

HDR-LFNet: Inverse Tone Mapping using Fusion Network

Mathieu Chambe, Ewa Kijak, Zoltan Miklos, Olivier Le Meur, Rémi Cozot, Kadi Bouatouch

▶ **To cite this version:**

Mathieu Chambe, Ewa Kijak, Zoltan Miklos, Olivier Le Meur, Rémi Cozot, et al.. HDR-LFNet: Inverse Tone Mapping using Fusion Network. 2022. hal-03618267

HAL Id: hal-03618267
https://hal.science/hal-03618267

Preprint submitted on 24 Mar 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Volume xx (200y), Number z, pp. 1–11

HDR-LFNet: Inverse Tone Mapping using Fusion Network

M. Chambe[1] and E. Kijak[1] and Z. Miklos[1] and O. Le Meur[1] and R. Cozot[2] and K. Bouatouch[1]

[1]Univ Rennes, CNRS, IRISA, Rennes, France
[2]Littoral Opal Coast University, Calais, France

Abstract

To capture the real-world luminance values, High Dynamic Range (HDR) image processing has been developed. HDR images have a richer content than the widely-used Standard Dynamic Range (SDR) images, and are used in a number of situations, e.g. in film industry. As HDR displays are more and more commercially available, we need to be able to process HDR images as well as SDR ones (for example, devising denoising algorithms, inpaiting or anti-aliasing). The most powerful methods to process images are now deep neural networks. However, the training of such networks calls for a lot of images, and HDR images datasets are relatively small.

One way to generate HDR images is inverse tone mapping operators (iTMOs). They are algorithms that expand the dynamic range of SDR images. In this paper, we propose HDR-LFNet, a novel iTMO, and its HDR training dataset. Our method merges several existing handcrafted iTMOs, combined in a supervised neural network to produce an HDR output. Our lightweight network requires less training images than state-of-the-art methods, and has faster training phase. Besides, the quality of the generated images is similar to the state-of-the-art. We present the architecture as well as the subjective and experimental evaluations of our method.

CCS Concepts

• ***Computing methodologies*** → ***Computational photography; Image processing;*** *Supervised learning;*

1. Introduction

High Dynamic Range (HDR) images consist of BitMap arrays which contain, for each pixel, the raw luminance captured by the sensor at this point. These values are, in theory, unbounded and not quantized. This is a much richer content than Standard Dynamic Range (SDR) images, which correspond to the common widespread images. HDR-compatible hardware, such as monitors or cameras, are more and more available to the general public. This creates a need for image processing algorithms adapted to HDR content, making the HDR imaging a trending research domain.

HDR photographs are usually taken using the exposure fusion technique. We take several photographs of the same scene with different exposure times. The longer the exposure time, the more light is coming through the camera, and the more information we can obtain in dark parts. On the other hand, short exposure time pictures provide information in the brightest areas of the scene. This allows to gather information in every area of the scene. The different photographs are then merged together to yield a single HDR image. This method is easily achievable using the bracketing function of the camera, that is why it is so popular.

As most of the image datasets available are composed of SDR images, it is useful to devise algorithms to recover lost information in SDR images. From a single SDR image, we must use approximations if we want to extend the dynamic range. Algorithms that extend dynamic range of SDR images are called inverse tone mapping operators (iTMO). Contrary to the classic method of exposure fusion, iTMOs only take as input a single SDR image and yield an HDR image. The result is an approximation of the truth, as SDR images do not contain as much information as HDR ones. However, iTMOs manage to avoid some artifacts caused by the exposure fusion, such as ghosting (due to movement in the scene), motion blur or Moiré effect (usually presents in high exposure pictures). While at first, iTMOs were based on content-based assumptions and were using photographic rules to extend the dynamic range, neural networks are now used to this end.

Inspired by other computer vision domains (such as saliency [MNL13] or denoising [Ker14]), we devise the HDR Light Fusion Network (HDR-LFNet), a new iTMO aggregating several existing iTMOs that are not based on learning algorithms. Hopefully, the fusion of all methods performs better than each input method individually. Our approach uses a supervised neural network with several orders of magnitude fewer parameters to learn compared to existing networks, which is achieved thanks to some pre-processing. This pre-processing lowers as well the number of training images needed. This is a real improvement, as

our network needs HDR training images and HDR images are not as easy to gather as SDR images.

However, as we need fewer images to train, each image is increasingly important and needs to carry a lot of information to effectively train the network. We consider that high resolution images have a higher probability to contain the qualities that we need (light sources, dark areas, smooth gradient and high frequency areas). The HDR datasets that already exist usually contain a few hundreds images at most, with size around (1920×1080). To train our network, we then collected high resolution HDR images, and we compiled them in a new dataset.

