 applied at different image scale ($\downarrow^k$ bicubic downscaling $\times 2^k$)

$$L_D = \sum_{k \in 0,1,2} \left(\mathbb{E}_{Y_{LDR}} \left[D_k(\downarrow^k Y_L) - 1 \right]^2 + \mathbb{E}_{Y_{HDR}} \left[D_k(\downarrow^k N(Y_c)) \right]^2 \right)$$

  - D_k is used to improve the ability of the generator (N) to match the natural appearance

$$L_{natural} = \sum_{k \in 0,1,2} \left(\mathbb{E}_{Y_{HDR}} \left[D_k(\downarrow^k N(Y_c) - 1) \right]^2 \right)$$ adversarial loss

The Loss Function Ref: VHF-2021

- Vinker et al. 2021:
  - To preserve the content and structure
    - measure based on Pearson correlation on two images I, J

$$\rho(I,J) = \frac{1}{n_p} \sum_{p_I, p_J} \frac{cov(p_I, p_J)}{\sigma(p_I)\sigma(p_j)} \qquad p_I, p_J = 5 \times 5 \quad pixels$$

    - Loos function

$$L_{struct} = \sum_{k \in 0,1,2} \rho\left(\downarrow^k Y_c, \downarrow^k N(Y_c)\right)$$

The Loss Function Ref: PKO-2021

- Panetta et al. 2021: min-max adversarial loss

$$Loss = L_{adv} + \lambda_1 L_{FM} + \lambda_2 L_{VGG} + \lambda_3 L_{GPL}$$

- Perceptual loss:

$$L_{VGG} = \sum_{i=1,\ldots N} \frac{1}{M_i} \left[|| F^{(i)}(Y) - F^{(i)}(G(X)) ||_1 \right]$$

$F^{(i)}$ i^{th} layer of the VGG19 network

M_i i^{th} element of the layer

- Feature matching loss:

$$L_{FM} = \mathbb{E}_{X,Y} \sum_{i=1,\ldots T} \frac{1}{N_i} \left[|| D^{(i)}(Y) - D^{(i)}(G(X)) ||_1 \right]$$

T total number of layers

N_i number of elements in each layer

$D^{(i)}$ i^{th} layer feature extractor of the discriminator

The Loss Function Ref: PKO-2021

- Gradient profile loss:
    - Preserve edge information between the ground truth and synthetic SDR images

$$L_{GPL}(Y,\hat{Y}) = \sum_{c}\left(\frac{1}{H}trace\left(\nabla\ G(\hat{Y})_c \cdot \nabla\ \hat{Y}_c^{\tau}\right) + \frac{1}{W}trace\left(\nabla\ G(\hat{Y})_c^{\tau} \cdot \nabla\ Y_c\right)\right)$$

$(\cdot)^{\tau}$ Transpose operator

$Y, \hat{Y}$ are the ground truth and the synthetic SDR images

H, W height and width of the image

The Loss Function Ref: RSV-2020

- Rana et al. 2020: min-max adversarial loss

$$Loss = \sum L_{adv} + \beta \sum L_{FM} + \lambda L_{VGG}$$

- Perceptual loss (same as PKO 2021):

$$L_{VGG} = \sum_{i=1,\ldots N} \frac{1}{M_i} \left[|| F^{(i)}(y) - F^{(i)}(G(x)) ||_1 \right]$$

$F^{(i)}$ i^{th} layer of the VGG19 network

M_i i^{th} element of the layer

- Feature matching loss (same as PKO 2021):

$$L_{FM} = \mathbb{E}_{X,Y} \sum_{i=1,\ldots T} \frac{1}{N_i} \left[|| D^{(i)}(y) - D^{(i)}(G(x)) ||_1 \right]$$

T total number of layers

N_i number of elements in each layer

$D^{(i)}$ i^{th} layer feature extractor of the discriminator

The Loss Function Ref: ZZWW-2022

- Zhang et al. 2020: classic cycle loss and min-max adversarial loss for both luminance and saturation

$$Loss = \lambda L_1 + L_{adv_f} + L_{adv_b} + \beta(L_{cycle_f} + L_{cycle_b})$$

- Perceptual pixel loss L1 norm for both luminance and saturation:

$$L_{pixel} = \mathbb{E}(x, y)||G(x) - y||_1$$

Self-Supervised

The Loss Function Ref: DF-2017

- Prabhakar et al. 2017: based on SSIM objective metric (which it gives a score)

$$Loss = 1 - \frac{1}{N} \sum_{p \in P} Score(p)$$

- $Score(p)$: takes into account the contrast and the desired structure, the luminance is discharged (local luminance comparison in the patches is not significant):

$$Score(p) = \frac{2\sigma_{\tilde{y}y_f} + C}{\sigma_{\tilde{y}}^2 + \sigma_{y_f}^2 + C'}$$

N number of pixels in the image

P number of pixels in the patch

$\tilde{y}$ estimated patch

y_f fused patch

$\sigma_{\tilde{y}}$ σ_{y_f} variance

$\sigma_{\tilde{y},y_f}$ covariance

The Loss Function Ref: WCS-2022

- Wang et al. 2022: L1 norm based on VGG features maps

$$Loss = ||f(VGG(I_\mu)) - f(VGG(I_{TM}))||_1$$

$$f(VGG(I)) = \frac{M_s}{1 + M_n}$$

$$I_\mu = \frac{\log(1 + \mu I_{HDR})}{log(1 + \mu)}$$

Pre-processing HDR input image to transform it into VGG features space, i.e., VGG is trained using SDR images

$$I_{HDR} = 0.5 \times \frac{I_{src}}{mean(I_{src})}$$

Feature contrast neighborhood-masking

$$M_n = \frac{\sigma_b}{|\mu_b| + \epsilon}$$

Feature contrast self-masking

$$M_s = sign(C)|C|^\alpha$$

Feature contrast

feature magnitude at pixel p

gaussian filtered feature value for patch P centre at pixel p

$$C_p = \frac{f_p - \tilde{f}_p}{|\tilde{f}_p| + \epsilon}$$

Future Directions

Color Rendition

- It is based on a simple concept of keeping into the tone mapped image the original color ratio of the high dynamic range input image:

$$RGB_{SDR} = \left(\frac{RGB_{HDR}}{Y_{HDR}} \right)^{s} Y_{SDR}$$

- However, several color mapping techniques have been developed:
  - The main aim is to minimize the hue distortion
  - Color gamut mapping
  - Color retargeting: based on optimal saturation parameter

Computational and memory management costs

- Complex models
  - Complex architectures
  - High number of parameters
  - High memory management costs
- Reduces their applicability where we need fast response
- Natural question
  - How to reduce the model complexity while retaining similar quality performance?

Any Questions?

Modern High Dynamic Range Imaging at the Time of Deep Learning

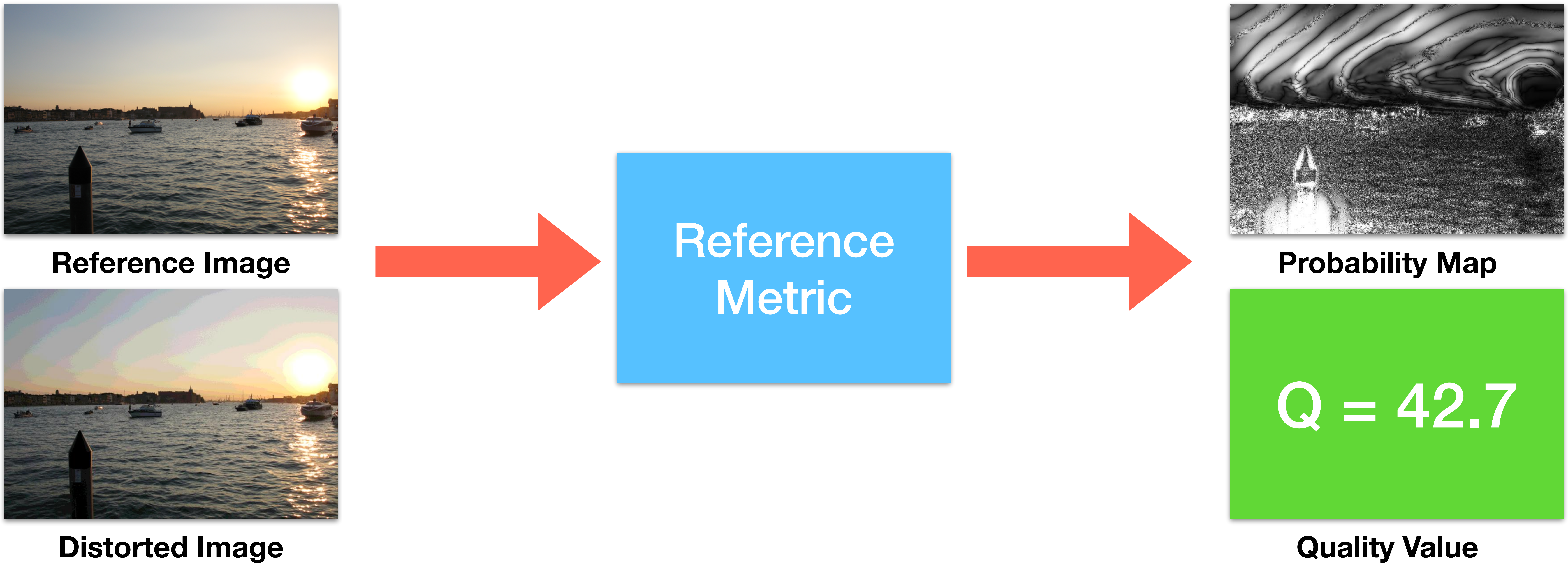
Deep HDR Metrics for Images

Francesco Banterle and Alessandro Artusi

Why Do We Need Metrics?

- In HDR/SDR Imaging, we need to determine and to understand what is happening during different steps of the pipeline:
  - **Acquisition**: we want to understand if there are artifacts due to acquisition or single image reconstruction;
  - **Compression**: we want small file size at maintaining high-quality;
  - **Tone mapping**: we want to adapt content for different display while keeping quality as it was “scene-referred”.

Reference Metrics



Reference Metrics: Current Limitations

- These models are very complex:
  - Difficult to port to GPUs with ease.
- They are computationally expensive; e.g., minutes of computations for a full HD image.
- Do we need a distortion map?
  - For most tasks we just need **a single value**!

DIQM: Deep Image Quality Metric

- A general and simple architecture meant for distilling reference-based metrics (e.g., HDR-VDP, DRIIM, etc.) into a CNN architecture.

DIQM: Datasets

	TRAINING SET	VALIDATION SET	TEST SET	TOTAL
HDR-C (HDR-VDP 2.2)	12,768	1,596	1,638	16,002
SDR-D (HDR-VDP 2.2)	11,536	1,441	1,441	14,418

