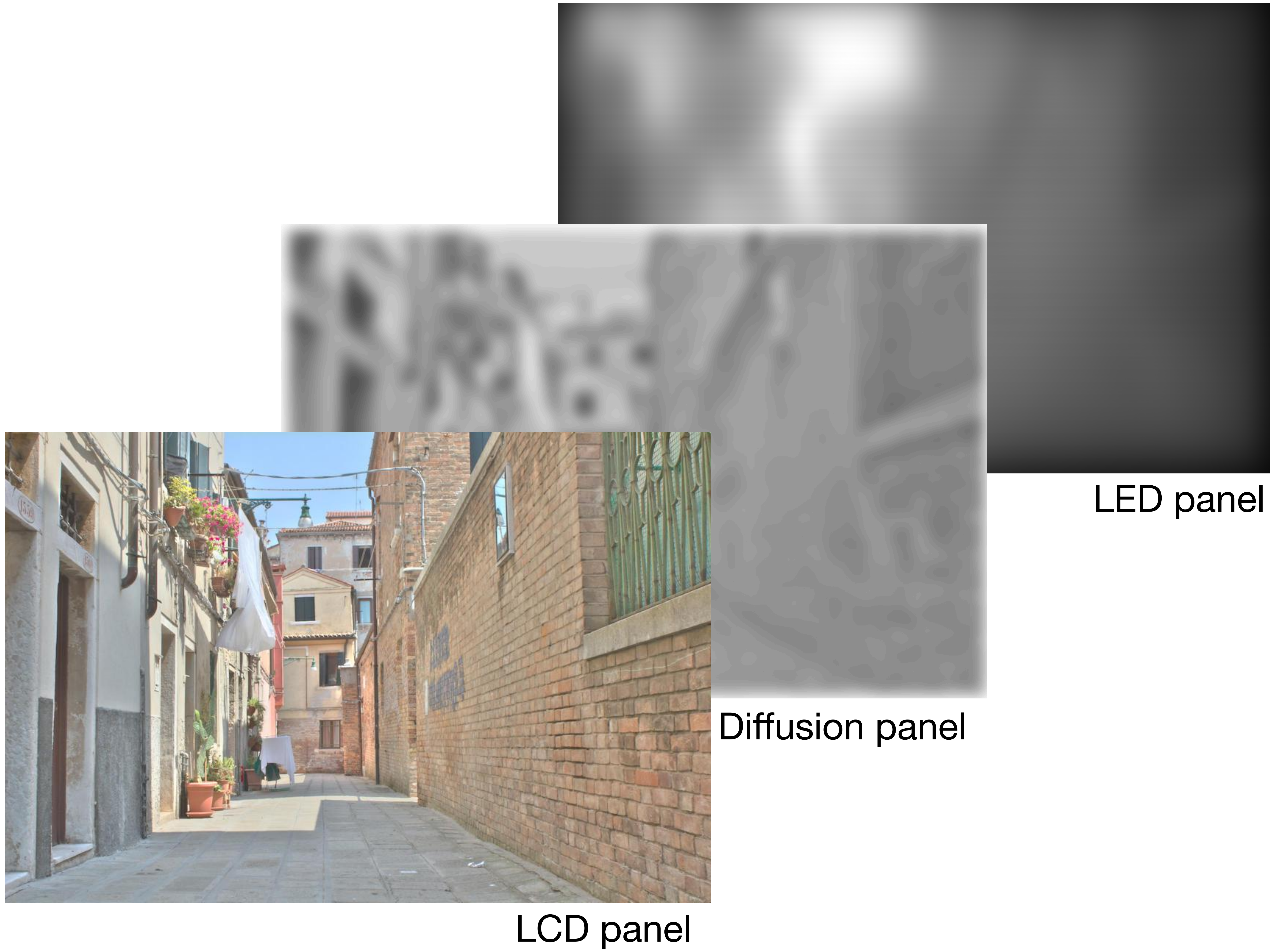
Modern High Dynamic Range Imaging at the Time of Deep Learning

Visualisation

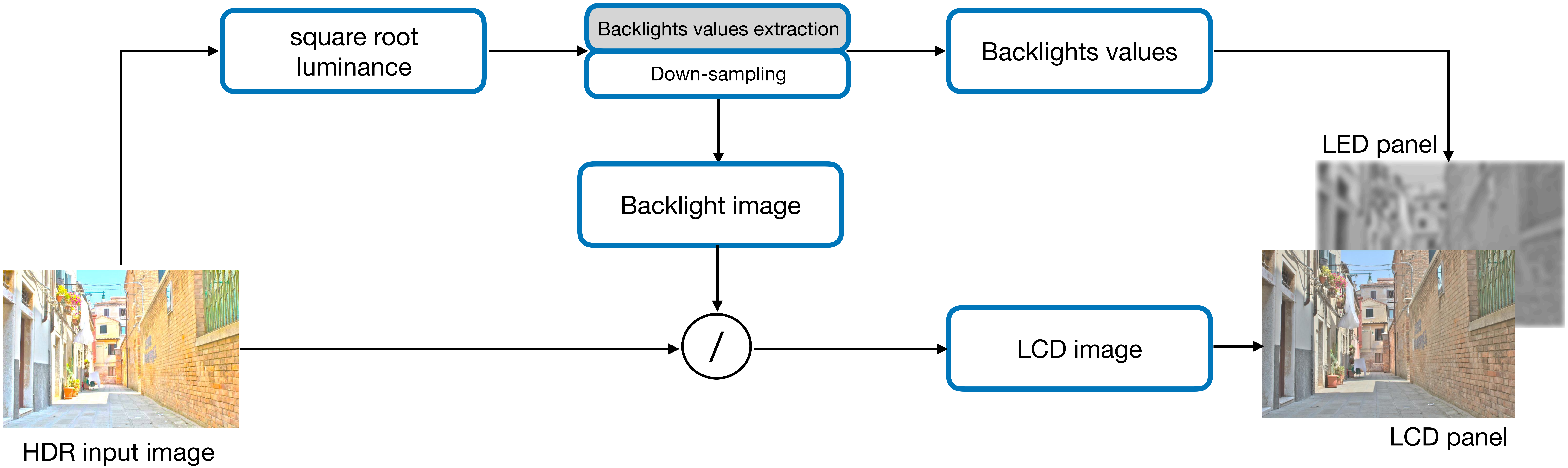
Francesco Banterle and Alessandro Artusi