We tested our method against state-of-the-art using four metrics on HDR images: HDR-VDP2, Harmonic HDR-IQA, PU-PSNR and PU-SSIM. Our method yields results similar to the state-of-the-art, but runs with fewer parameters. We also conducted a user experiment to compare HDR-LFNet to state-of-the-art methods. Results show that HDR-LFNet is preferred to others in the user study. Along with the short training time required by our network, this user study shows that our method is usable and effective in a wider range of applications.

Our contributions are:

1. devising a novel inverse tone mapping light architecture that merges several existing iTMOs to get a more powerful one;
2. proposing a new dataset of high definition HDR images composed of 496 pairs of middle exposure SDR and the corresponding HDR;
3. evaluating our work and other state-of-the-art methods using objective metrics and a user experiment.

The rest of this article is divided as follows. Section 2 presents the recent work about HDR and inverse tone mapping. We explain our approach in section 3 and then present the evaluation of our method in section 4. Finally, section 5 concludes the paper.

2. Related work

2.1. Hardware-based HDR generation

Intuitively, the best way to create HDR content is to devise specific devices. One method is to use other kinds of sensors along with classic CCD sensors to get more information. For example, Han et al. [HZD*20] propose a new method to fuse an SDR image with an intensity map provided by an event camera (also called neuromorphic camera) using deep learning.

Instead of adding new information (such as intensity maps) of the same scene, another method is to modify an existing camera to better reconstruct the HDR afterwards. Some articles [MIPW20; STF*20] combine the design of a new lens – which point spread function is thus known and optimized for HDR content – and a neural network to reconstruct HDR content from the image taken by this modified lens.

2.2. Software-based HDR generation

Multi-exposure fusion. Using only a classic camera, the most popular way to create HDR content is by fusing multiple images of the same scene with different exposure times. Several fusion algorithms exist [MKV08; DM08]. The fusion methods are known to yield HDR images of very good quality, but the process of taking the image is more difficult than other methods: both the camera and the scene must stand still for several seconds, the time to take several images with different exposure times. If the camera, or objects in the scene, move between the different shots, some ghosting artifacts will appear on the merged HDR image.

Single-exposure non-deep fusion. One way to reduce ghosting artifacts from HDR images is to use only one image and expand its dynamic range using iTMO. By doing so, HDR generation becomes an ill-posed problem as we do not have all the information needed to reconstruct faithfully the HDR image. In this situation we need to make some assumptions about the images we have. For example, we can assume that only the high luminance areas are lost. In that case, any pixel in the low or medium ranges would not be modified by the iTMO. Many algorithms were devised to expand the dynamic range from a single image. One idea is to use a non-linear function to modify the luminance of the image differently based on the pixel value of the SDR version. We can cite several methods, such as Akyuz [AFR*07], Kovaleski & Oliveira [KO14], or Landis [Lan02]. All of them differ from one another by the non-linear function they use to improve the dynamic range. The modification of the luminance value is solely based on the SDR pixel value, and so the context is not taken into account. Later works try to consider the semantics of each pixel to better improve the output quality using neural networks.

Deep neural networks for single image HDR reconstruction. The latest and more powerful algorithms for single image HDR reconstruction are based on deep-learning algorithms, and more specifically supervised learning methods. These algorithms allow for tuning several millions of parameters using big datasets of images as ground truth. The first widely recognized deep CNN for single image HDR reconstruction is HDRCNN [EKDM17]. It uses a really deep network of about 30 million parameters to enhance the brightest part of the SDR picture in input. The output of the network is then combined with an augmented version of the SDR (obtained with an average inverse camera response function estimated over a dataset) using a mask to use the network output in bright zones and the augmented SDR in the other areas. The network has been pre-trained on simulated HDR data (using a simple iTMO on a large image dataset) and fine-tuned with true HDR images. Other papers [SRK20; LLC*20] improve on HDRCNN by using inpainting-like tasks in the network – either as pre-trained weights or as another module. The idea is that such a network must reconstruct the over-exposed areas of the images, as this information is lost in the SDR image. Usually, HDR imaging focuses on high lights, and therefore the proposed methods work on improving the over-exposed areas. However, the same tricks can be used to improve the quality of under-exposed areas as well. The network of Marnerides et al. [MBD21] uses generative adversarial networks and inpainting to improve the dynamic range in both lowly and highly lit parts of the image.

In our approach, we use preprocessing to reduce significantly the number of parameters of our network, and postprocessing to correct

the network output as best as possible. This allows us to improve lowly and highly lit parts of images.

3. Our fusion network

Our goal is to expand the dynamic range of images. To achieve this goal, we use supervised learning algorithms to train a neural network. In this section we present the network architecture and the training process along with the training dataset. As we decided to use expanded version of images through iTMOs as input of our algorithm, we also present how we choose those operators. Our method is represented in Figure 1.