DIQM: SDR-D Dataset

REFERENCE SDR IMAGE

BLUR DISTORTION

NOISE DISTORTION

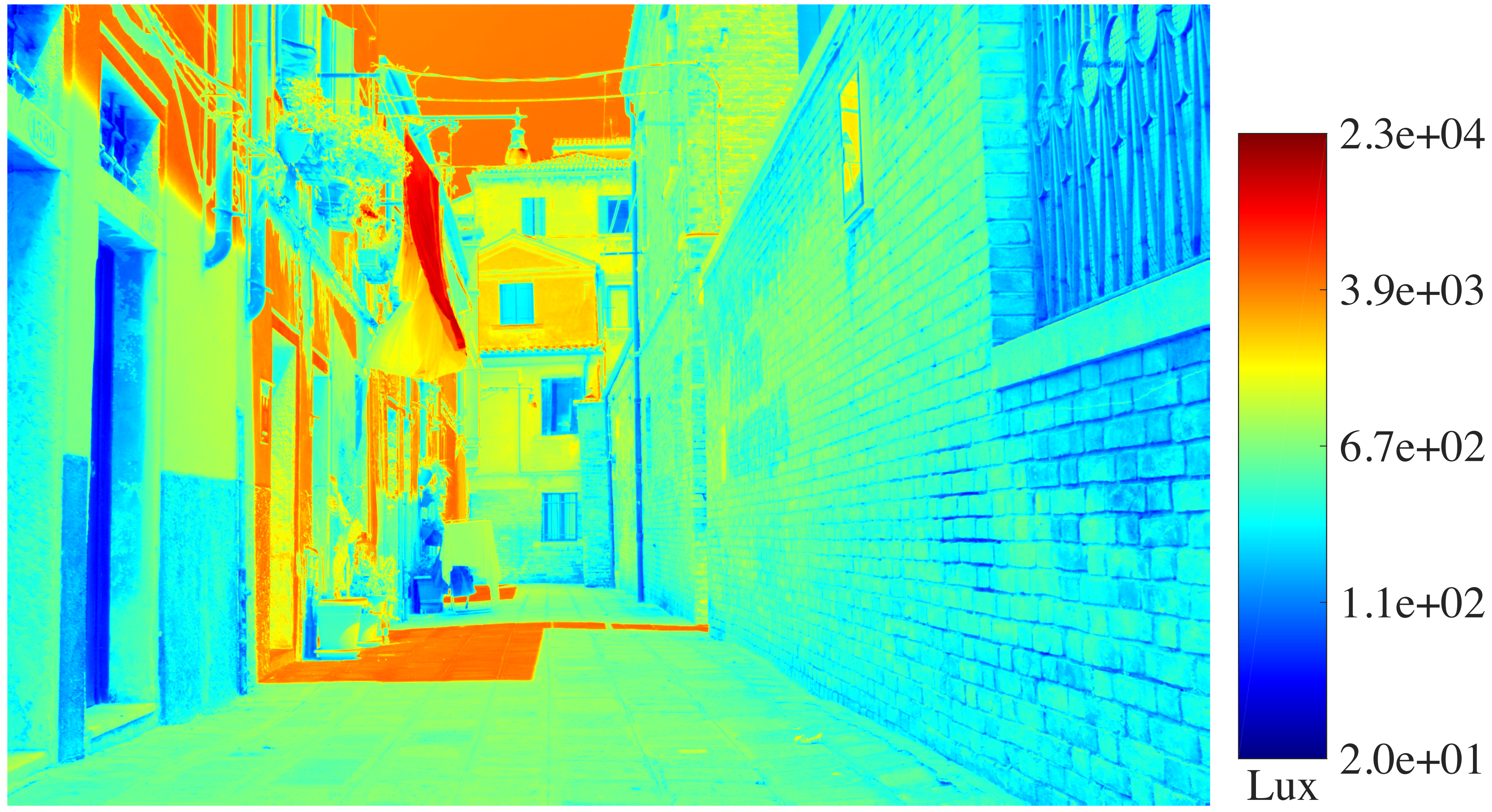
DIQM: SDR-D Dataset

REFERENCE SDR IMAGE

QUANTIZATION DISTORTION

SIN GRATE DISTORTION

DIQM: HDR-C Dataset



HDR Image

JPEG-XT:
- Random Profile
- Random Residual Compressione

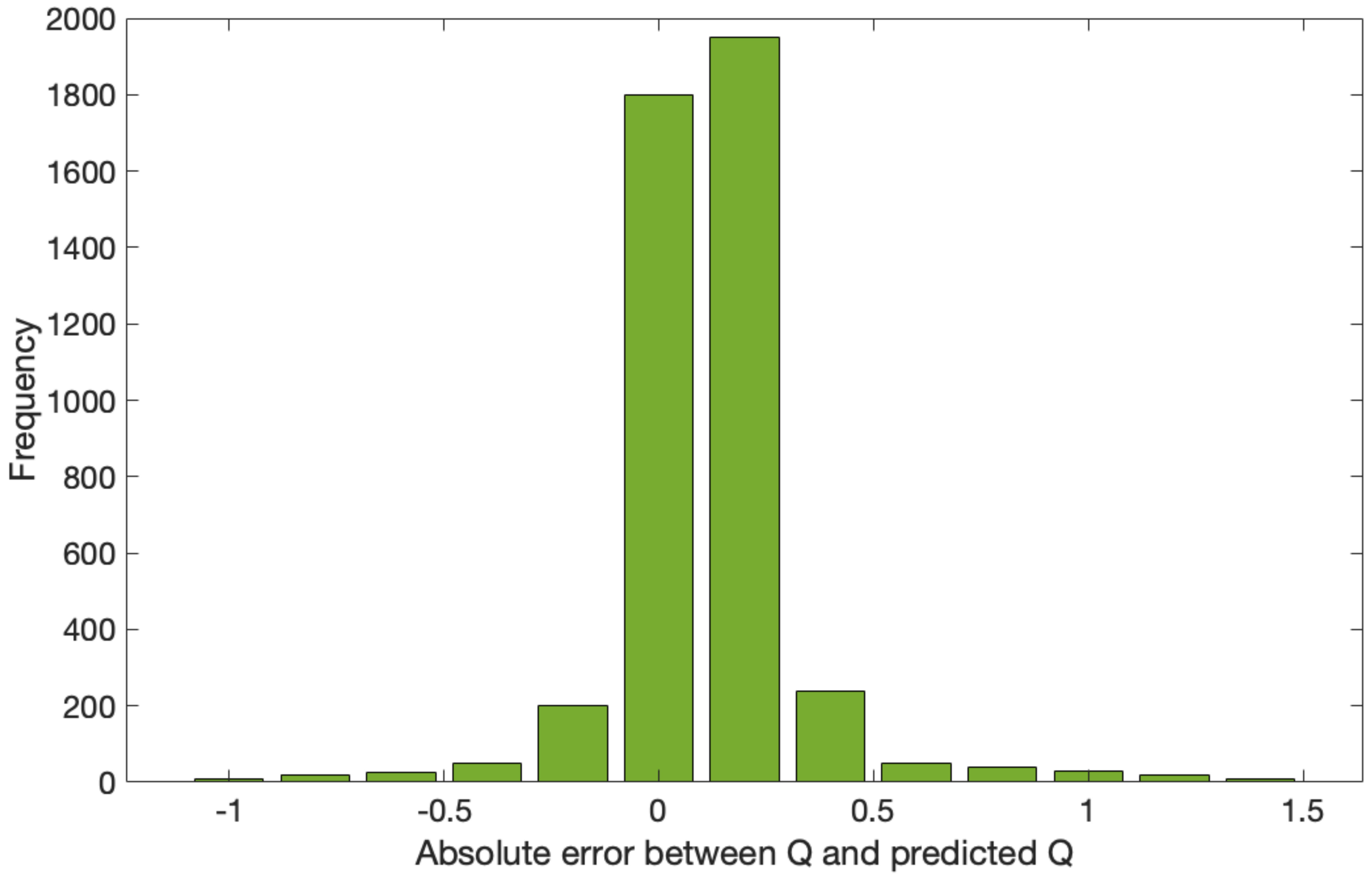
8-bit Layer

METADATA

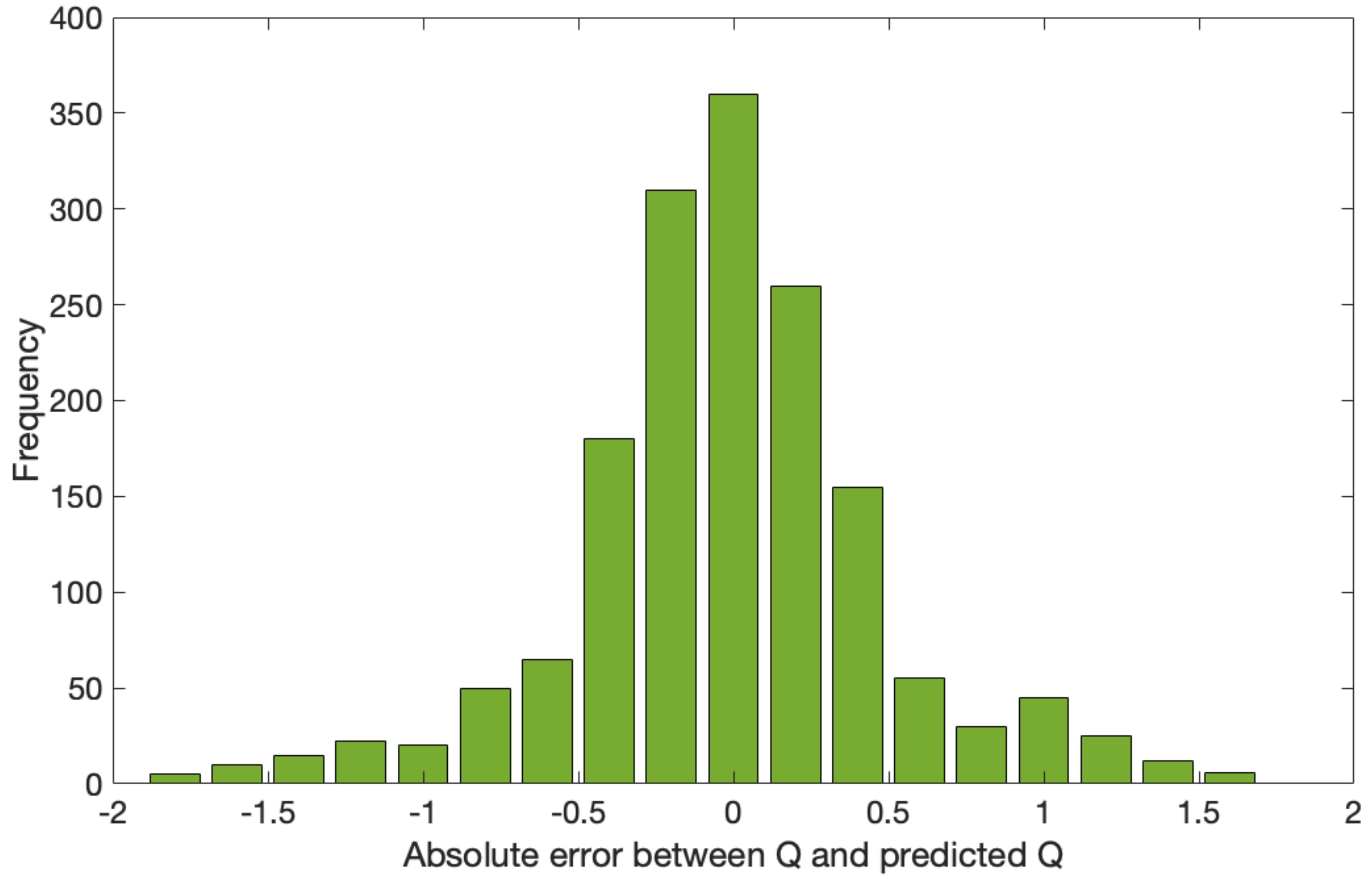
DIQM: Loss and Encoding

- Loss is a classic MSE; it works well for predicting quantitative values.
- Encoding:
  - SDR Images: linear scaling to fit the range $[0,1]$
  - HDR Images: $\log_{10}(x + 1)$