HDR Direct Visualisation - HDR Display

HDR Display: Modulating Backlight

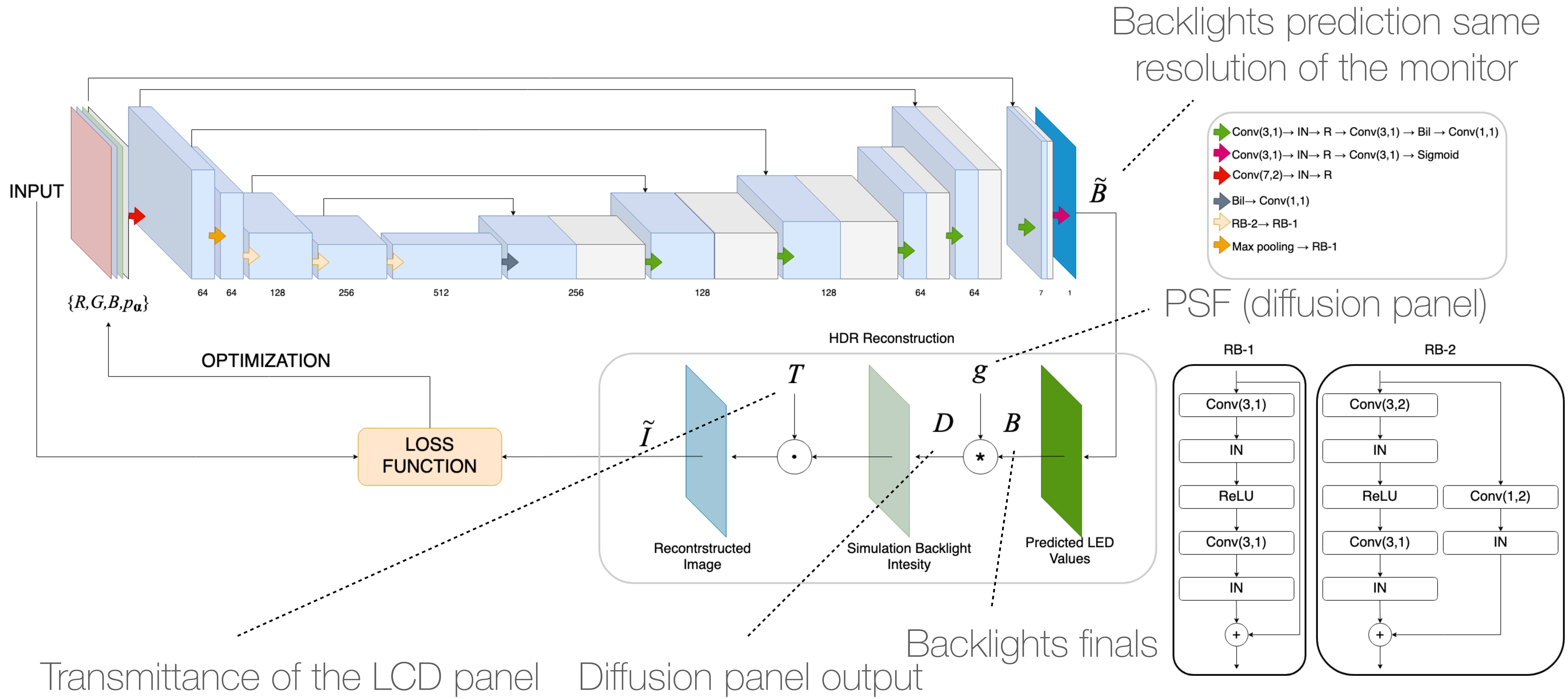


Baseline Method for Backlight Display



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

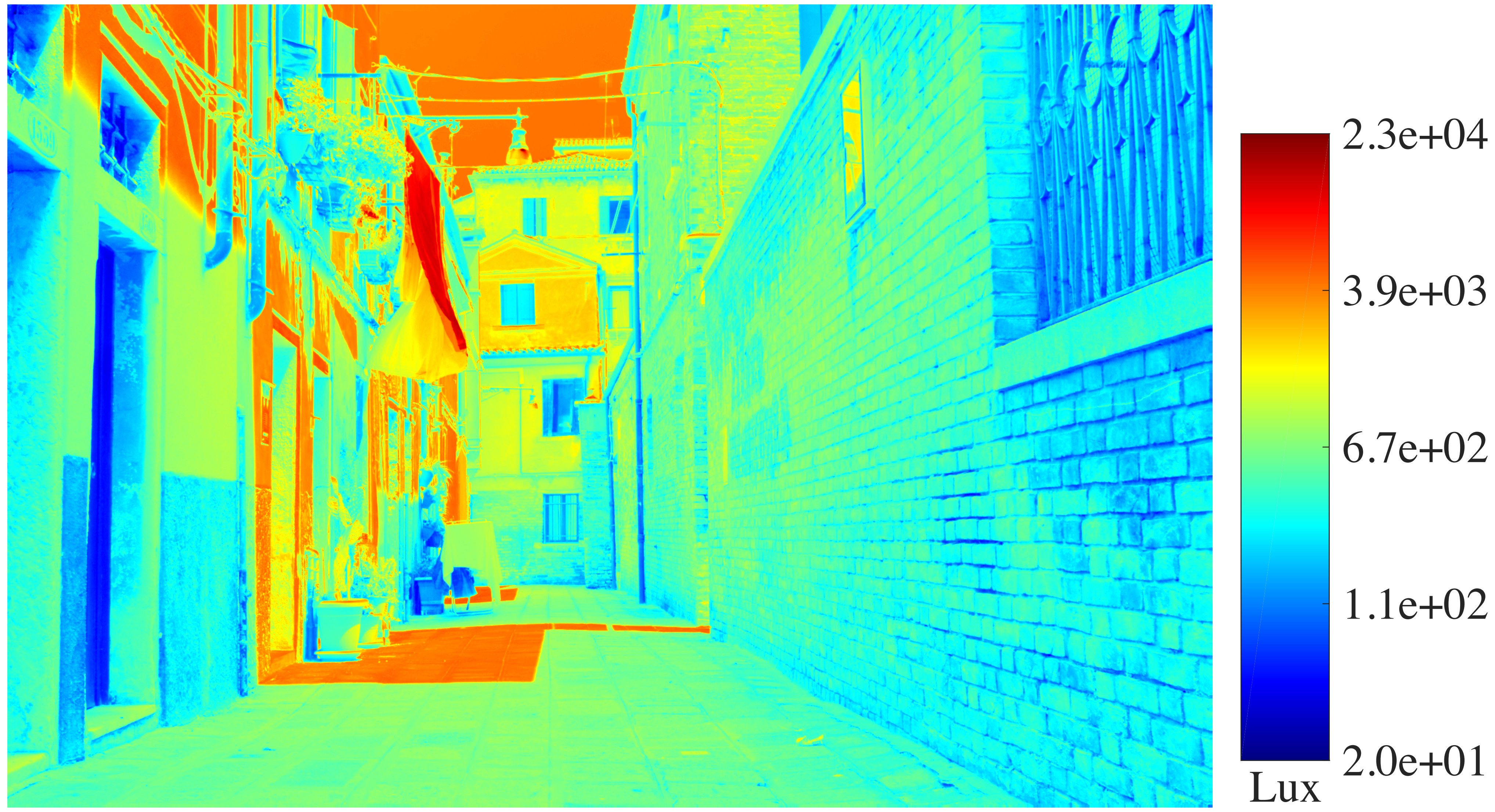
Deep-learning Approach for BLD



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

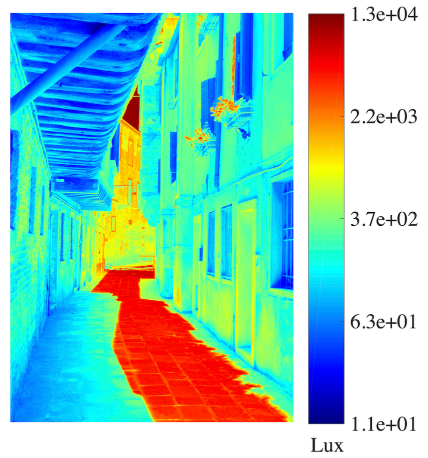
8-bit Tone Mapped Image

The Full Pipeline

The Full Pipeline

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

The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

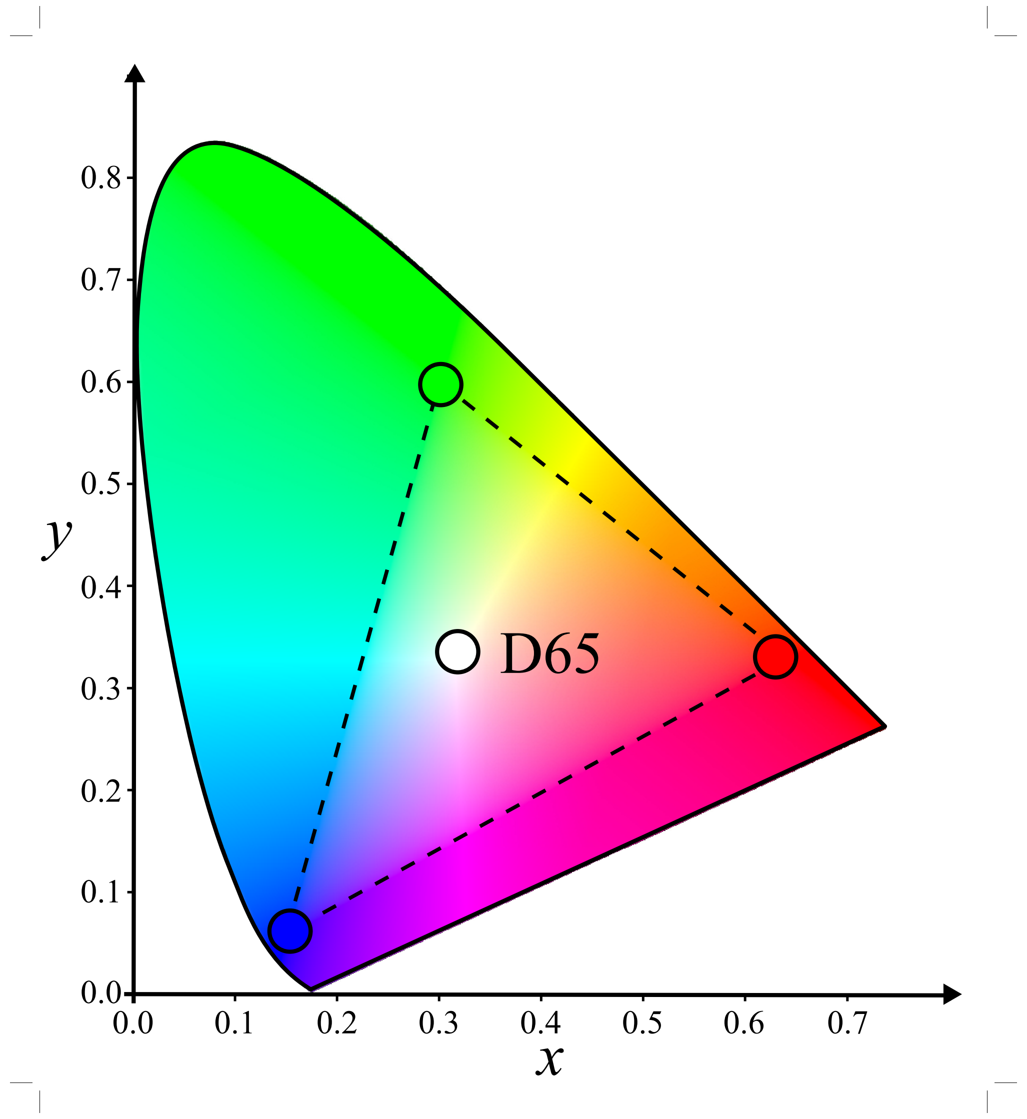
$$Y_{HDR} = w_1 R + w_2 G + w_3 B$$

$sRGB:$

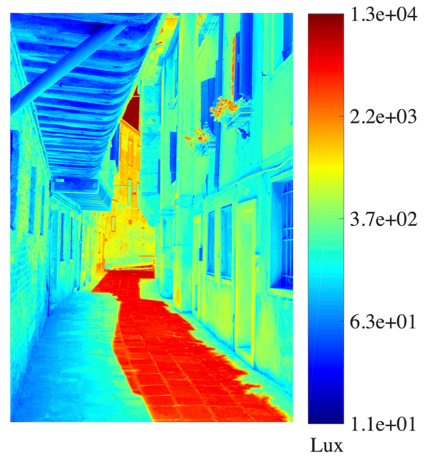
$w_1 = 0.2126$

$w_2 = 0.7152$

$w_3 = 0.0722$



The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

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

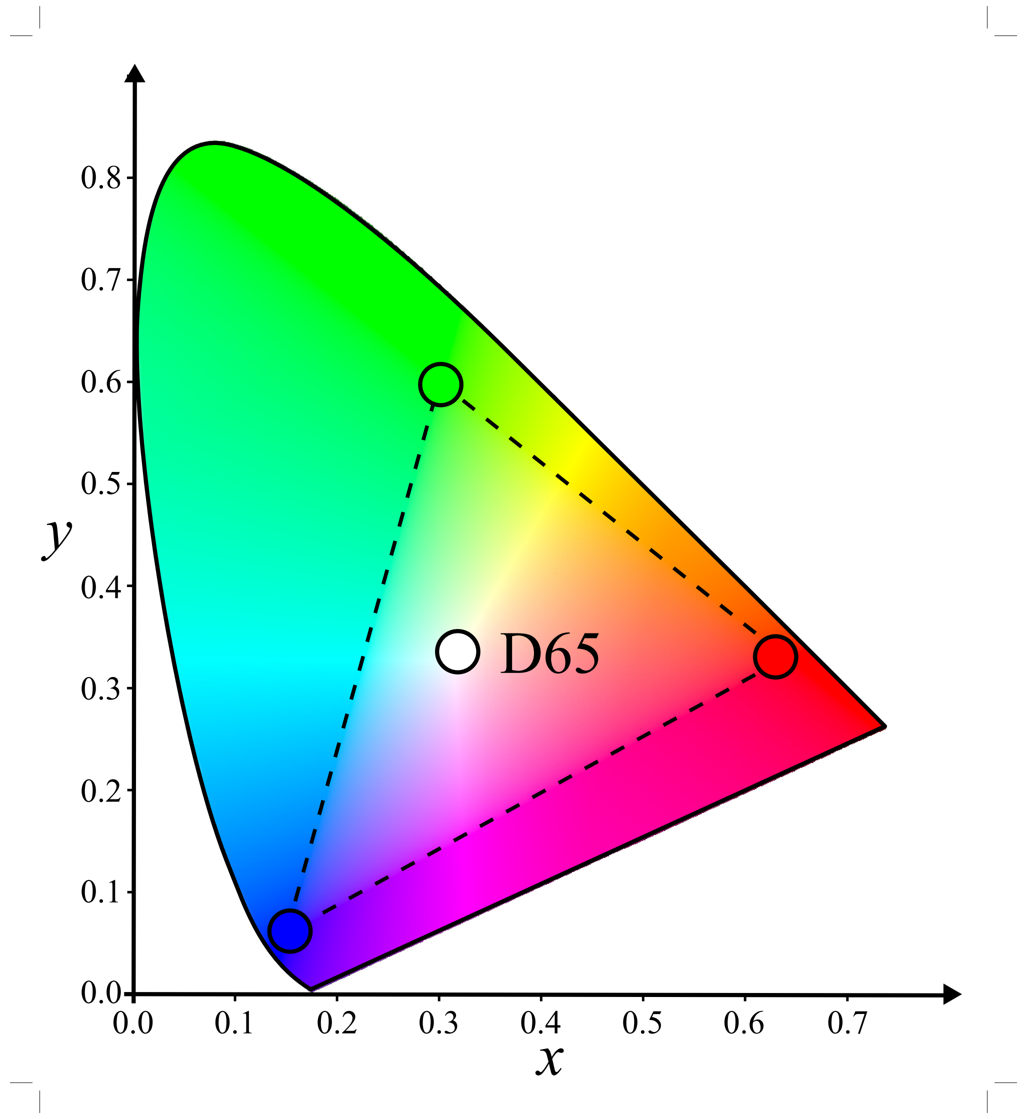
$$Y_{SDR} = F(mY^{\gamma}_{HDR})$$