3.1. Overview of 3D convolutions

In this section we present some general characteristics of 3D convolutions, that we extensively use in our network.

An image can be represented as a tensor of size (H,W,C) with H the height, W the width and C the number of channels. A channel is a scalar array of size (H,W) containing information about a specific characteristic. For example, in traditional coloured images, $C = 3$ (each channel contain information about a specific colour component for example in *RGB* values), or if we only use gray level images, $C = 1$. Convolutional neural networks work by learning the weights of convolution filters. At each level (that we call layer) in the encoder section, the number of channel increases, to learn more and more structured information. It is not rare to see tensors with size $(H,W,64)$ in the middle of a CNN. Tensors which are not the input or the output tensors of a neural network are called feature maps.

A classic 2D convolution in a CNN layer is characterized by its kernel size, denoted by (k_x,k_y). A 2D convolution between a layer with C channels and another one with C' channels will have $k_x \times k_y \times C \times C'$ parameters to learn. All scalar values in a (k_x,k_y,C) voxel of the input tensor are multiplied term by term with the kernel weights, and then added together to yield a single scalar value. This process is repeated C' times with C' different sets of weights, to get C' values. Starting with a tensor of size (H,W,C), this convolution yields a tensor of size $(H-k_x+1,W-k_y+1,C')$.

In our case, we use 3D convolutions, which work on volumes of images of size (H,W,D,C). The convolution kernels also have one more dimension (k_x,k_y,k_z), but they are similar to 2D convolution: they have $k_x \times k_y \times k_z \times C \times C'$ parameters to learn, and starting with a tensor of size (H,W,D,C), this convolution yields a tensor of size $(H-k_x+1,W-k_y+1,D-k_z+1,C')$. Besides, as 2D convolutions, they consider all channels when computing the convolution sum. The advantage of 3D convolutions in our case is to specify the depth dimension: this allows for more control over the training of the network, and its behaviour.

2D convolutions and 3D convolutions are related: if we consider tensors with size (H,W,C) as $(H,W,1,C)$, 2D convolutions (k_x,k_y) are the same as 3D convolutions $(k_x,k_y,1)$. On the other hand, we can view tensors (H,W,D,C) as (H,W,DC) and convolutions (k_x,k_y,D) are the same as (k_x,k_y), but convolutions (k_x,k_y,k_z) with $k_z < D$ are not translatable to 2D convolutions, so 3D convolutions are strictly more expressive than 2D ones.

3.2. Architecture

As represented on Figure 1, our method uses a neural network at its core. The architecture of our network is represented on Figure 2. We adopt an encoder-decoder shape to reduce the size of the images during the forward pass, and thus reducing the time and memory needed for training. Many different characteristics of our network are explained in this section.

Our network only processes the luminance of images – the L^* component in the $L^*u^*v^*$ colour space –, and thus the input has one channel. The L^* channel is linear, so we need to apply a gamma correction to the network output, as well as adding back colour to yield a final HDR output. These postprocessing operations are presented in Section 3.4. The activation function is based on ReLU. As HDR images usually do not contain values of 0, we define a new activation function called Nonzero-Relu as

$$ReLU_a(x) = \begin{cases} a & \text{if } x \leq a \\ x & \text{else} \end{cases} \quad (1)$$

with a value of $a = 10^{-12}$.

While designing our architecture, we follow some common guidelines to avoid well known problems. To avoid artifacts usually caused by deconvolutions, we instead upsample the feature maps, and then apply a classic convolution in the decoder part. Besides, along with maxpooling to reduce the dimension of our network, we use dropout layers to stabilize the training and dodge local minima of the loss function.

Inspired by other neural networks [LLC*20; KSL*16], we decided to lighten our network - in order to improve its performances - by using an architecture specific to our problem. For this purpose we have provided the network with two new characteristics: (1) the input of our network are images expanded with existing iTMOs, (2) 3D convolution layers are used to force the network to learn the added value of each pair of expanded images. These two characteristics are detailed in this section.

By using already inverse tone-mapped images as input of the network, this latter must learn an easier transformation from HDR to HDR rather than from SDR to HDR. These images are concatenated on the depth dimension of the tensor. We manage to drastically reduce the number of parameters of our network. By using three expanded versions of the same image (obtained with three different iTMOs) as input, we are able to reduce the number of parameters to approximately 2×10^5, against the 10^6 to 10^8 parameters of state-of-the-art networks. The choice of iTMOs we use as input of our network is discussed in Section 3.6. As the number of trainable parameters is quite low compared to other networks, it should be harder to train using small crops of images as other methods tend to do. This assumption is verified in Section 4.2 by training our architecture with the HDR-Real dataset.