DIQM: Results Test Set

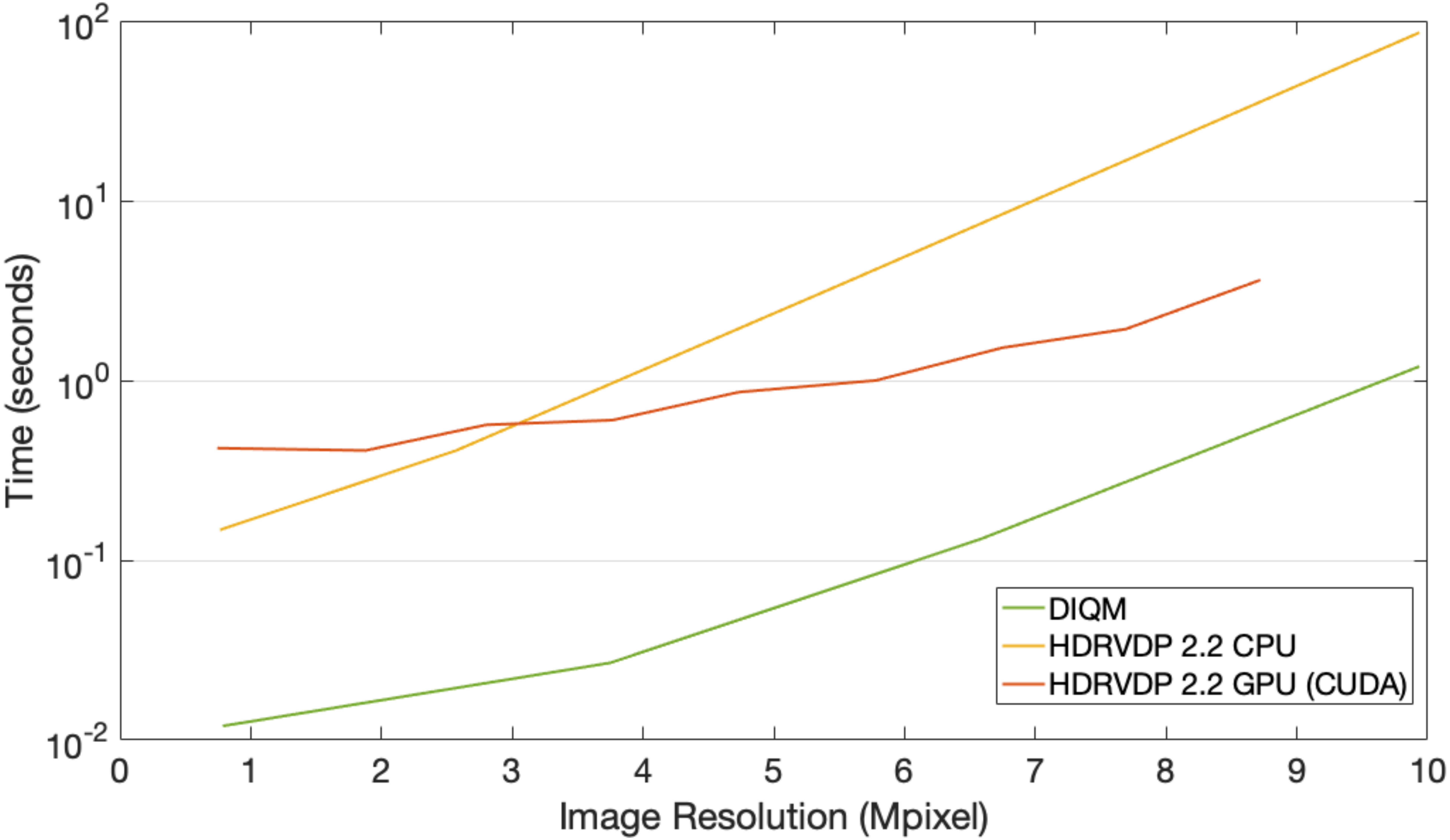


HDR-C



SDR-D

DIQM: Timings Results



DIQM: Conclusions

- There two main results:
  - We can distill metrics into a CNN with reasonable quality;
  - The CNN can be simple; no need of overly complex models:
    - The CNN runs real-time at inference time;
    - Small weights.

Visibility Distortion Maps CNN-based

- Several applications (imaging and computer graphics) are requiring a visual difference map.
  - Traditional objective metrics can not be used; e.g., single numeric value.
  - Existing visibility metrics produce a visual difference map, but they are inaccurate.
    - Lack of large image collections with good coverage of possible distortion.
    - A large dataset of image pairs (ground truth, distorted) is collected, e.g., user marking indicate wether the distortion is visible.
    - A CNN is used and trained on this large dataset.

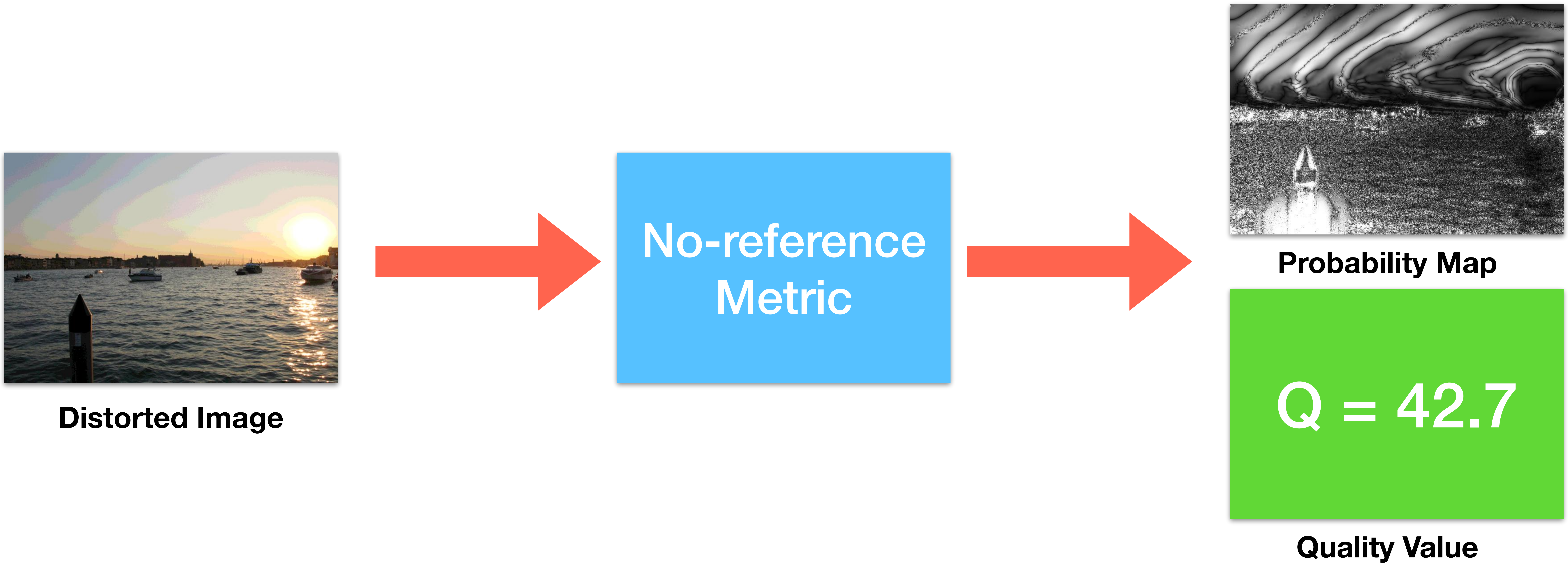
Visibility Distortion Maps CNN-based

Visibility Distortion Map: Conclusions

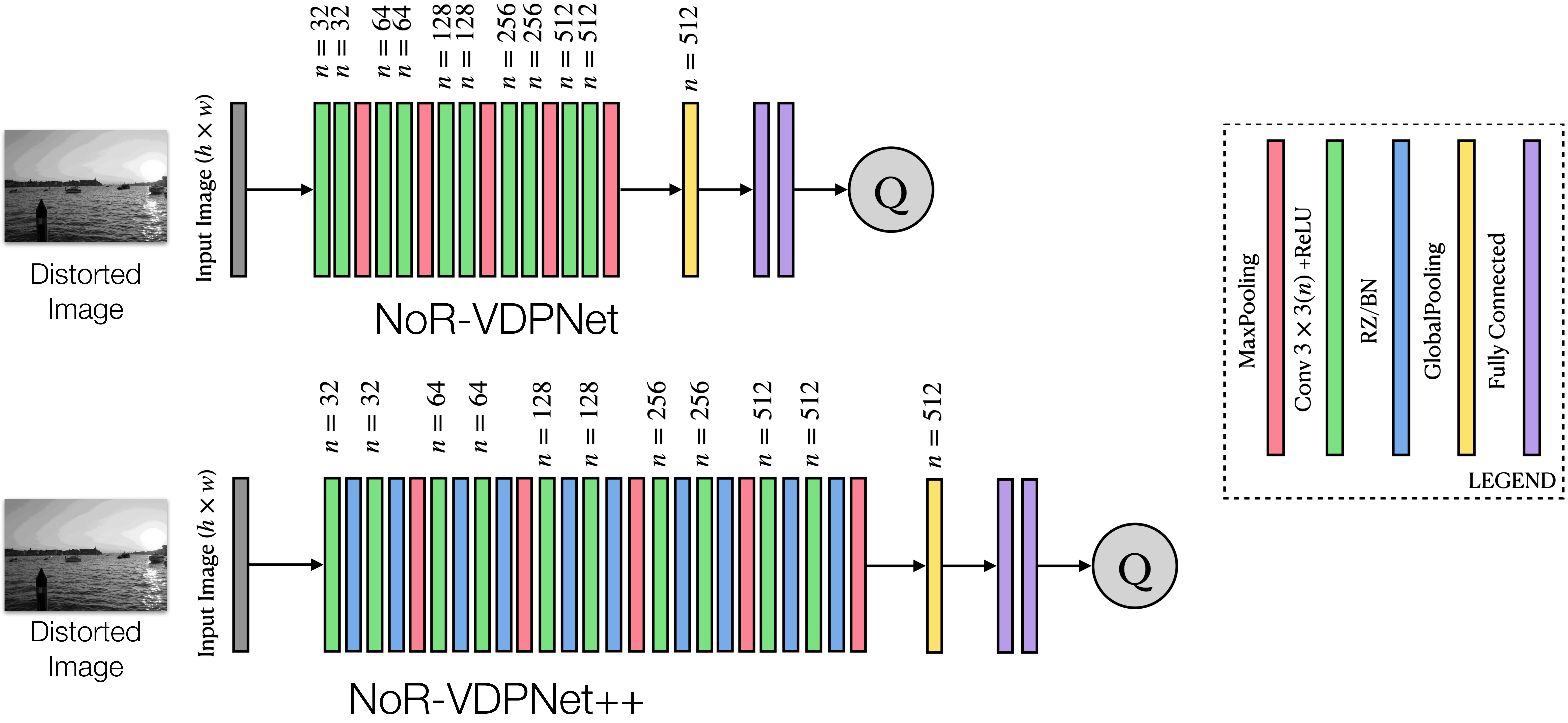
- There main results:
    - A statistical model has been proposed to fit the large data collected and used as loss function.
    - Existing visibility metrics can be improved through the usage of a CNN based method, which it is trained using the collected dataset and using as loss function the proposed statical model.