$sRGB:$

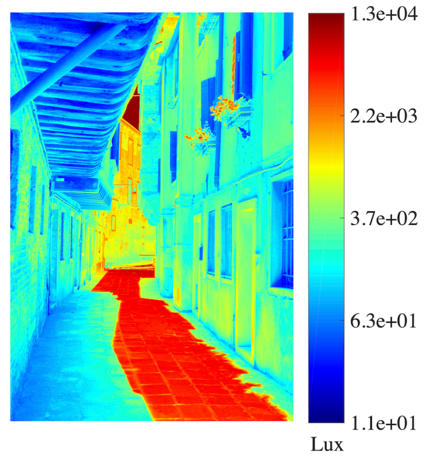
$w_1 = 0.2126$

$w_2 = 0.7152$

$w_3 = 0.0722$



The Full Pipeline



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

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

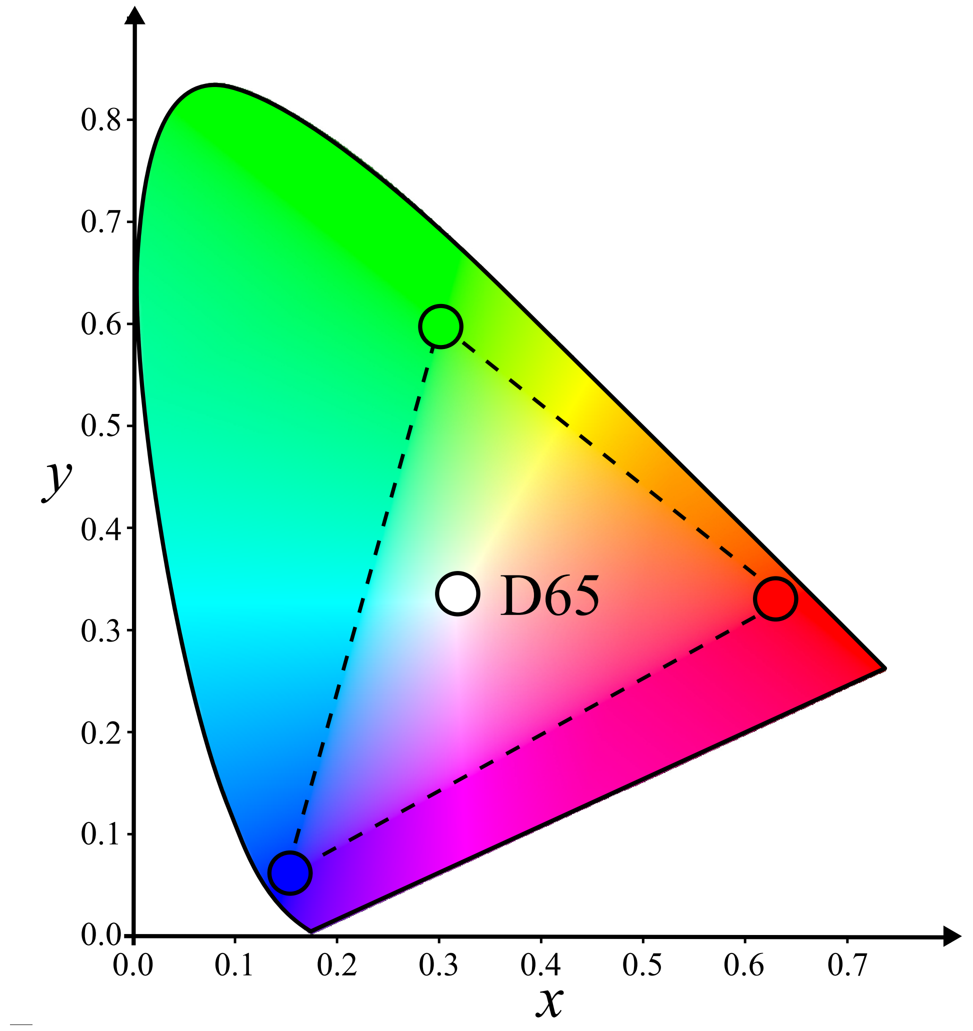
$$Y_{SDR} = F(m Y^{\gamma}_{HDR})$$

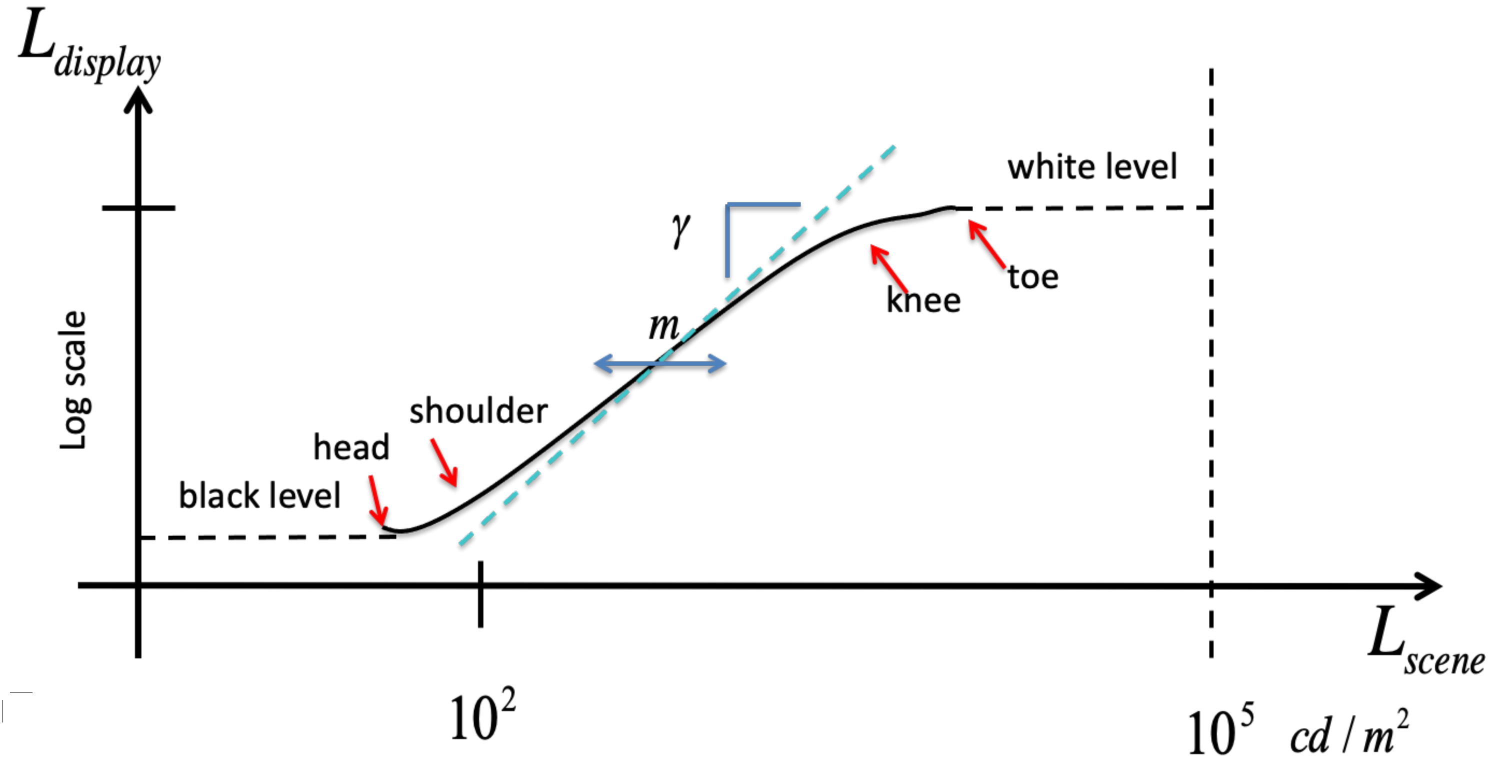
$sRGB:$

$w_1 = 0.2126$

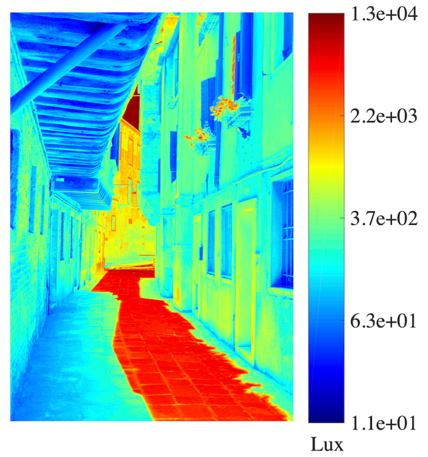
$w_2 = 0.7152$

$w_3 = 0.0722$





The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

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

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

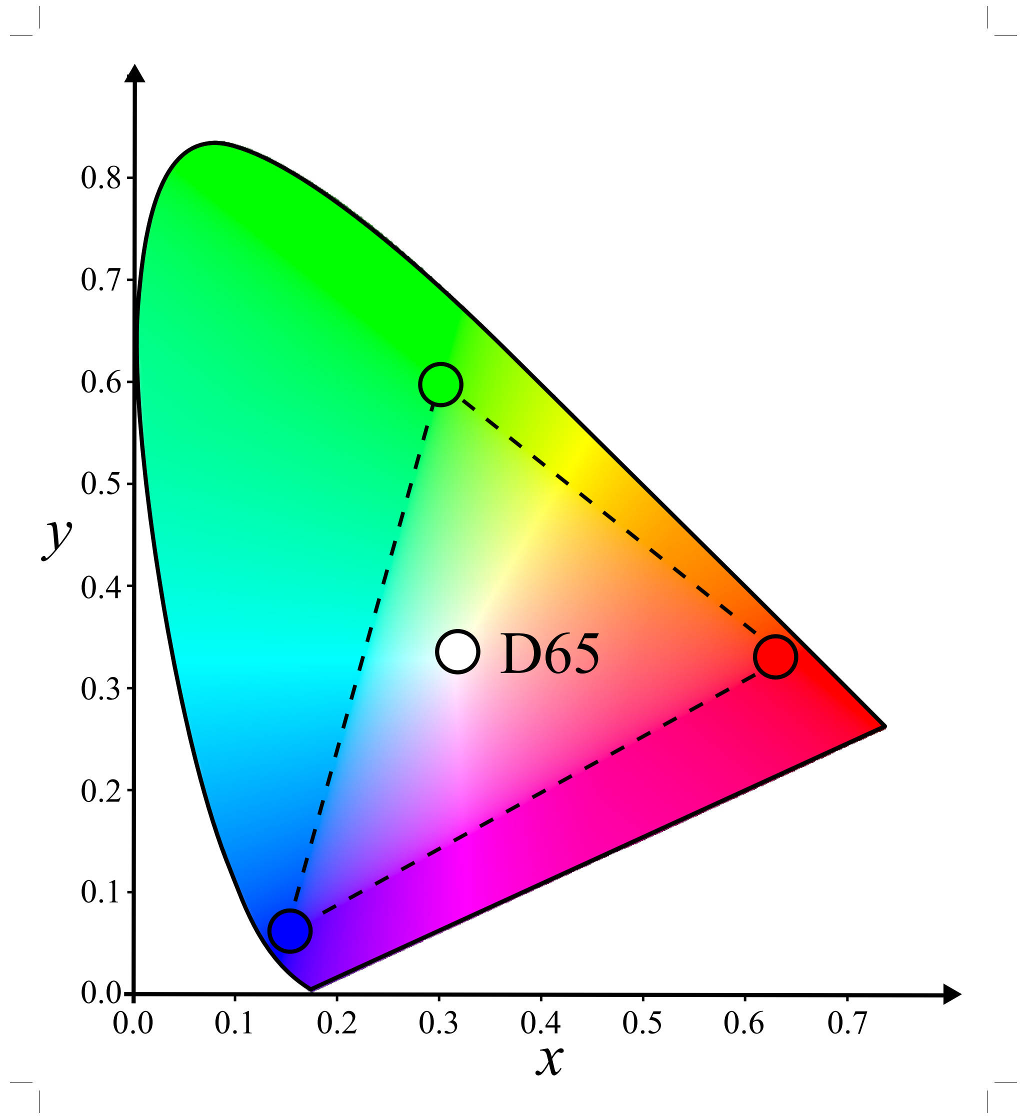
$$Y_a = G_s(Y_{HDR})$$

$sRGB:$

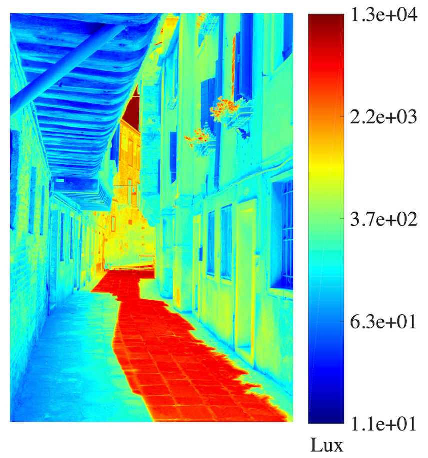
$w_1 = 0.2126$

$w_2 = 0.7152$

$w_3 = 0.0722$



The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

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

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

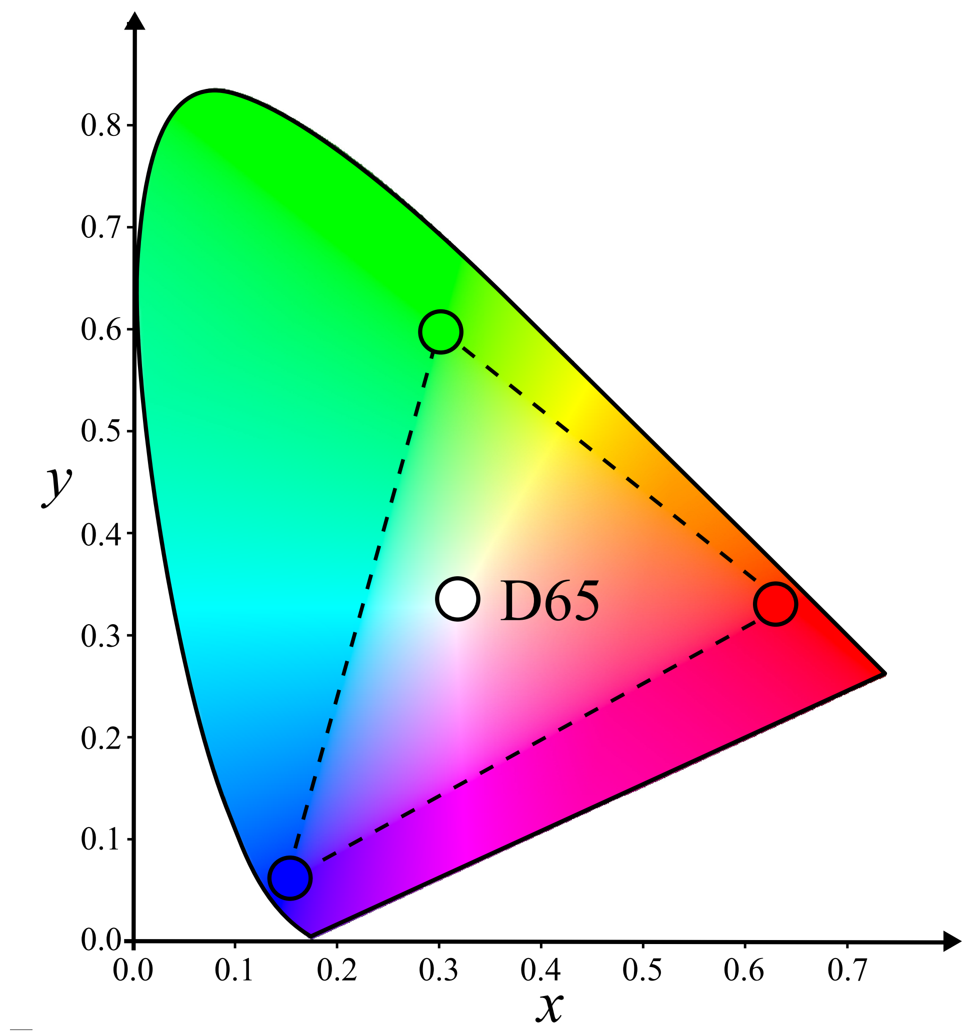
$$Y_a = G_s(Y_{HDR})$$

$sRGB :$

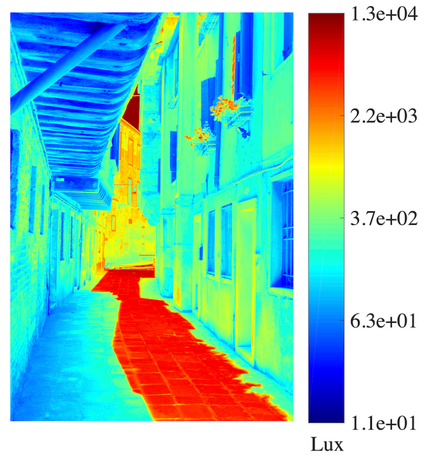
$w_1 = 0.2126$

$w_2 = 0.7152$

$w_3 = 0.0722$



The Full Pipeline



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

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

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

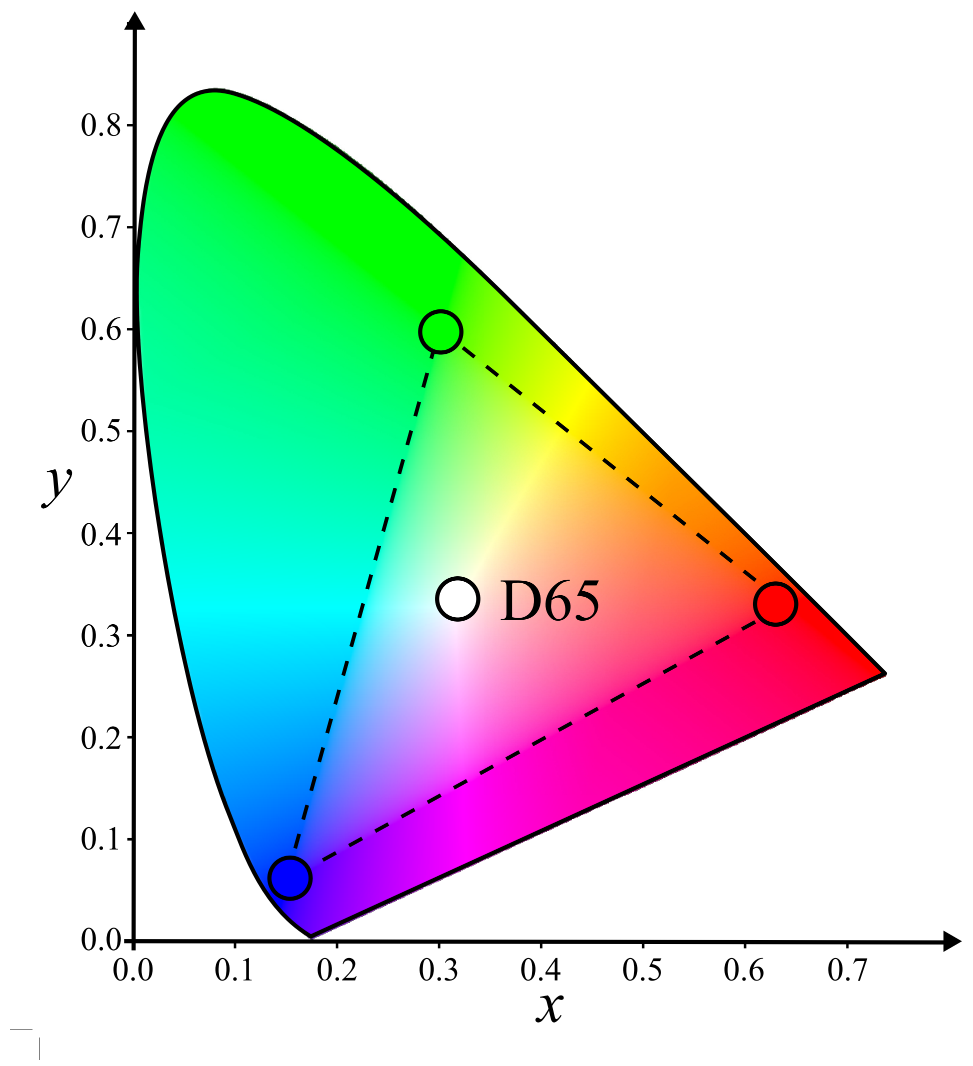
$$Y_a = G_s(Y_{HDR})$$

$sRGB:$

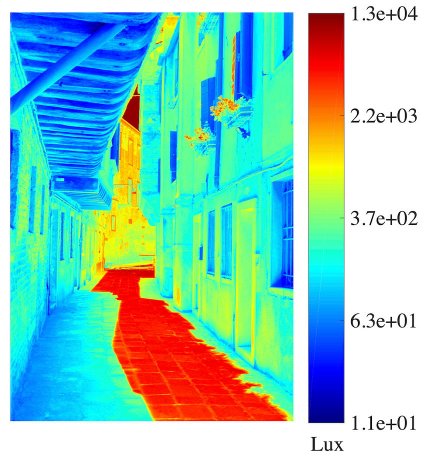
$w_1 = 0.2126$

$w_2 = 0.7152$

$w_3 = 0.0722$



The Full Pipeline



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

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

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

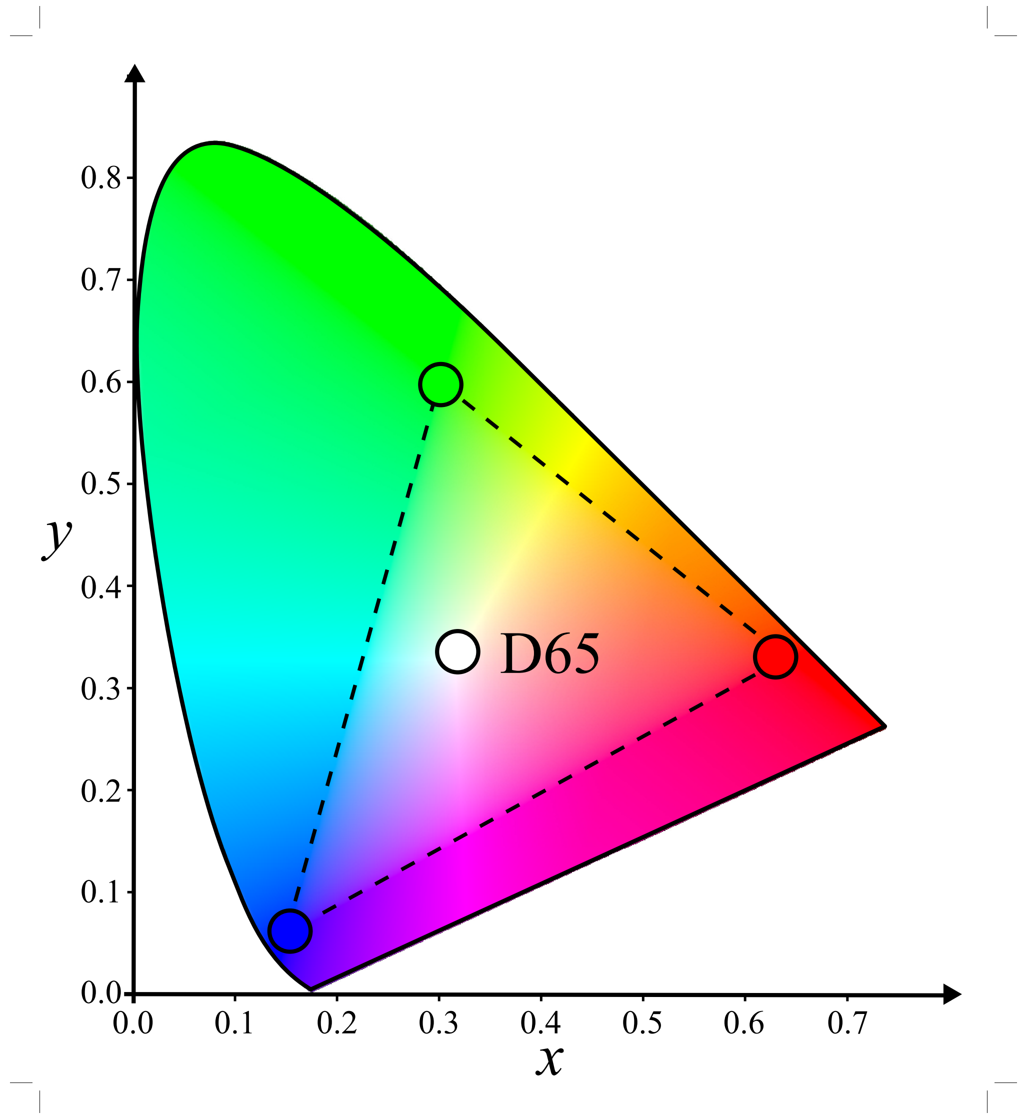
$$Y_a = G_s(Y_{HDR})$$

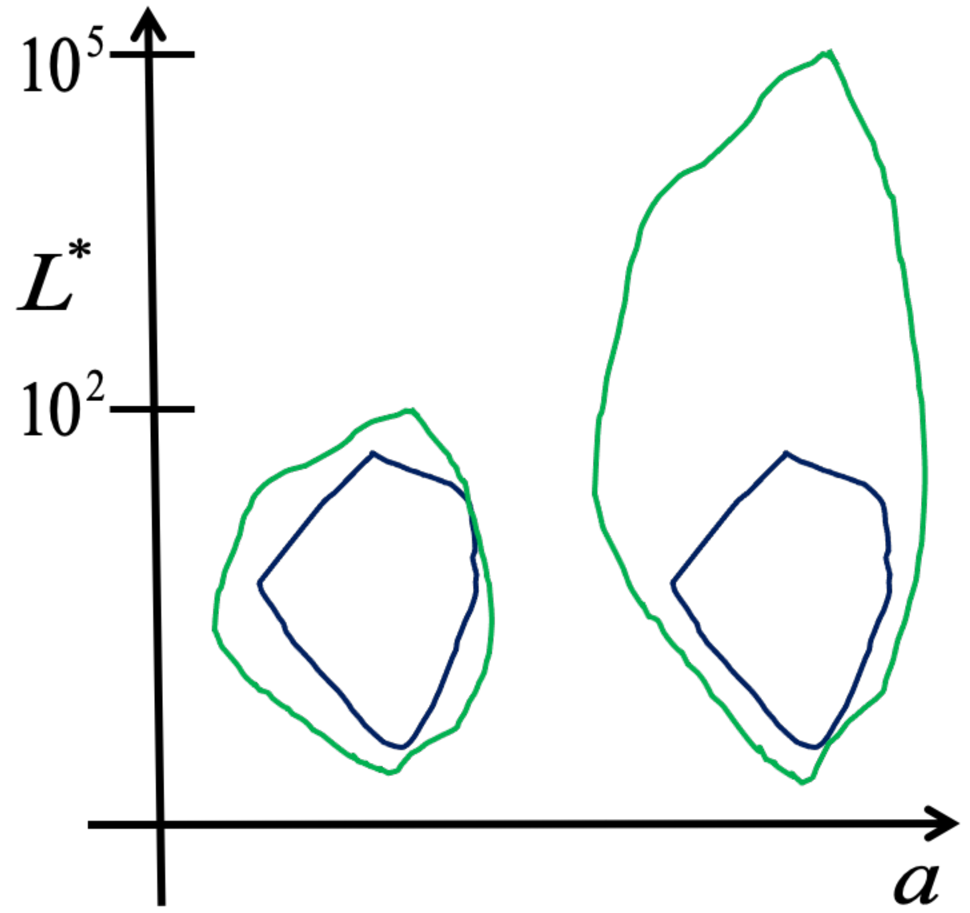
$sRGB:$

$w_1 = 0.2126$

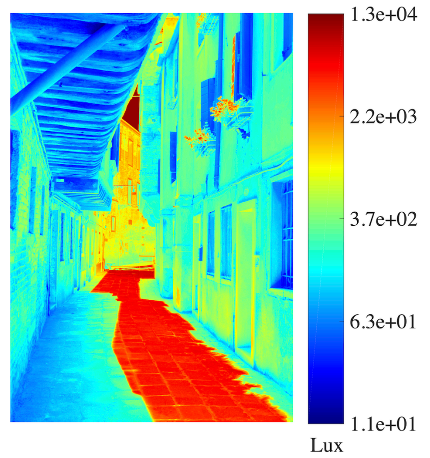