Going No-Reference

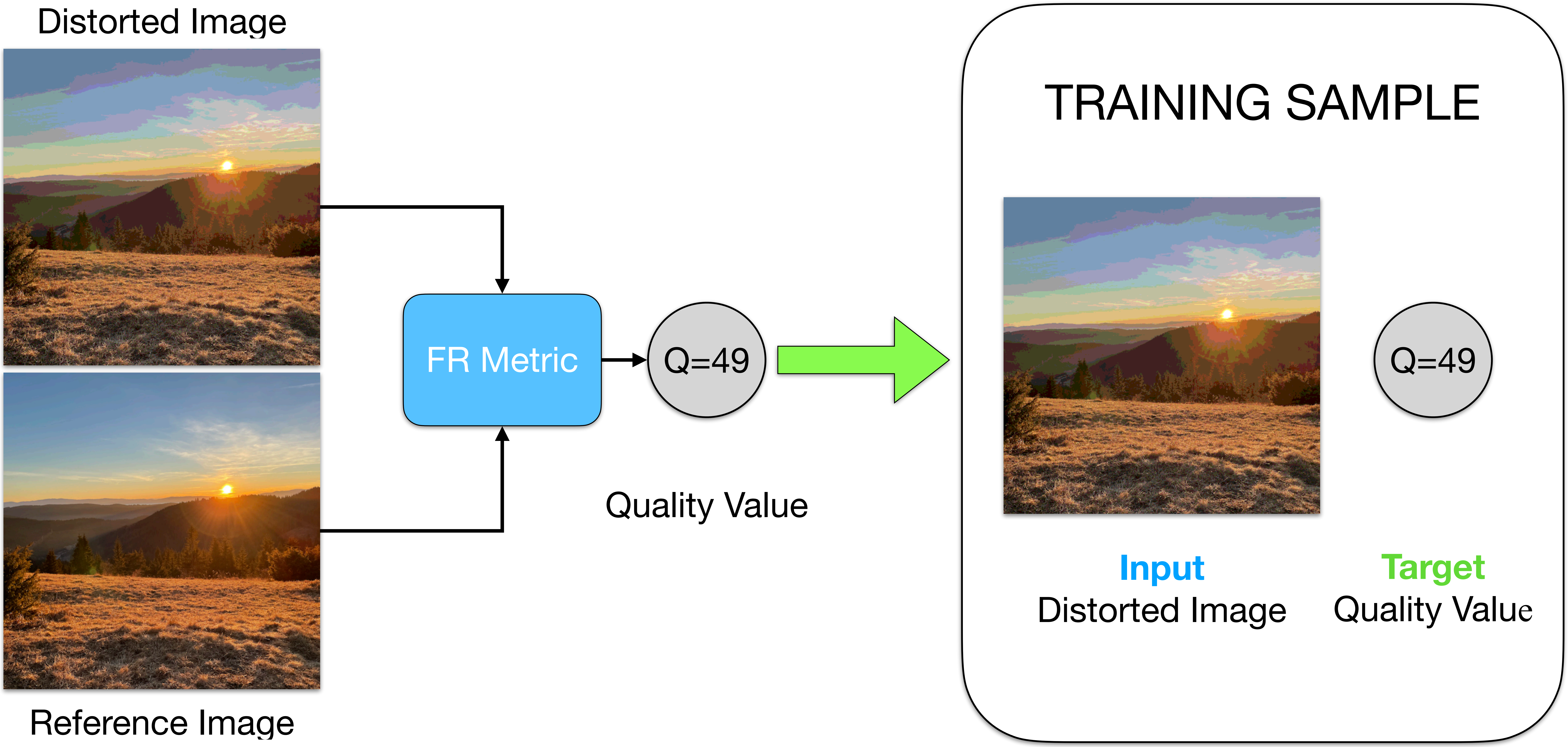
No-Reference Metrics



NoR-VDPNet(++): Architecture



Training Set

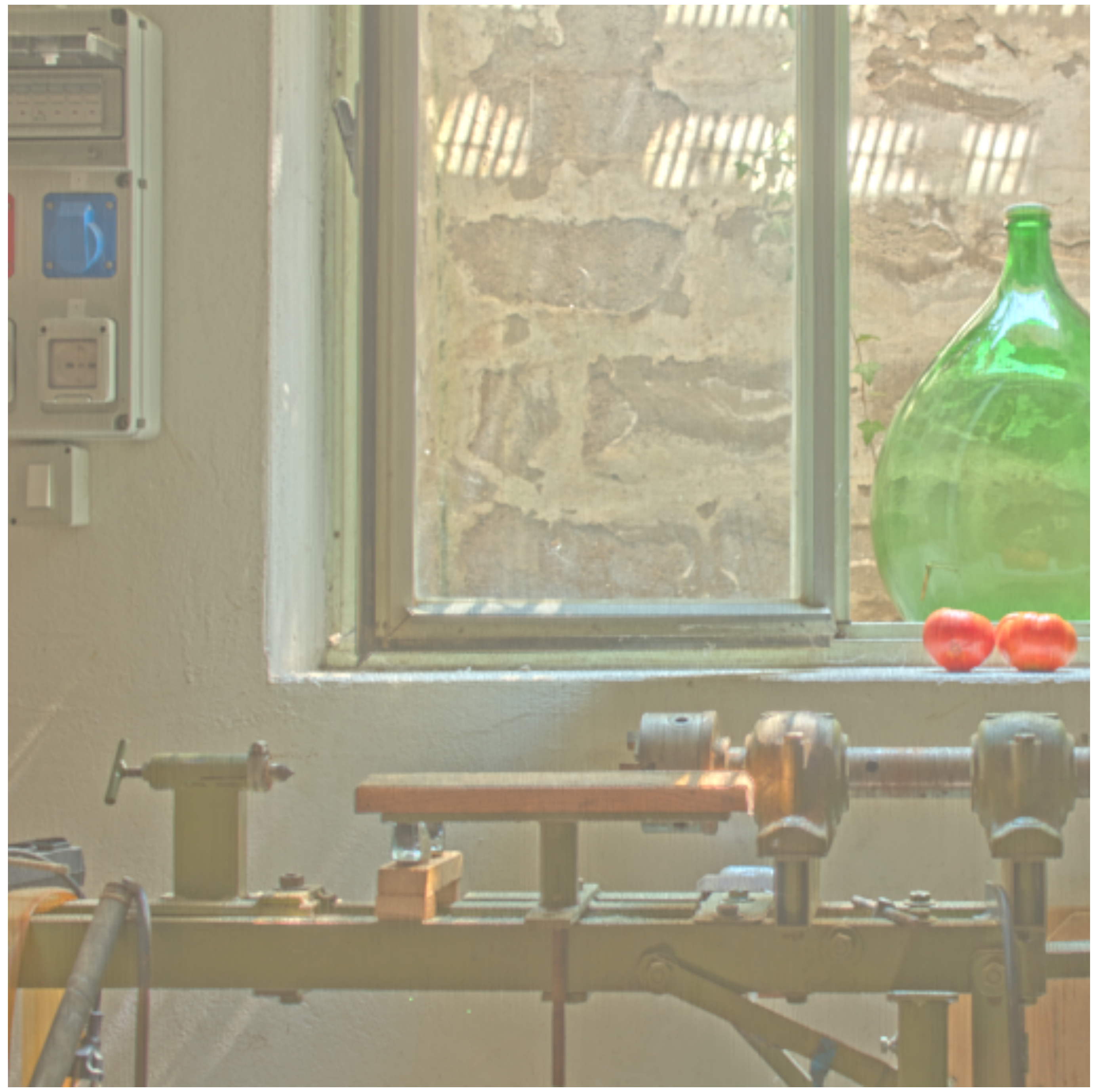


NoRVDPNet(++): HDR-VDP2.2/TMQI Datasets

	TRAINING SET	VALIDATION SET	TEST SET	TOTAL
HDR-C (HDR-VDP2.2)	49.602	6.216	6.216	62.034
SDR-D (HDR-VDP2.2)	80.244	10.025	10.044	100.313
TMO (TMQI)	106.290	13.320	13.320	132.930
ITMO (HDR-VDP2.2)	106.290	13.320	13.320	132.930

NoRVDPNet(++): TMO Dataset

Drago et al. 2003



Durand and Dorsey 2002

Reinhard et al. 2002

18 tone mapping operators from the HDR-Toolbox: https://github.com/banterle/HDR_Toolbox/

NoRVDPNet(++): ITMO Dataset

Input SDR Image

Eilertsen et al. 2017
(tonemapped)

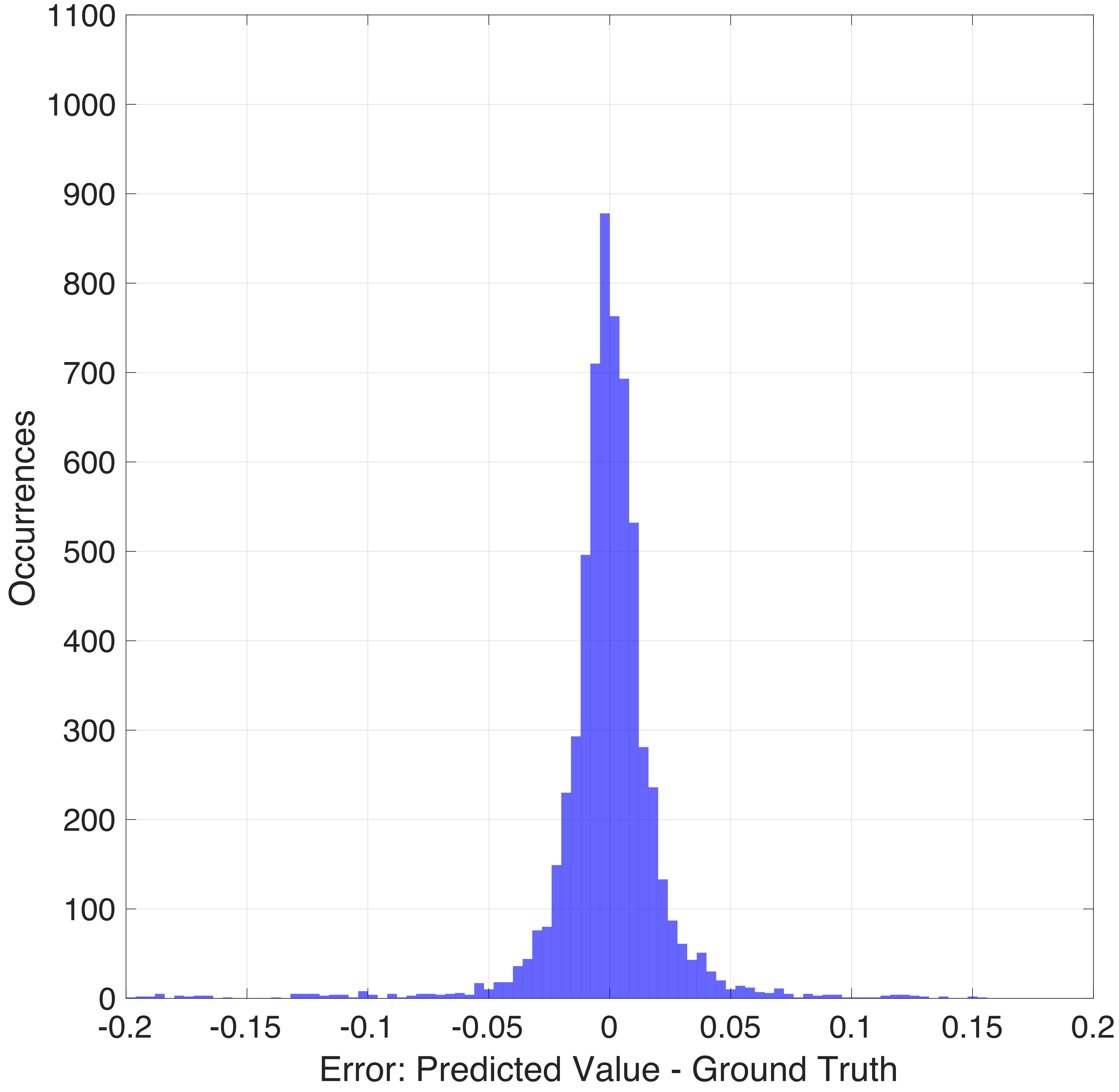
Santos et al. 20202
(tonemapped)

6 inverse tone mapping operators 4 available in the HDR-Toolbox: https://github.com/banterle/HDR_Toolbox/

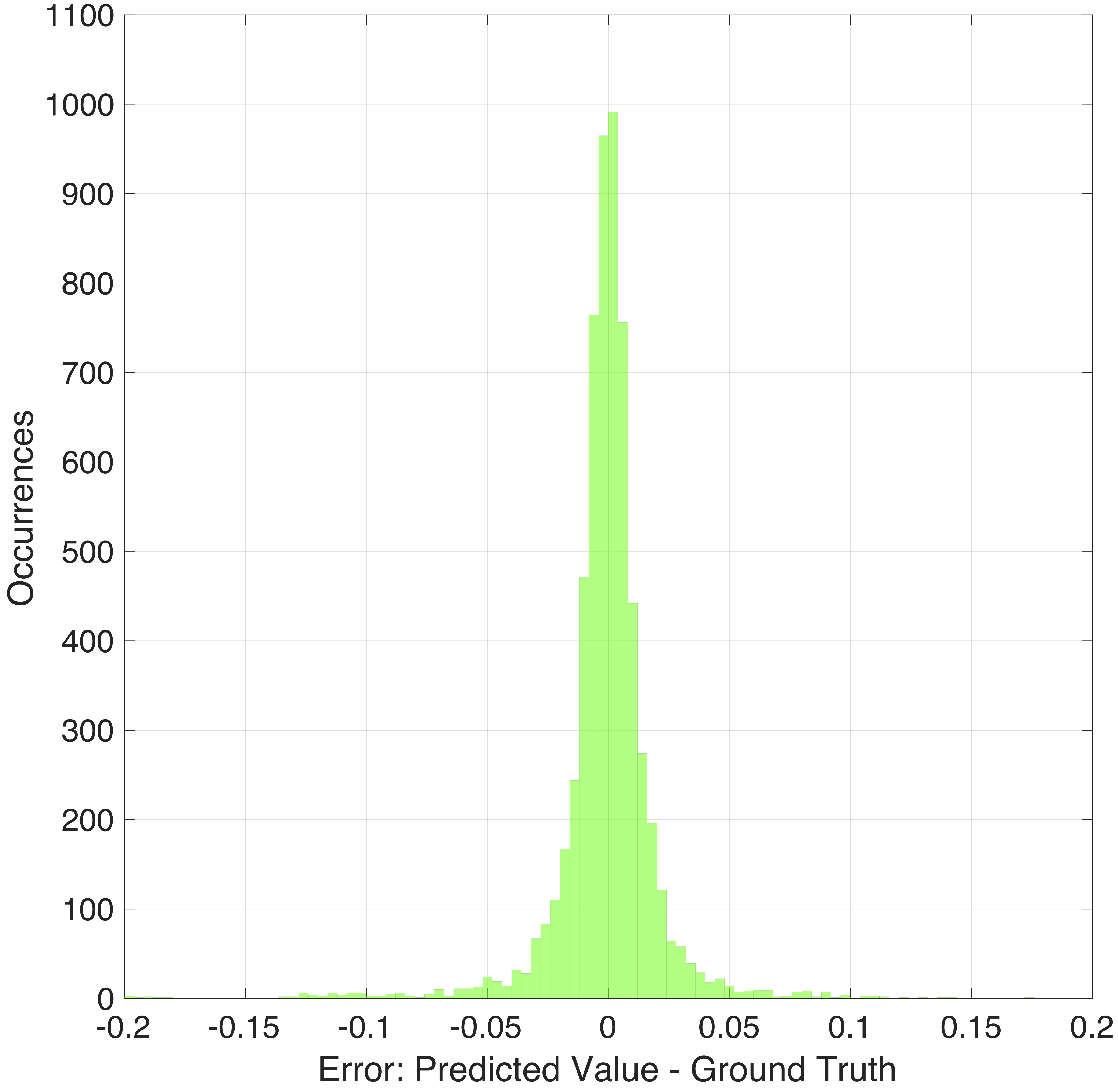
NoR-VDPNet(++): Loss and Encoding

- Loss is a classic MSE; it works well for predicting quantitative values:
- Encoding:
  - SDR Images: linear scaling to fit the range $[0,1]$
  - HDR Images: $\log_{10}(x + 1)$

Results: HDR-C Test Set

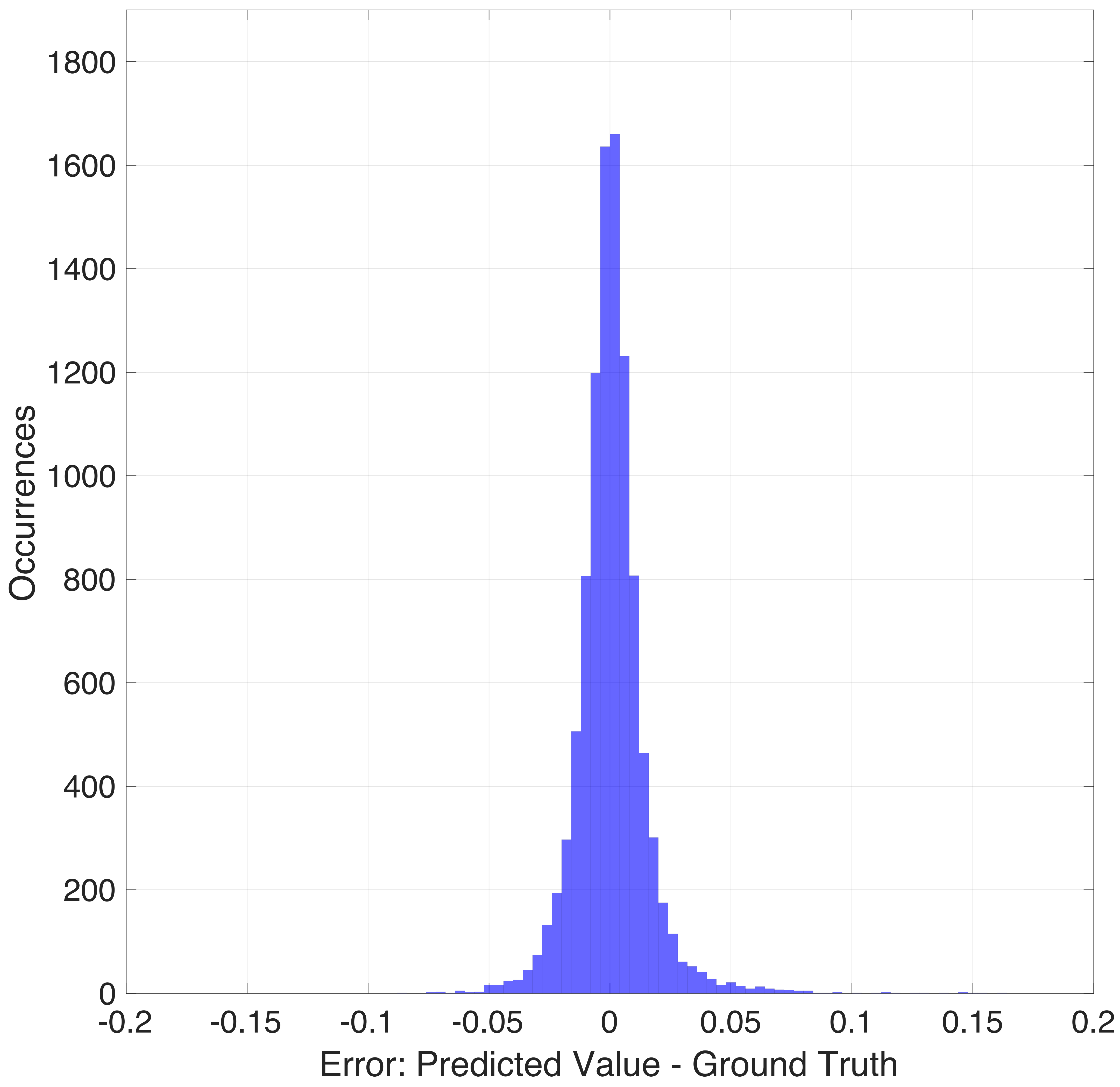


NoRVDPNet

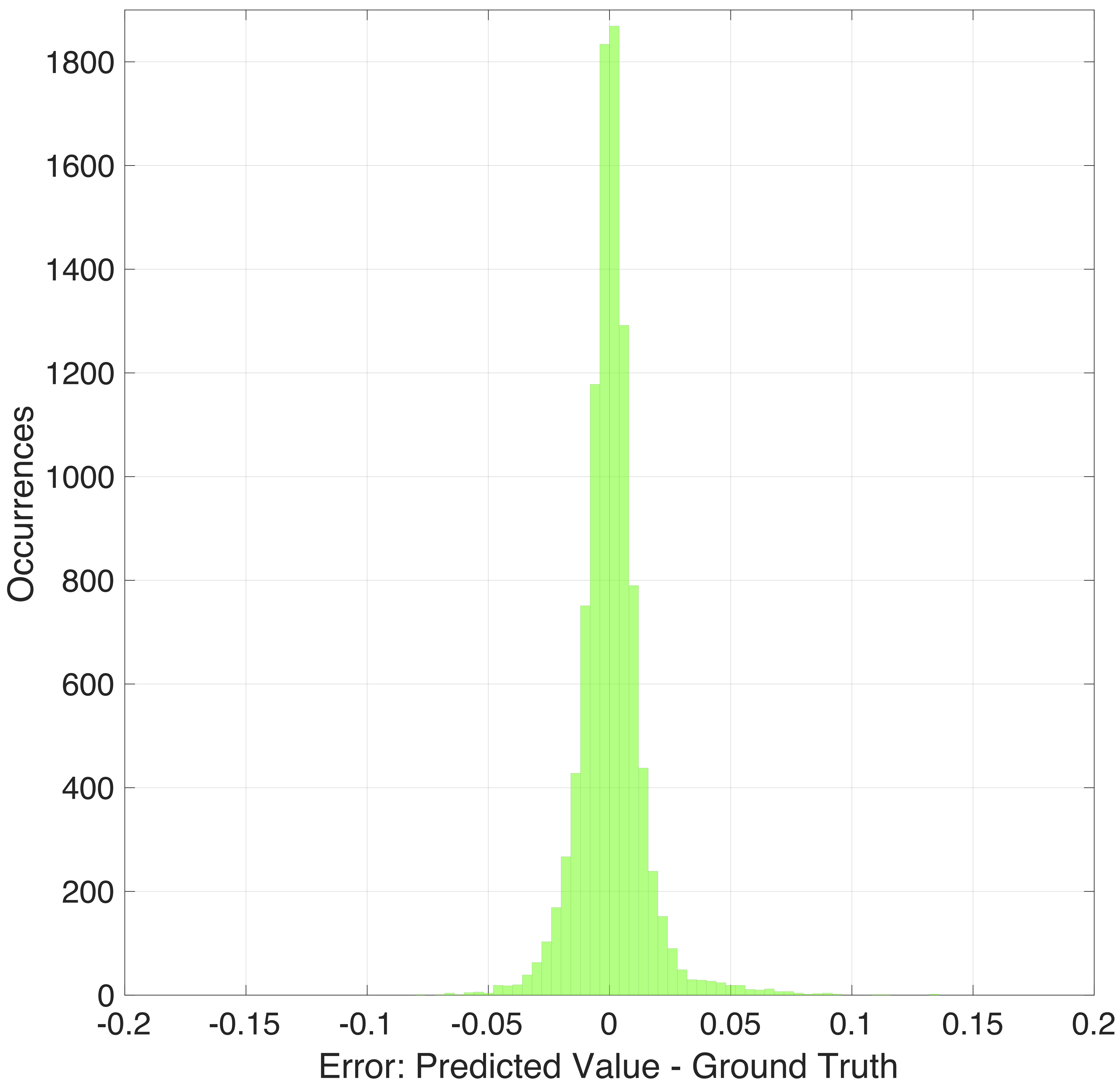


NoRVDPNet++

Results: SDR-D Test Set

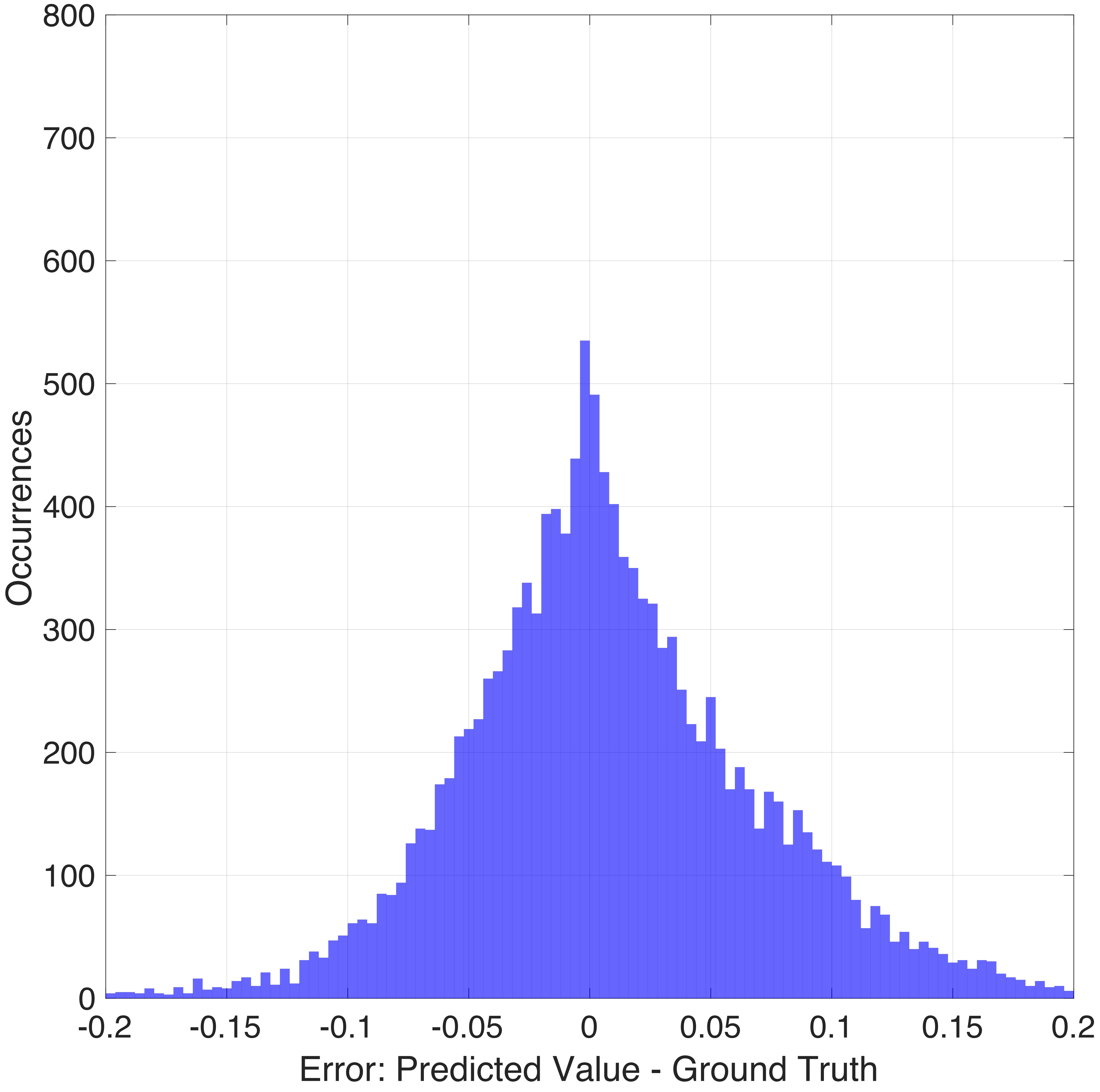


NoRVDPNet

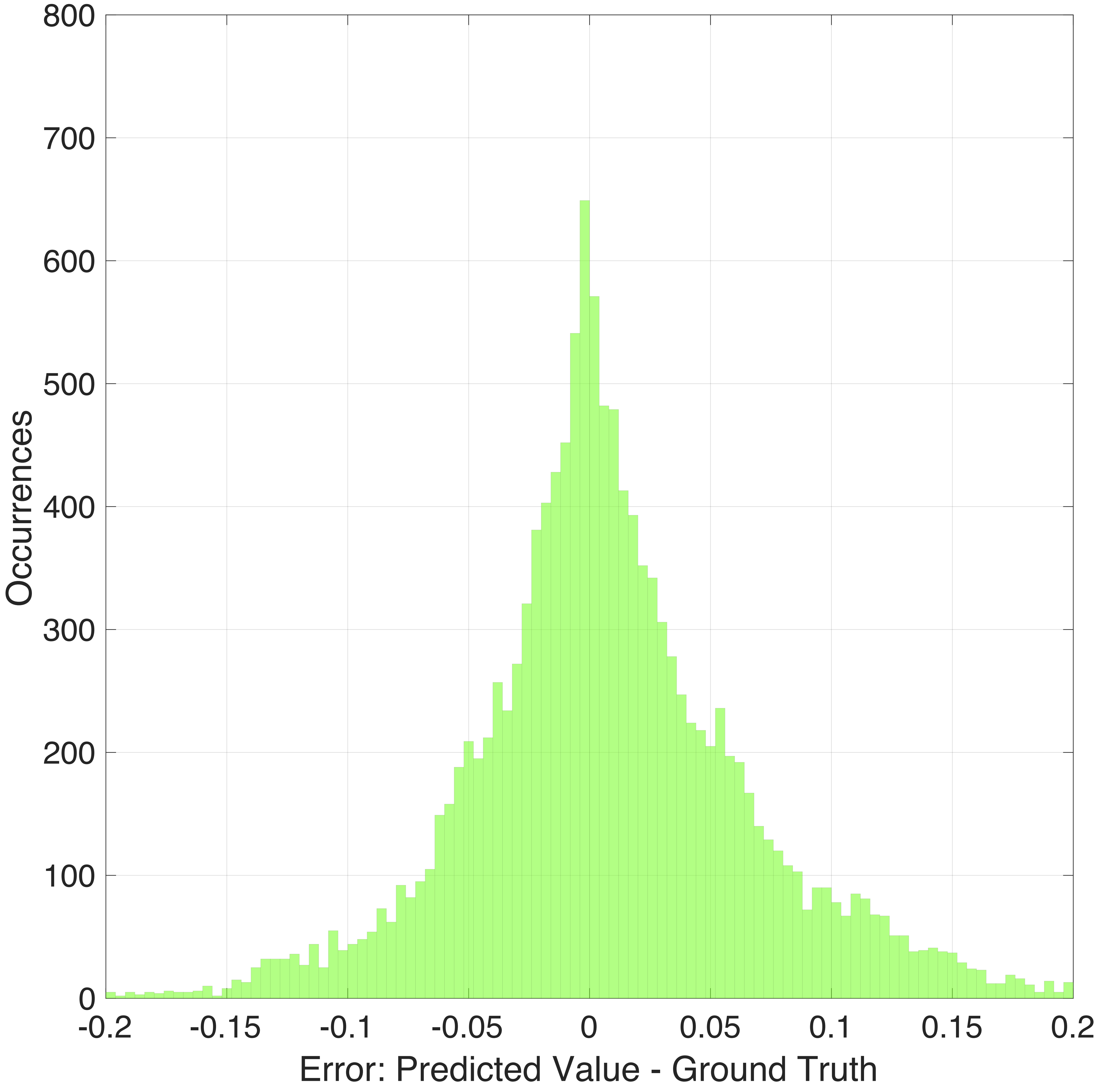


NoRVDPNet++

Results: ITMOS Test Set

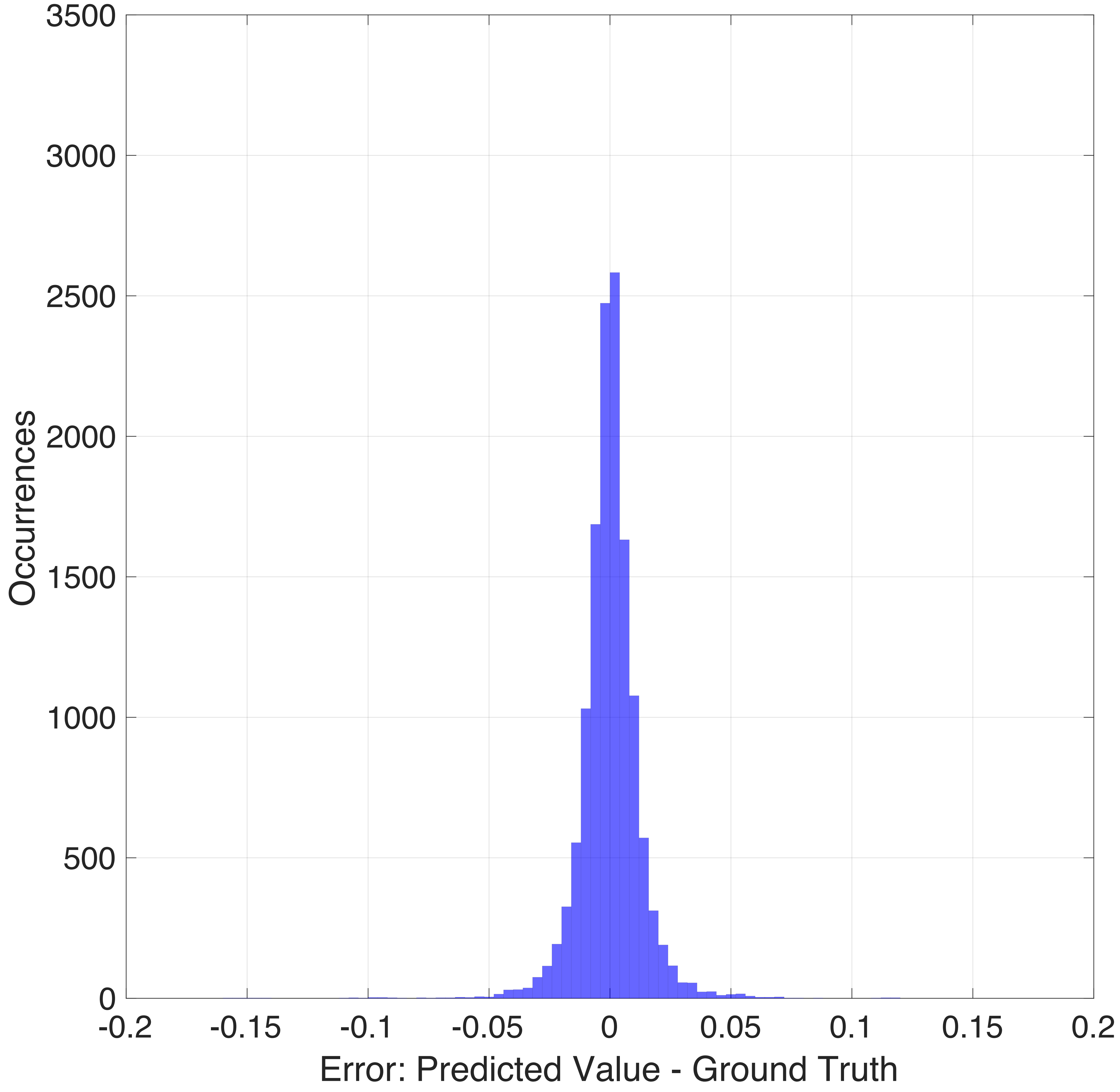


NoRVDPNet