$w_2 = 0.7152$

$w_3 = 0.0722$





The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

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

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

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

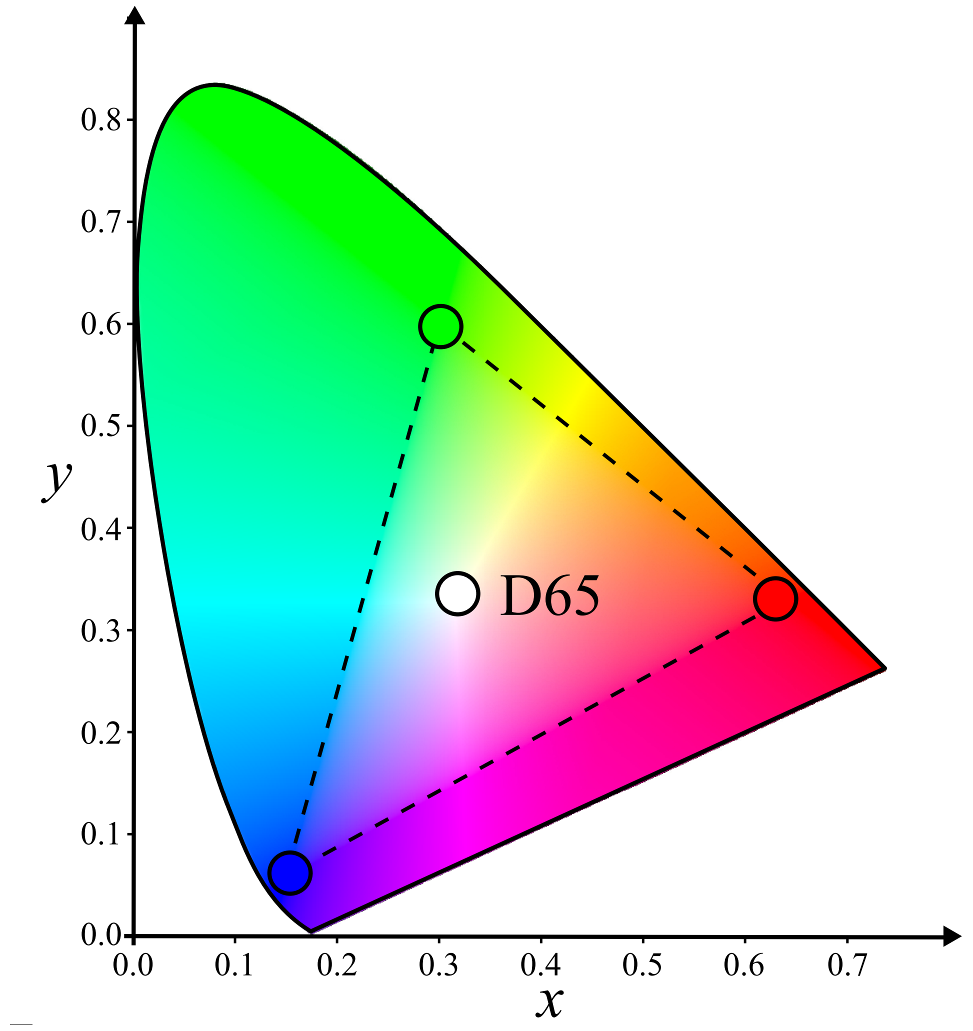
$$RGB_{SDR} = \left(\frac{RGB_{HDR}}{Y_{HDR}}\right)^s Y_{SDR}$$

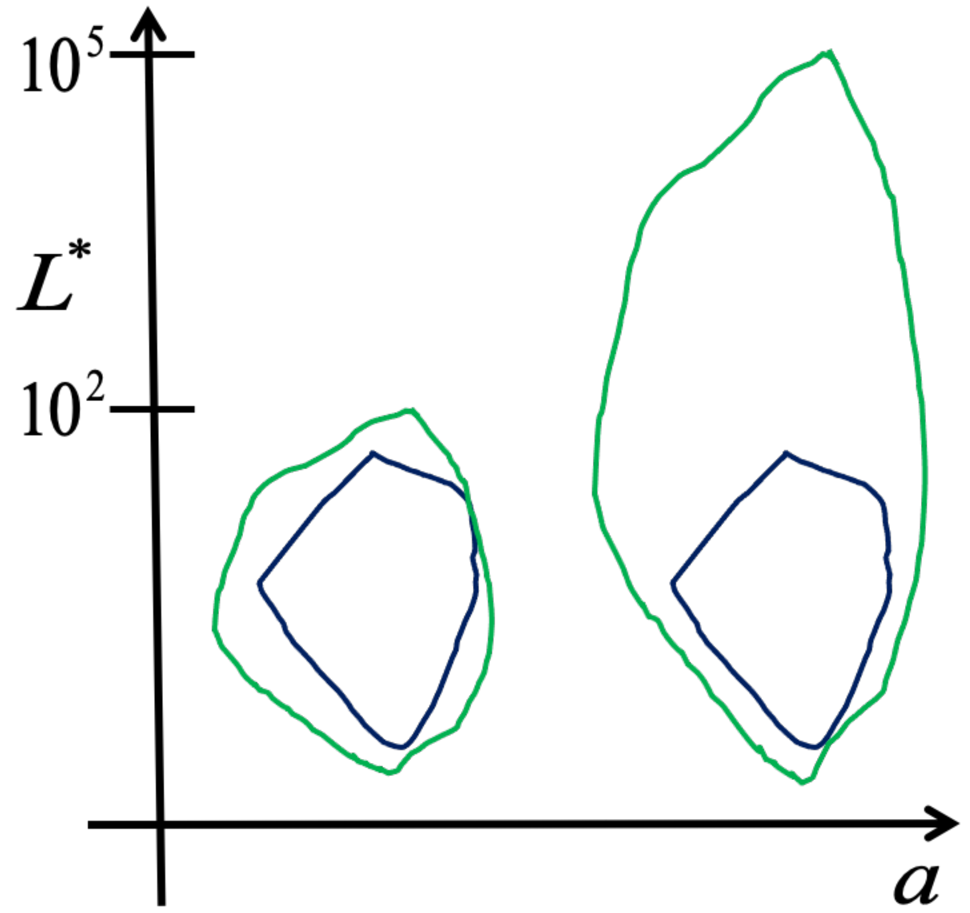
$sRGB:$

$w_1 = 0.2126$

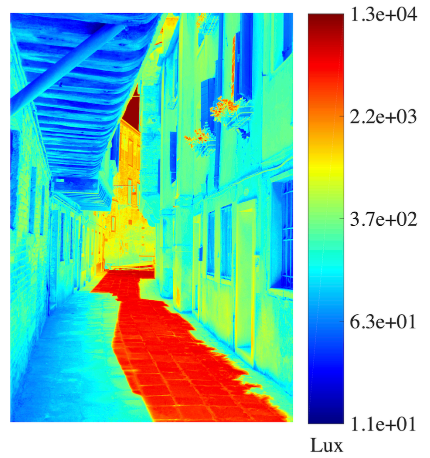
$w_2 = 0.7152$

$w_3 = 0.0722$





The Full Pipeline



RGB → Y

Luminance mapping

Color mapping

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

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

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

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

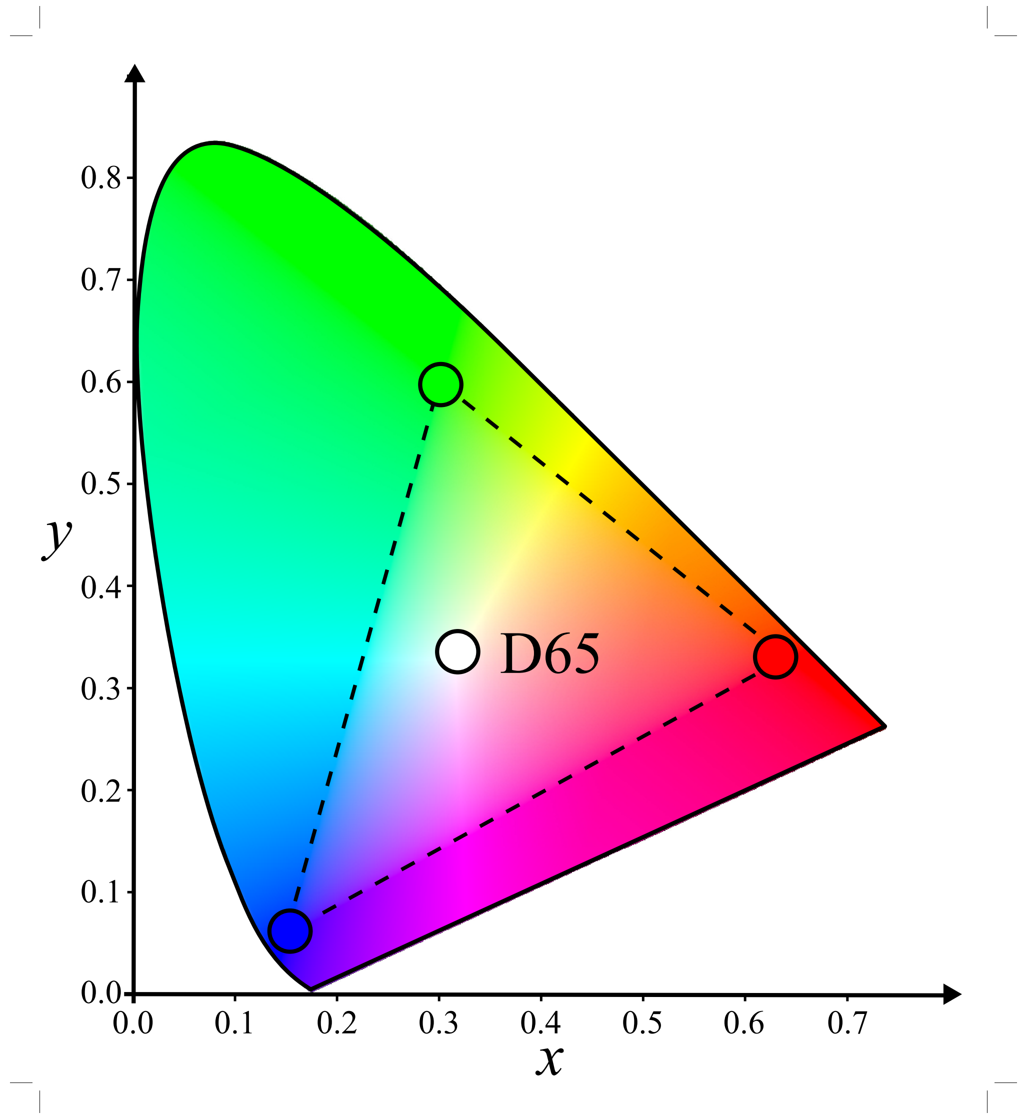
$$RGB_{SDR} = \left(\left(\frac{RGB_{HDR}}{Y_{HDR}} - 1 \right) s + 1 \right)^s Y_{SDR}$$