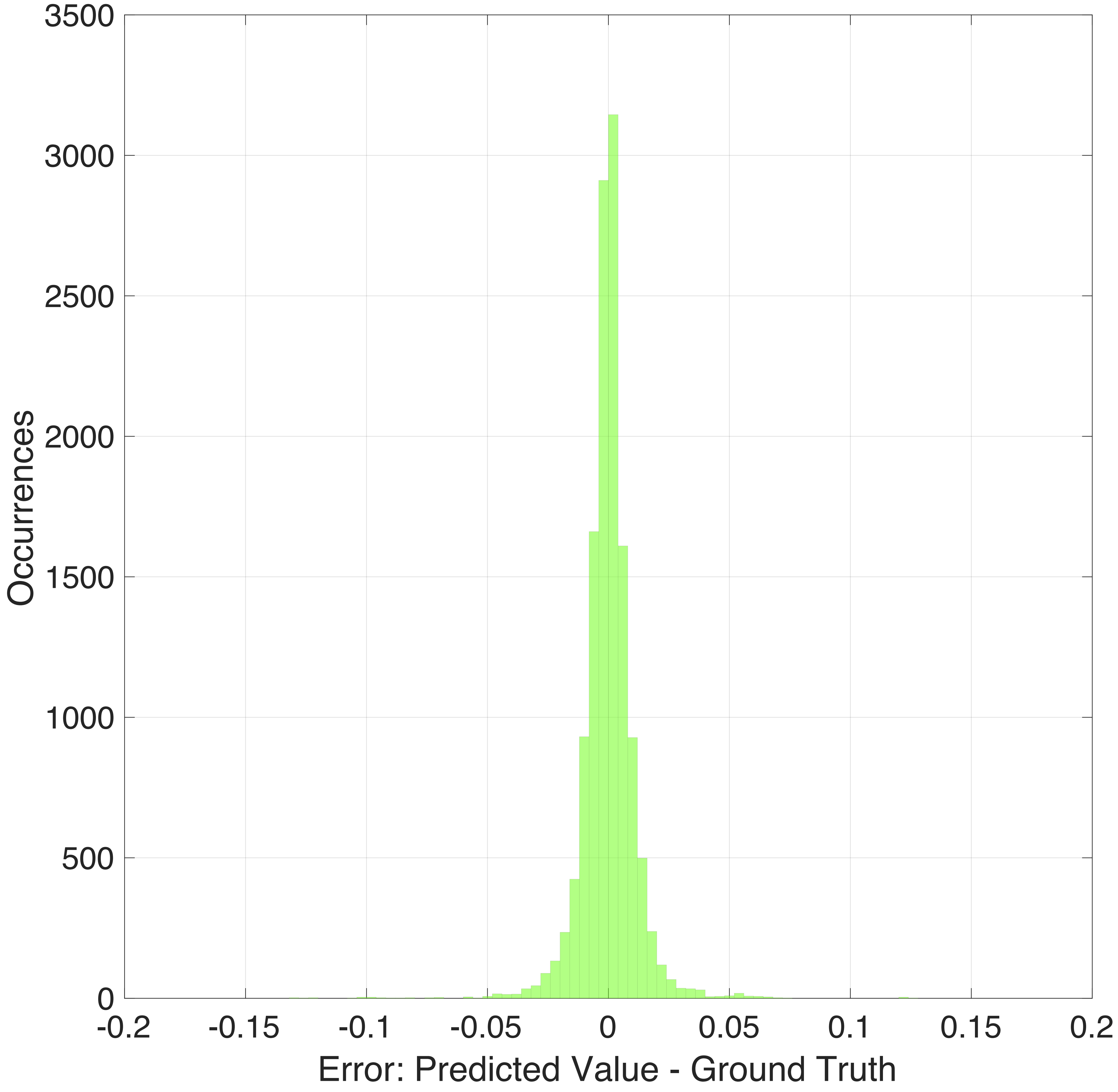


NoRVDPNet++

Results: TMOS Test Set

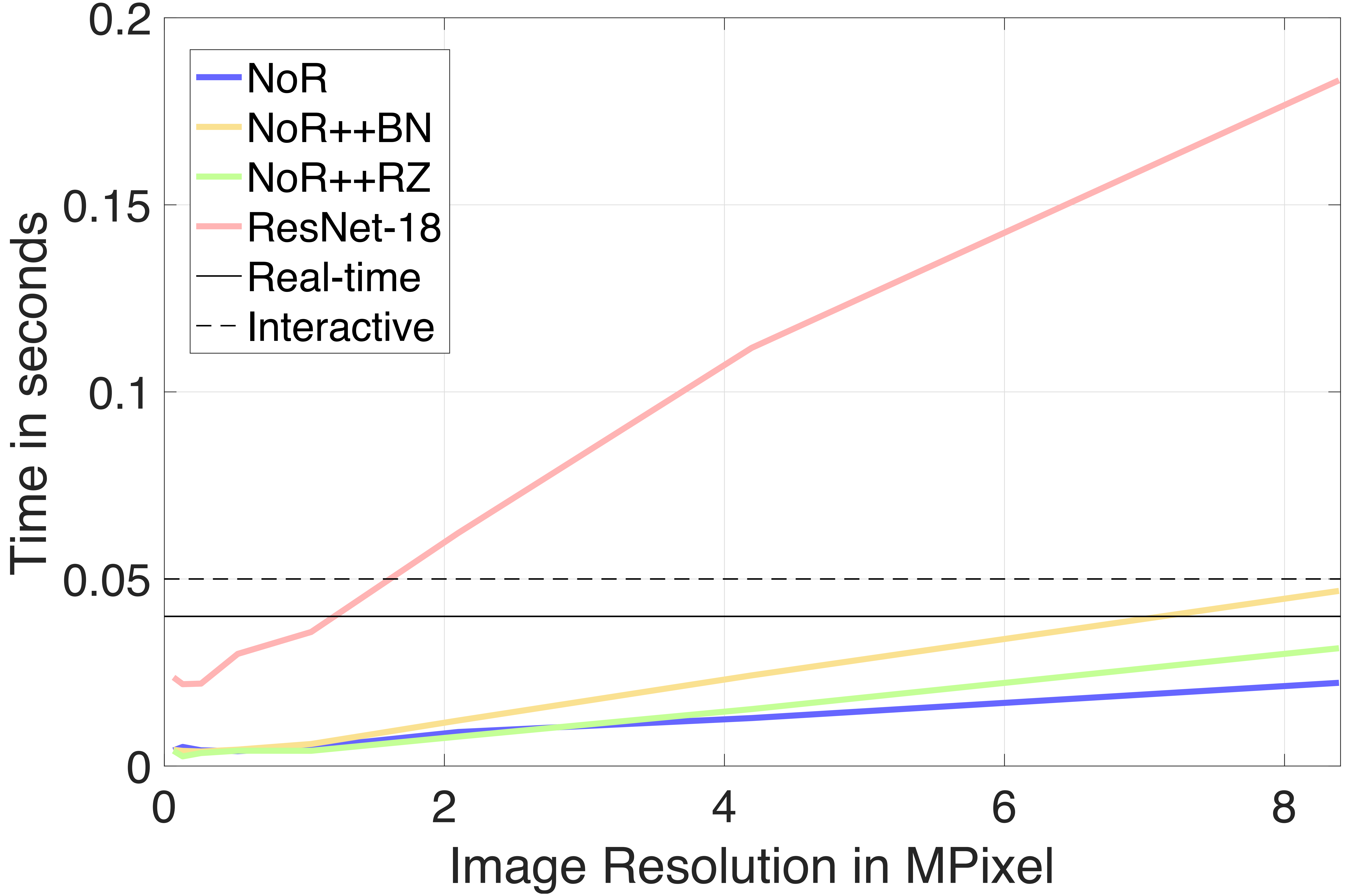


NoRVDPNet



NoRVDPNet++

Timings



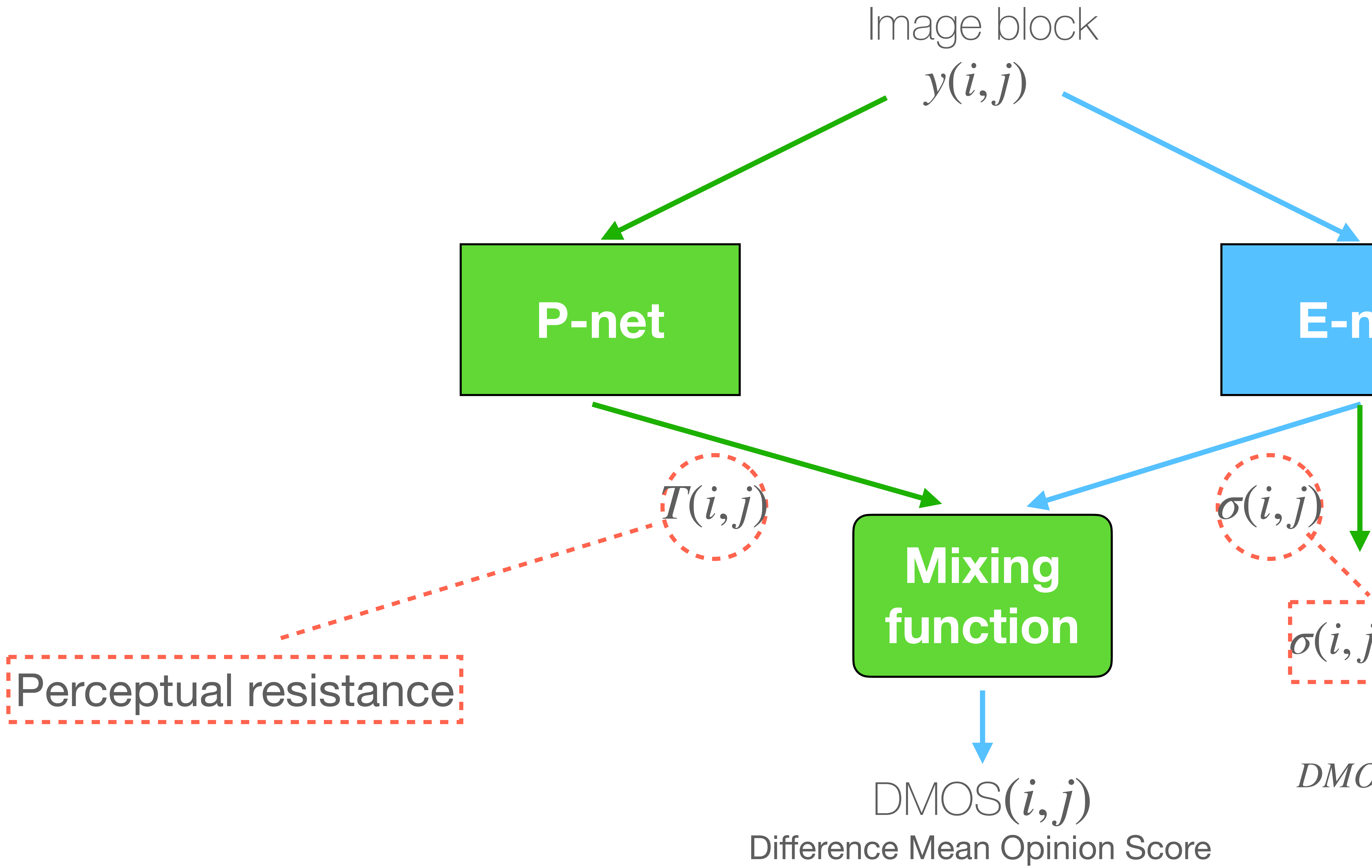
NoR
NoR++BN
NoR++RZ
ResNet-18
Real-time
Interactive
Time in seconds
0.2
0.15
0.1
0.05
0
0
2
4
6
8
Image Resolution in MPixel

NoR-VDPNet(++): Conclusions

- We can go from reference to no-reference;
- When we model several distortions we have a larger error than a single distortion;
- Layer normalization increases quality;
- This scheme works for TMQI (SSIM-based);
- Still real-time performance.

NR-IQA

NR-IQA Principle

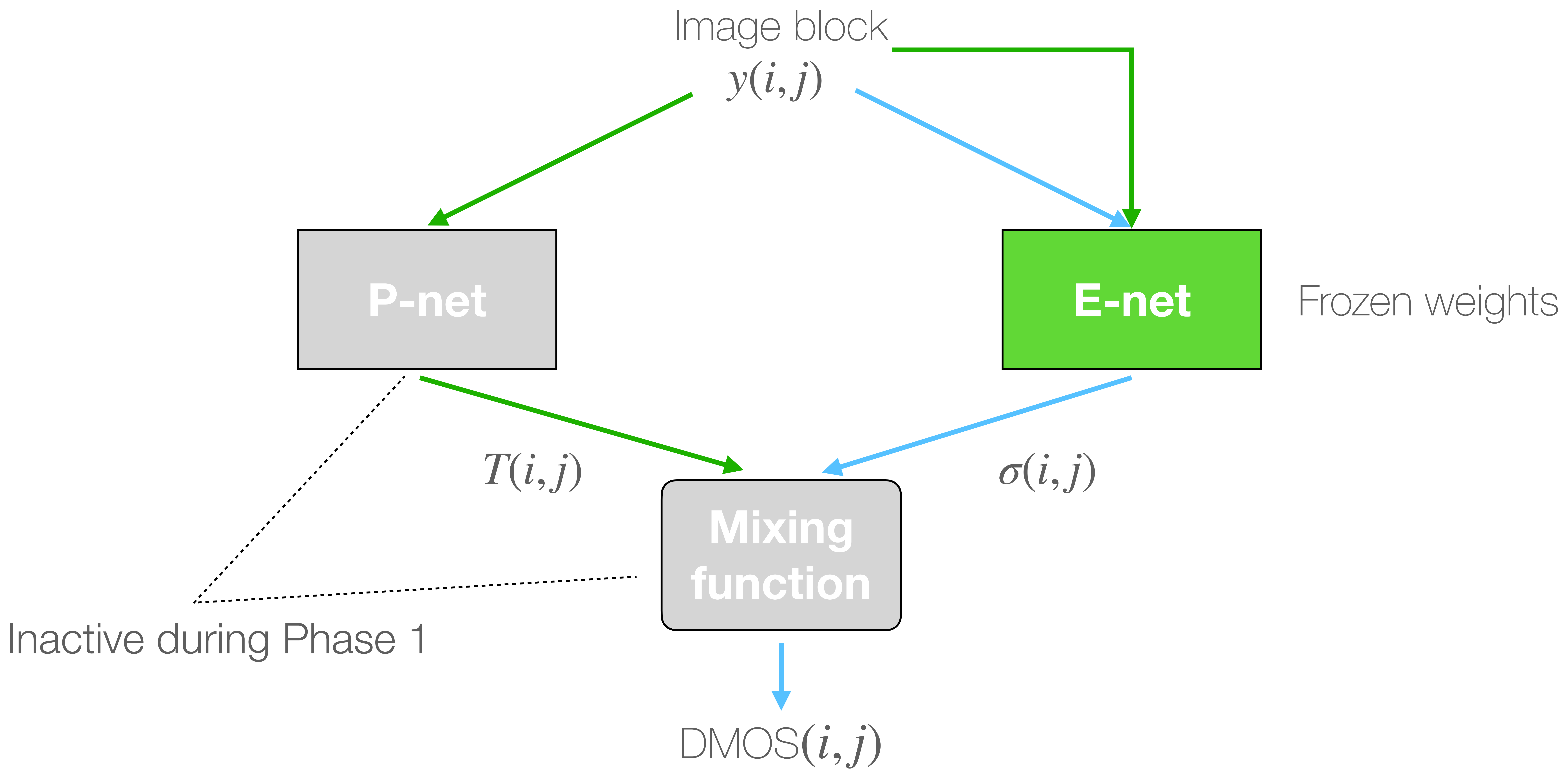


- Perceptual threshold
  - Not from psychophysical experiments
  - Data driven method