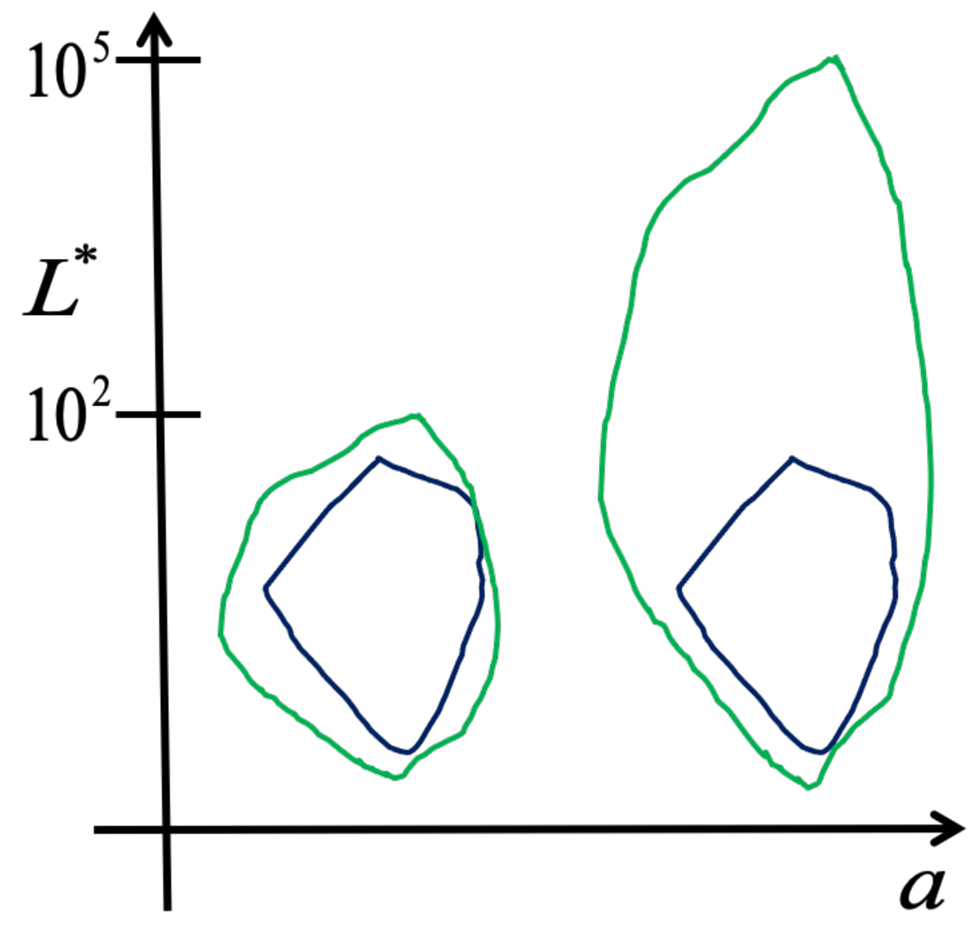
$sRGB:$

$w_1 = 0.2126$

$w_2 = 0.7152$

$w_3 = 0.0722$





Aims/Goals

Aims/Goals

- **Quality optimisation**
  - 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

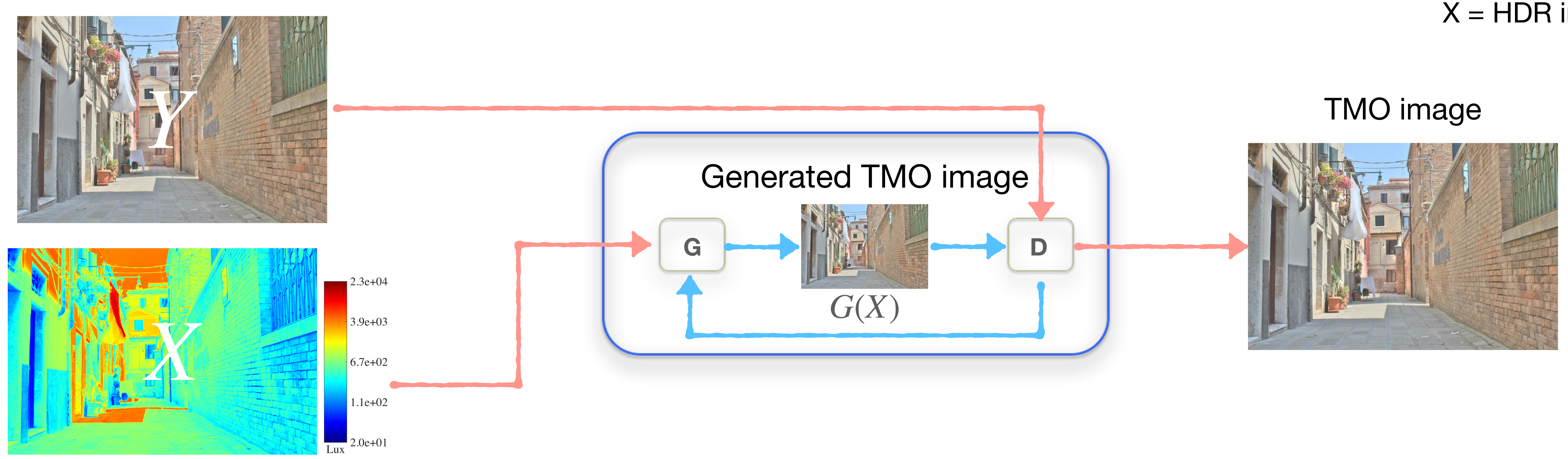
D = Discriminator

Y = Ground truth SDR

X = HDR input

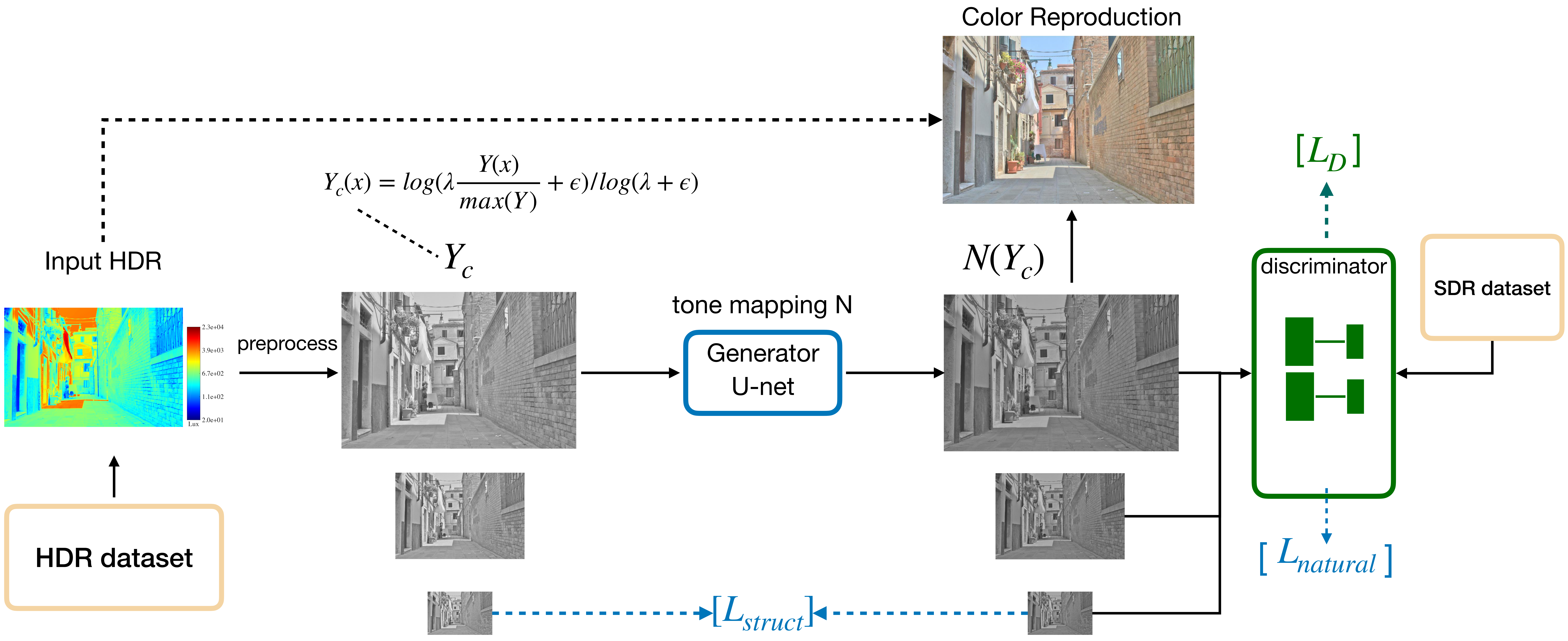
$$L_{GAN}(G, D) = \mathbb{E}_y[logD(Y)] + \mathbb{E}_x[1 - logD(G(X))]$$