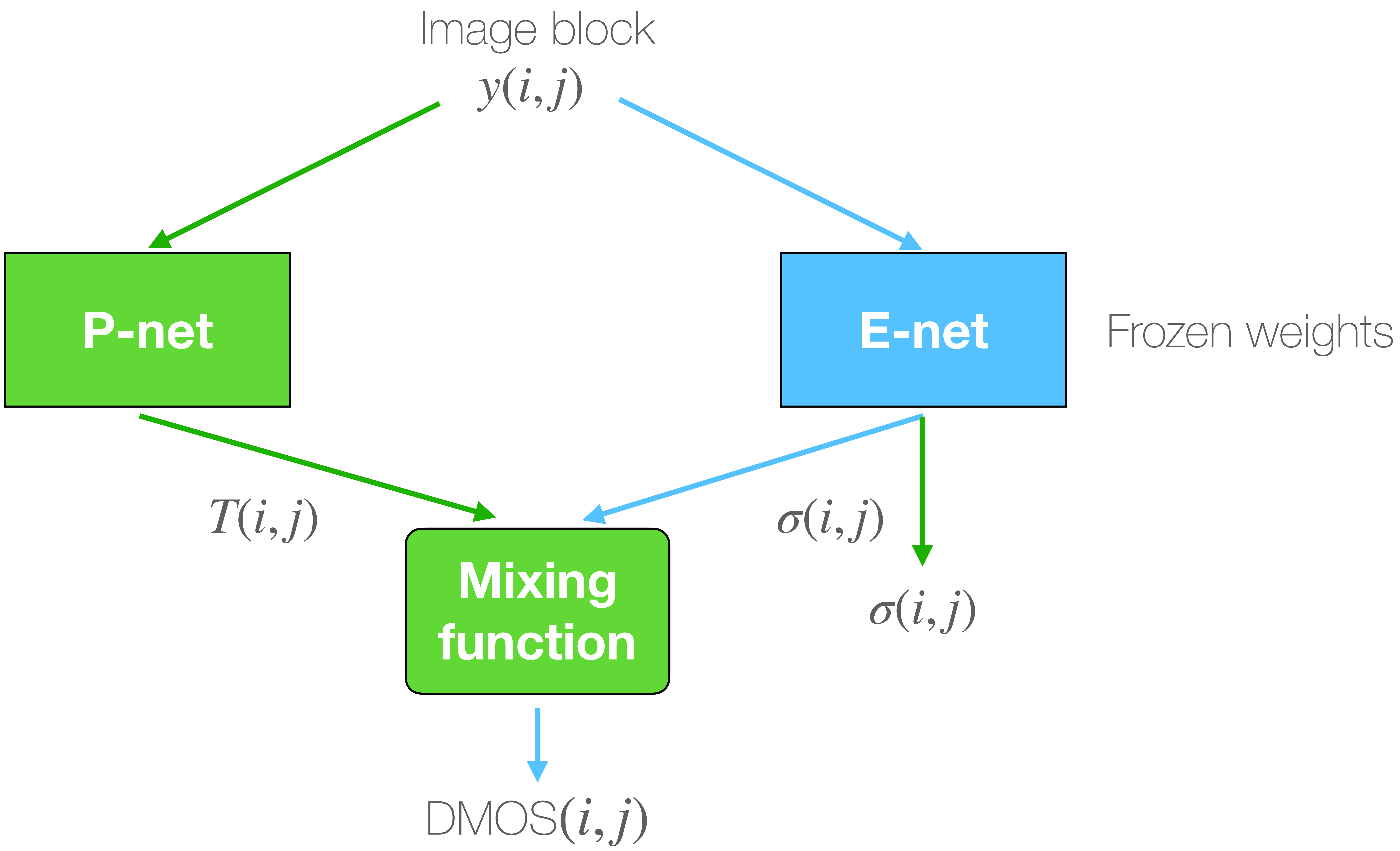
$$\sigma(i,j) = \text{mean}\left(\left|Y - Y_{\text{distorted}}\right|\right)$$

$$DMOS(i,j) = 1 - exp\left(-\left|\frac{k \times \tilde{\sigma}(i,j)}{T(i,j)}\right|\right)$$

NR-IQA Training - Phase 1



NR-IQA Training - Phase 2

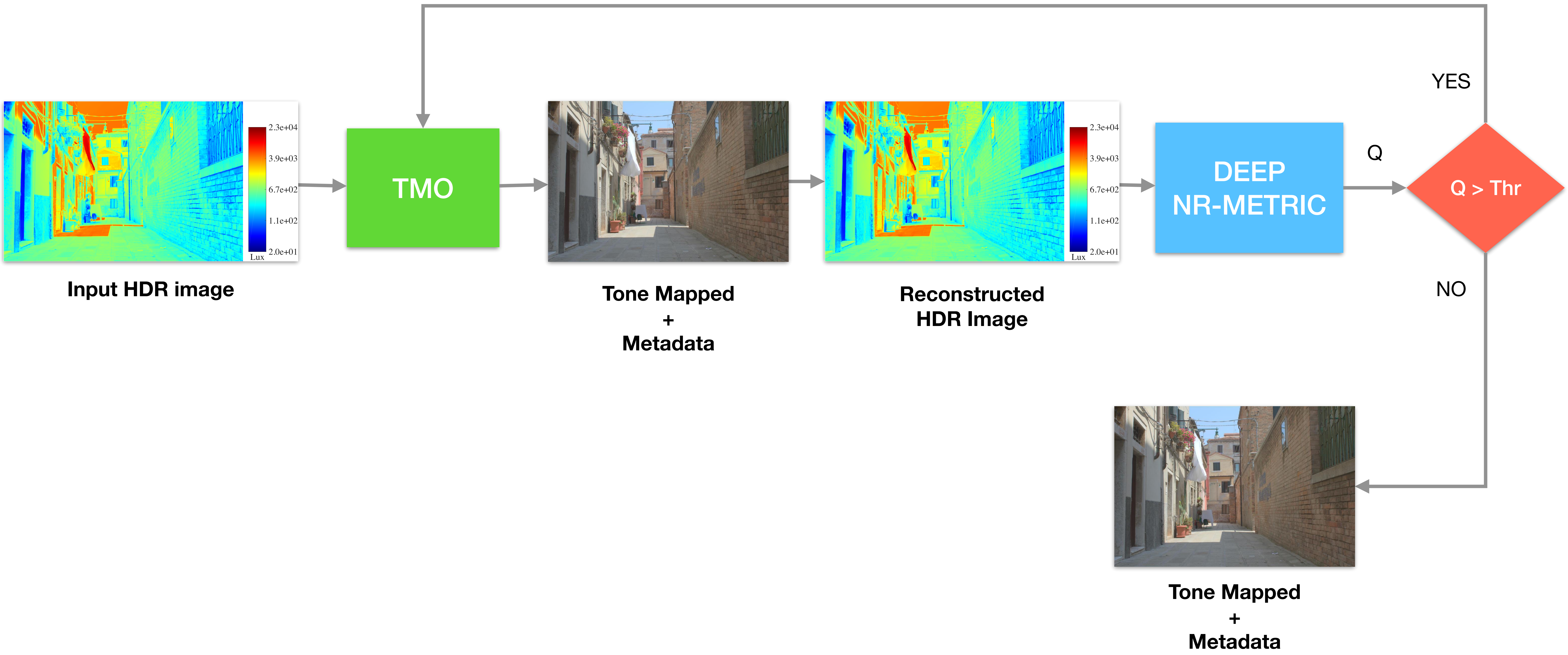


NR-IQA: Conclusions

- Main results:
  - Computational performances are not real-time, but it can be still optimized.
  - It outperforms other NR-IQA methods.
  - It is comparable to HDR FR-IQA:
    - without the need of a reference image.

Applications

Applications: Optimization Tasks



Applications: Optimized TMO

TMO without optimized parameters

TMO with optimized parameters

Video Courtesy of Jan Fröhlich - Stuttgart HDR Video Dataset

Application: A Differentiable TMO

$$L_d = \frac{L_w \alpha}{L_w \alpha + \mu} \qquad C_d = \left(\frac{C_w}{L_w} \right)^{\gamma} L_d$$

Application: A Differentiable TMO

$$L_d = \frac{L_w \boxed{\alpha}}{L_w \boxed{\alpha} + \boxed{\mu}} \quad C_d = \left(\frac{C_w}{L_w}\right)^{\boxed{\gamma}} L_d$$

Application: A Differentiable TMO

(a) $\hat{Q} = 0.903\ /Q = 0.885$

(b) $\hat{Q} = 0.906/Q = 0.930$

(c) $\hat{Q} = 0.933\ /Q = 0.914$

(d) $\hat{Q} = 0.918/Q = 0.903$

(e) $\hat{Q} = 0.902/Q = 0.889$

(f) $\hat{Q} = 0.841/Q = 0.771$

(g) $\hat{Q} = 0.951/Q = 0.831$

(h) $\hat{Q} = 0.875/Q = 0.909$

(i) $\hat{Q} = 0.951/Q = 0.967$

(j) $\hat{Q} = 0.958/Q = 0.974$

(k) $\hat{Q} = 0.967/Q = 0.976$

(l) $\hat{Q} = 0.997/Q = 0.979$

Applications: JPEG-XT Compression

Input HDR image

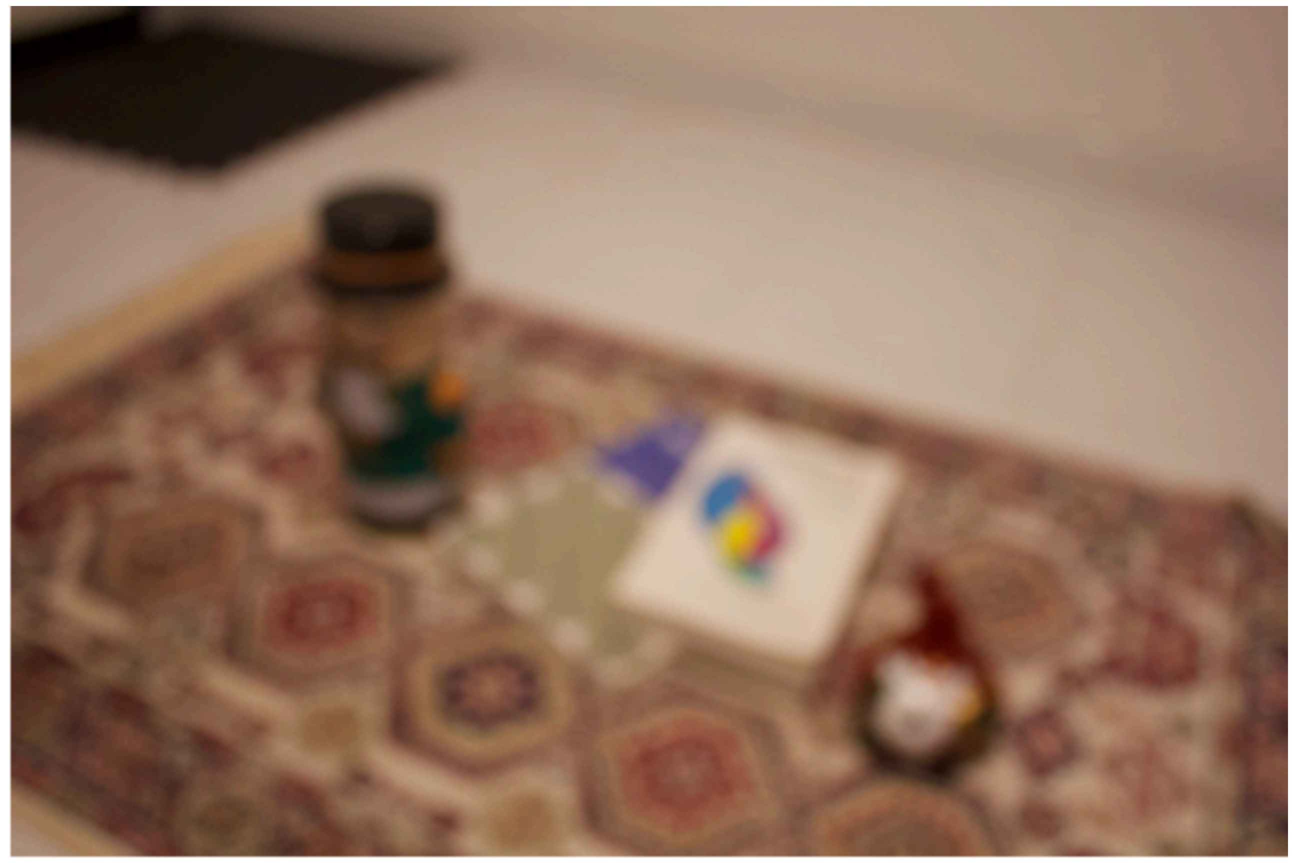
Reinhard et al.'s TMO
optimized with NoRVDPNet

Tone Mapped HDR image
for JPEG-XT

Applications: Photo Selection

Q=86.99

Q=86.92



Q=56.46

Q=91.39

Q=76.26

Q=59.9

Applications: Photo Selection

Q=86.99

Q=86.92

Q=56.46

Q=91.39

Q=76.26

Q=59.9

Future Directions

Future Directions

- Novel datasets have been published for HDR videos with MOS:
  - https://live.ece.utexas.edu/research/LIVEHDR/LIVEHDR_index.html
  - HDR videos/NeRFs metrics seem a natural next step.
- HDR Metrics based on deep-learning have only now started to appear.
- We still need to rely on experiments for capturing large datasets.

Thank you for your attention!

Please contact us at:

a.artusi@cyens.org.cy francesco.banterle@isti.cnr.it

or visit us:

https://deepacamera.org.cy http://vcg.isti.cnr.it

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 739578 and No. 820434, and from the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

This project is co-financed by the European Regional Development Fund and the Republic of Cyprus through the Research and Innovation Foundation (POST-DOC/0916/0034).