LDR-TMO image



Scene-referred HDR image

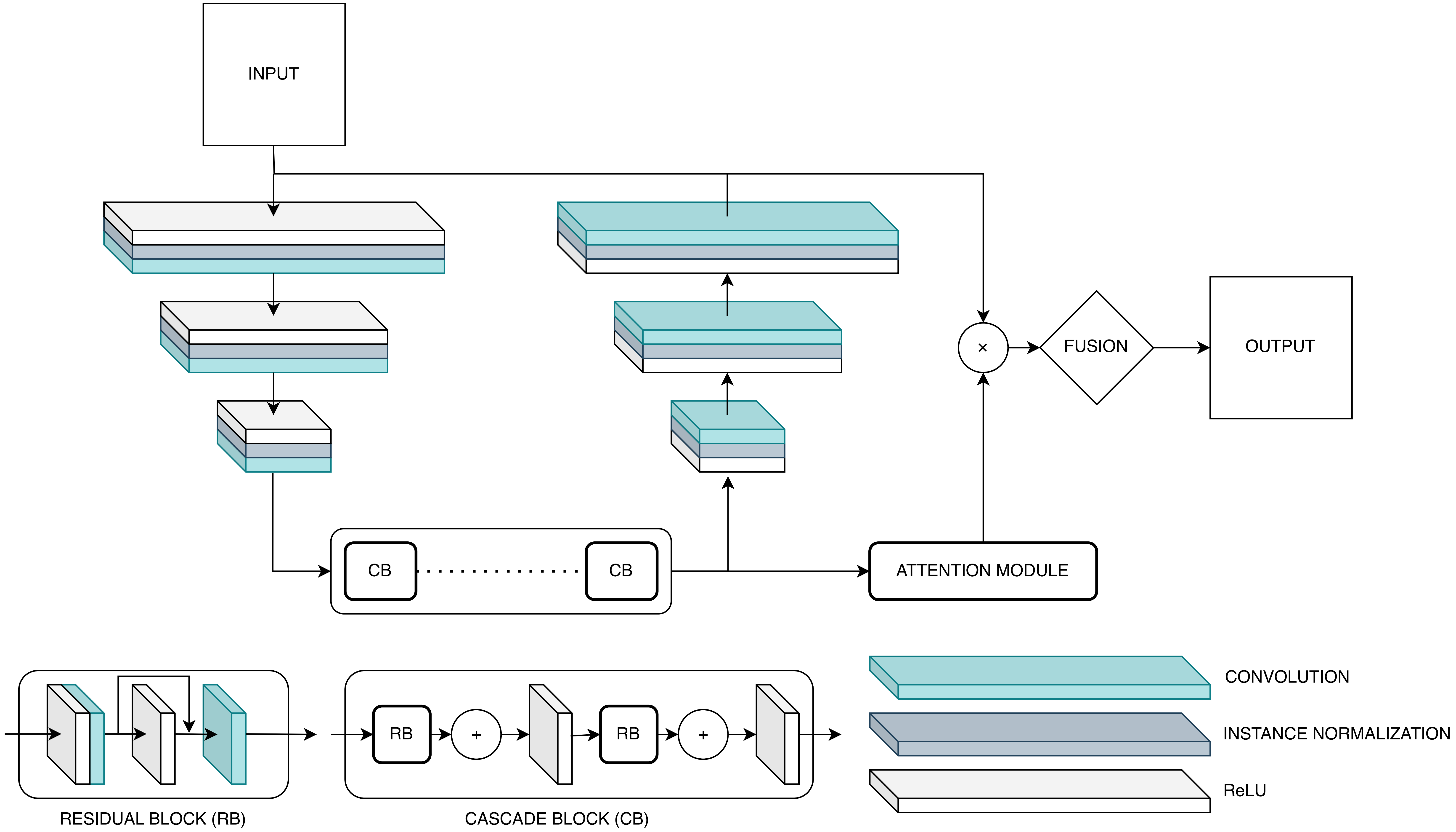
Architectures - Generative Adversarial Network Ref: VHF-2021



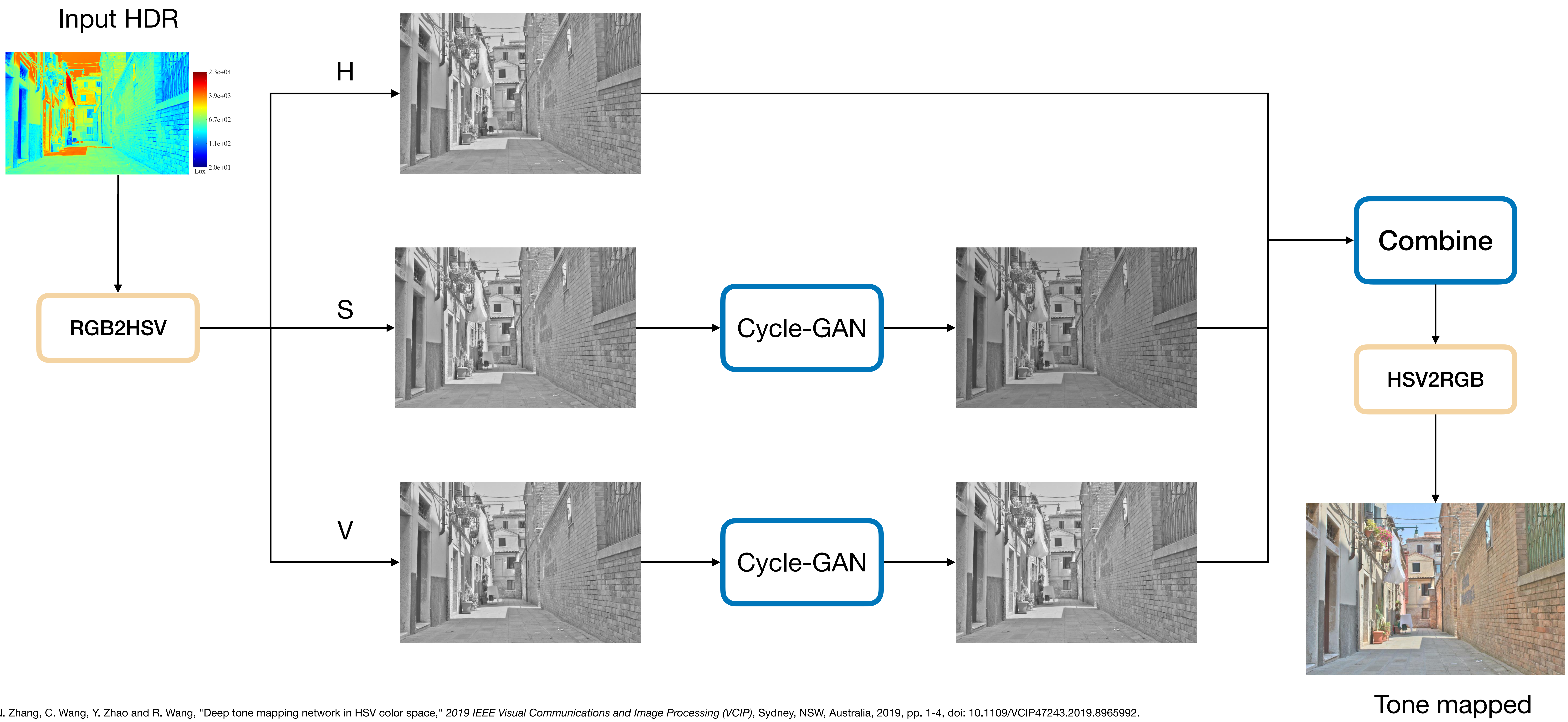
VINKER Y., HUBERMAN-SPIEGELGLAS I., FATTAL R.: Unpaired learning for high dynamic range image tone mapping. In 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021 (2021), IEEE, pp. 14637–14646. URL: https://doi.org/10.1109/ICCV48922.2021.01439.

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)



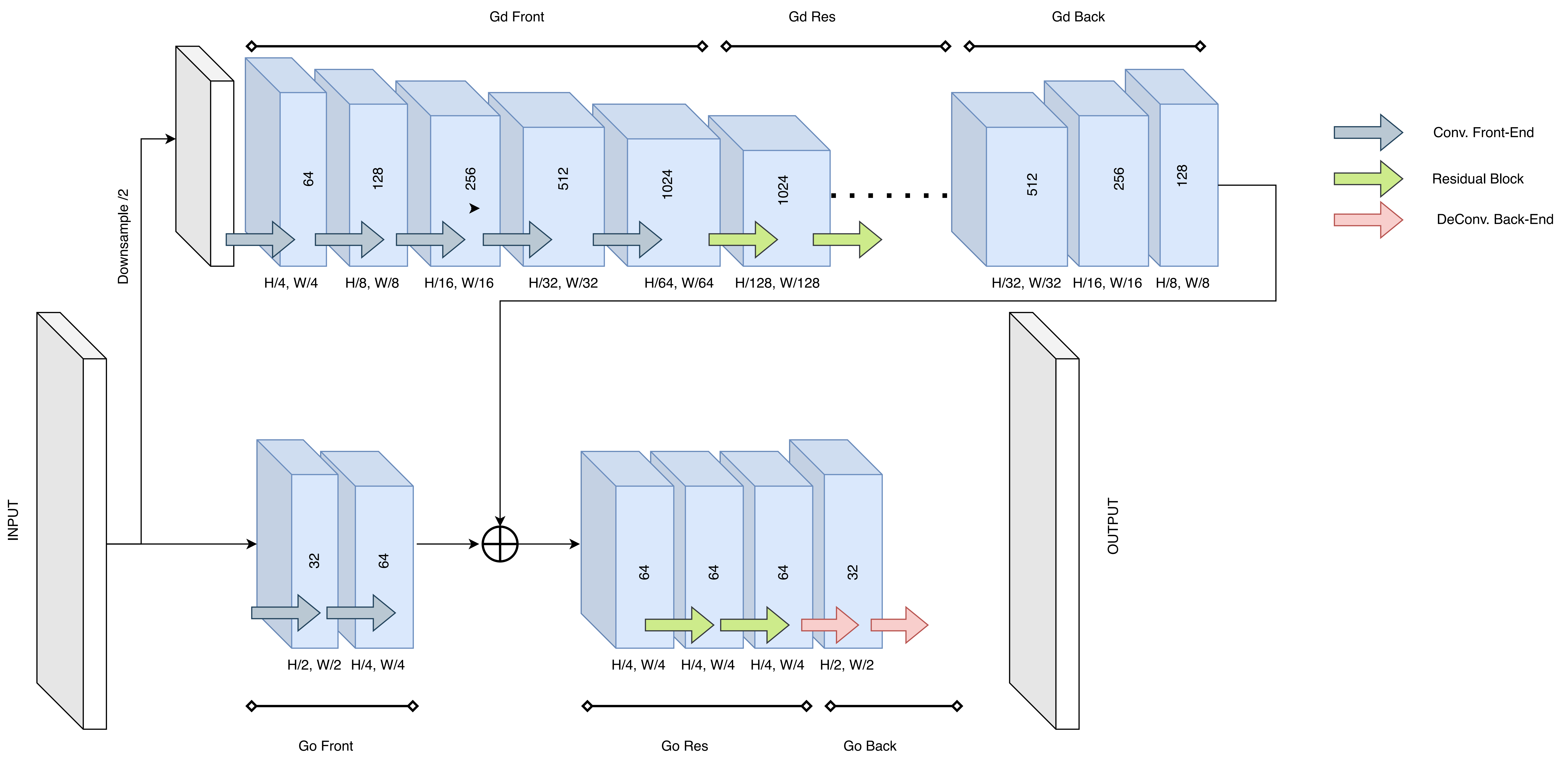
Architectures - Cycle-GAN Approach Ref: ZZWW-2020-2022



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.

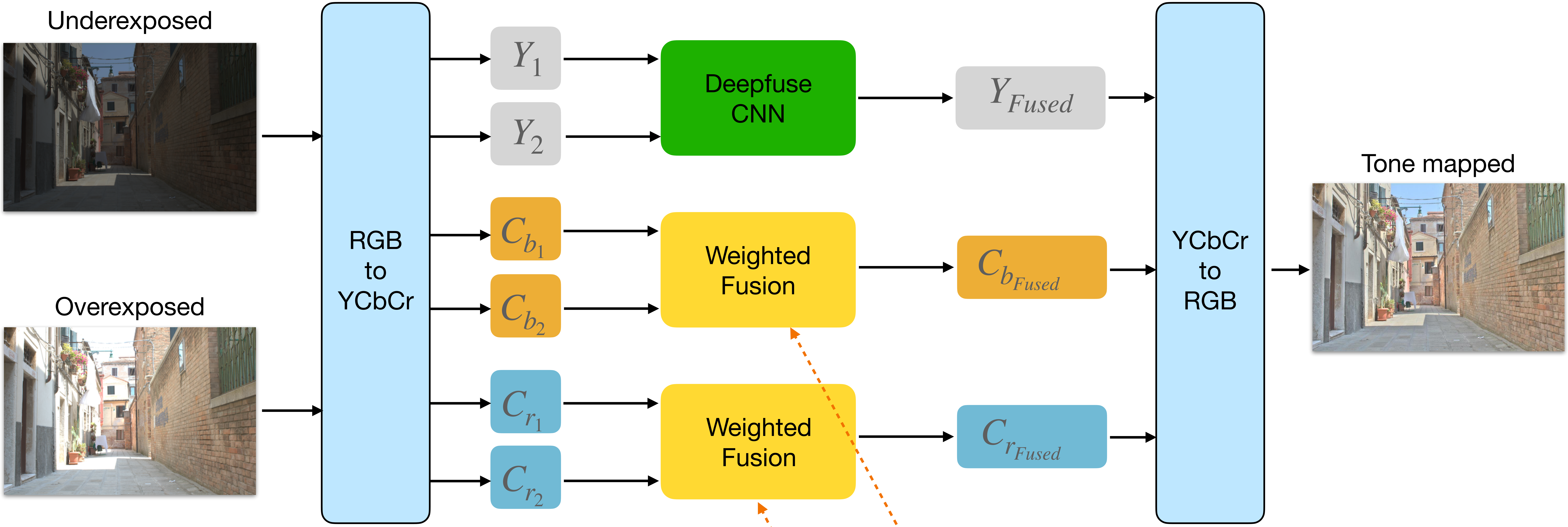
ZHANG N., ZHAO Y., WANG C., WANG R.: A real-time semi-supervised deep tone mapping network. IEEE Trans. Multim. 24 (2022), 2815–2827. URL: https://doi.org/10.1109/TMM.2021.3089019, doi:10.1109/TMM.2021.3089019. 2

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



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

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

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

Constant value =128

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

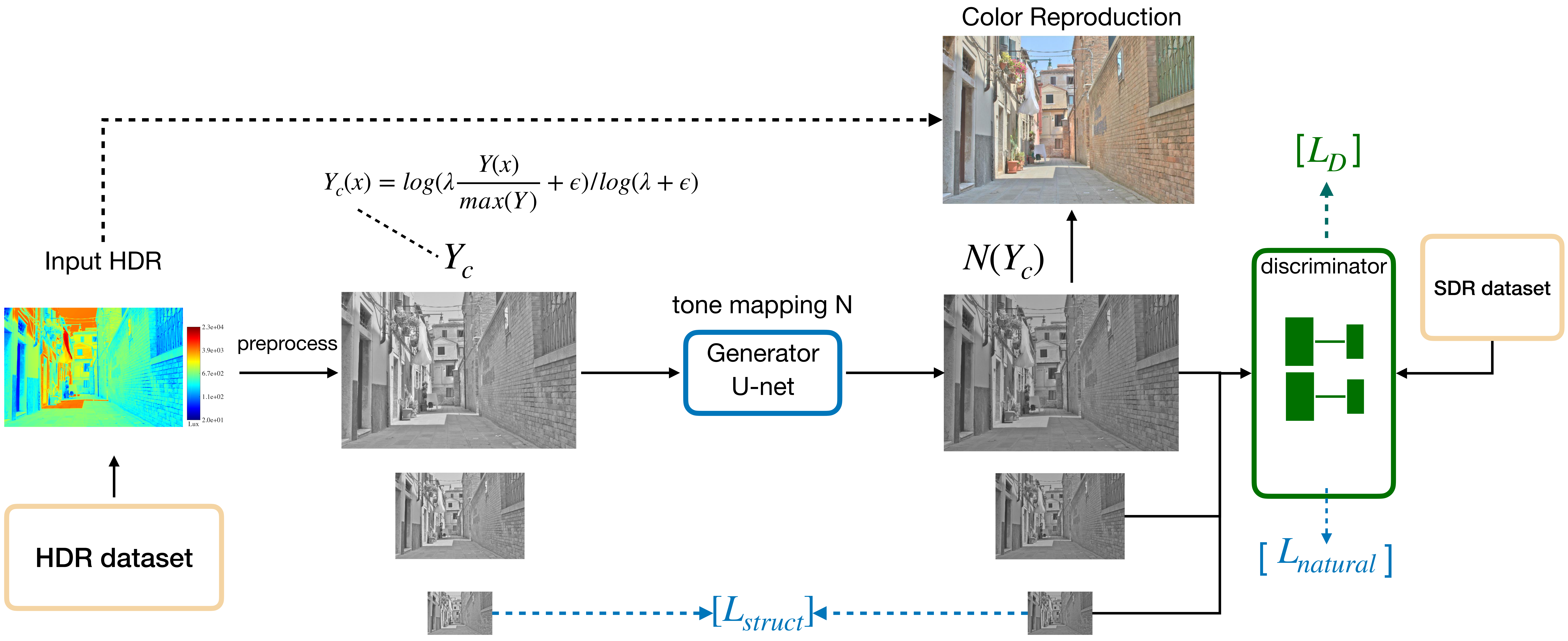
The Loss Function- General Approach

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

Adversarial Loss:
•Image appearance matching

Other Loss functions:
- To preserve the content and structure
- Pixel-wise loss
- Perceptual loss, feature matching, gradient, etc.

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.

The Loss Function Ref: VHF-2021

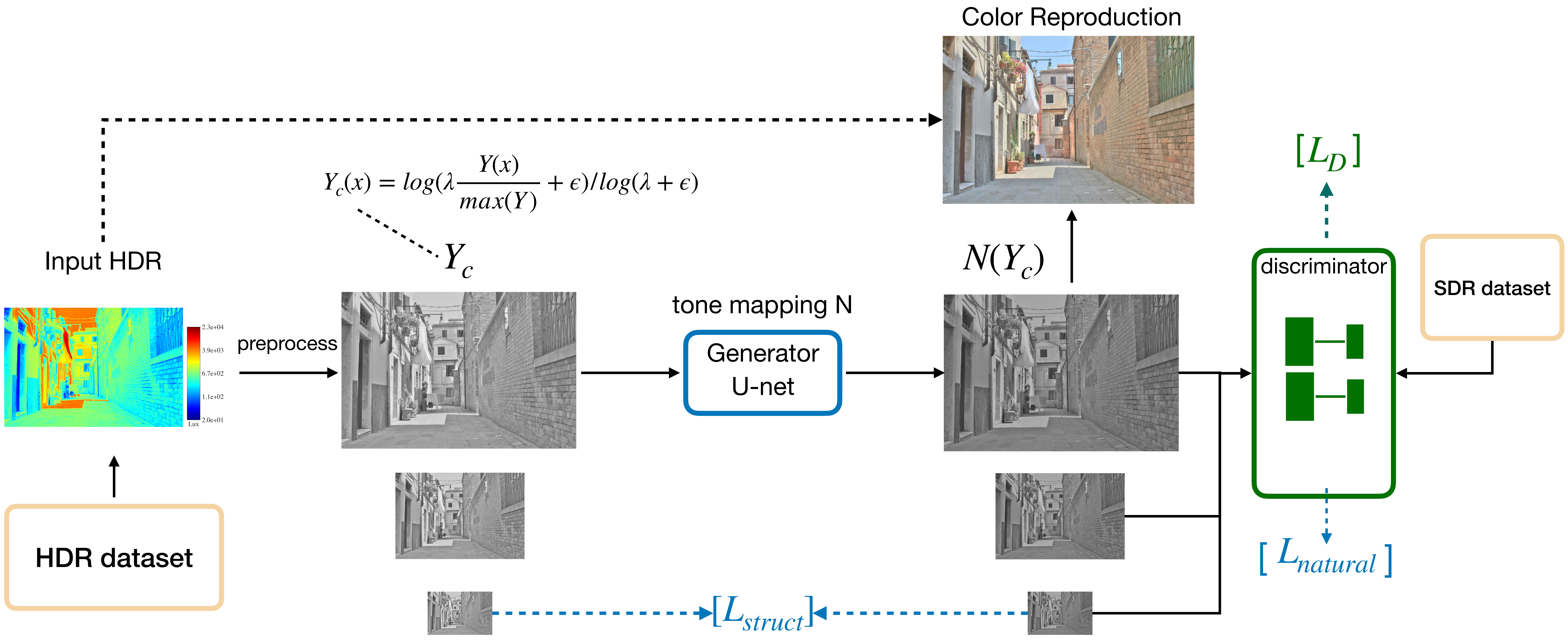
- Vinker et al. 2021 do not train the generator (N) to conceive new images from scratch:
    - Removing biases in Y_c with respect to regular SDR images.
- Discriminator, 3 applied at different image scale ($\downarrow^k$ bicubic downscaling $\times 2^k$)

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

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

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

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.

The Loss Function Ref: VHF-2021

- Vinker et al. 2021 proposed to preserve the content and structure:
  - Measure based on Pearson correlation on two images I *and* J :

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

  - Loss 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:

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

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

- Feature matching loss:

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

T is the number of layer
N_i is number of elements in each layer
$D^{(i)}$ is the 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:

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

$(\cdot)^{\tau}$ is the transpose operator

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

H, W are 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):

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

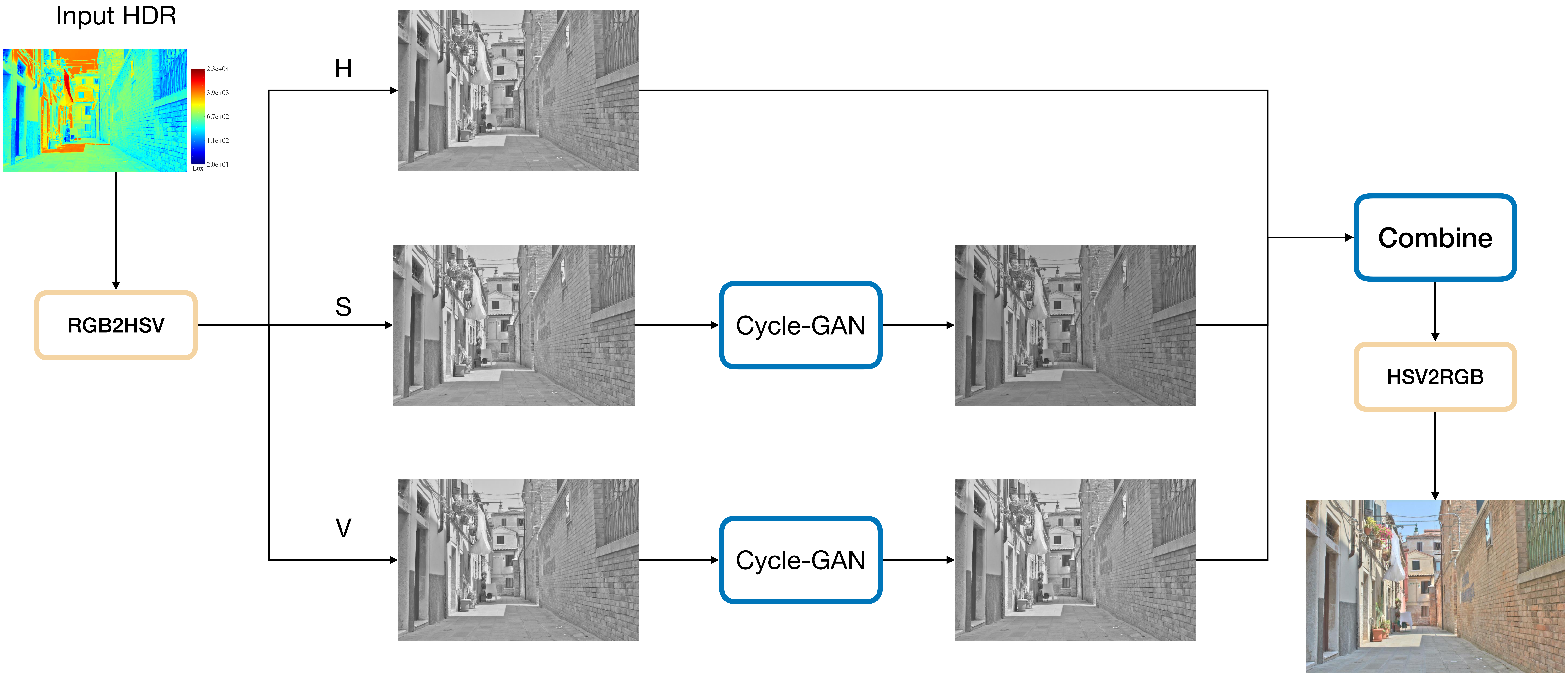
$F^{(i)}$ i-th layer of the VGG19 network
M_i i-th element of the layer

- Feature matching loss (same as PKO 2021):

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

T is the number of layer
N_i is number of elements in each layer
$D^{(i)}$ is the i-th layer feature extractor of the discriminator

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

The Loss Function Ref: ZZWW-2022

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

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

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

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

Others Loss Functions

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.

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

Architectures - Autoencoder Ref: WCS-2022

The Loss Function Ref: WCS-2022

- Wang et al. 2022: L1 norm based masked features contrast maps

feature magnitude at pixel p

Feature contrast

gaussian filtered feature value for patch P centre at pixel p

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

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

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

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

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

Feature contrast neighborhood-masking

$$M_n = \frac{\sigma_p}{|\mu_p| + \epsilon}$$

f_p

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

Compression power factor

Feature contrast self-masking

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

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?

Thank you for your attention!

Any Question?