As we have several versions of the same image with only one channel, we can induce the network to learn how each iTMO interacts with the others. To this end, we use 3D convolutions instead of the classic 2D ones to focus the training.

To ensure that each pair of iTMO is considered, we need to input redundant information in our network. Our input tensor is

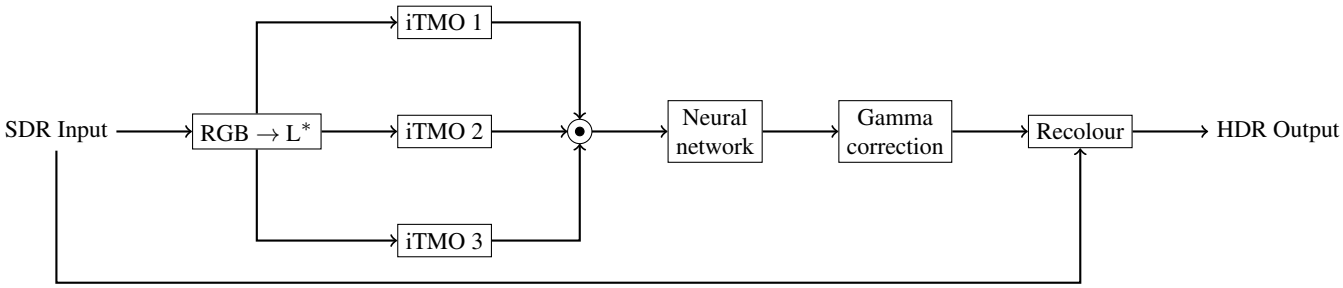


Figure 1: Our method HDR-LFNet uses a neural network to fuse several expanded versions of the input. This allows for faster training and lighter network. As we process linear luminance values, we use a gamma correction and colourization as post-processing.

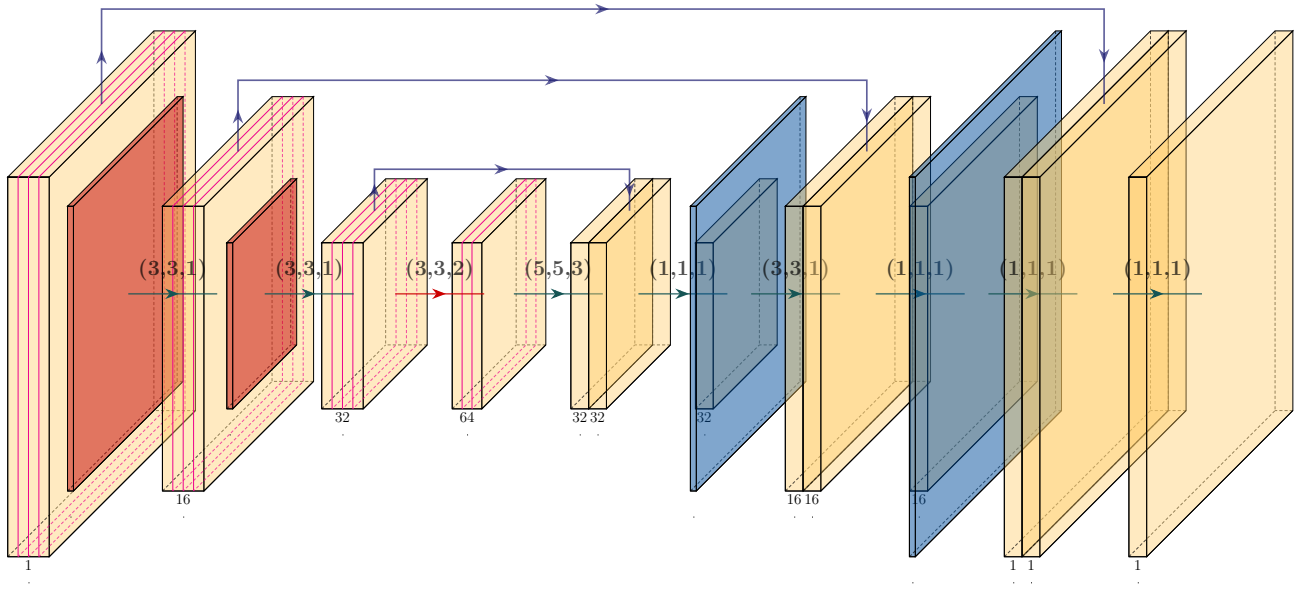


Figure 2: Architecture of our network, which uses 3D-convolutions. The network processes volumes of images (H,W,D,C) (represented in yellow), where depth D is represented by magenta lines. The number below each feature map corresponds to the number of channels C (not graphically represented on the figure). Green arrows represent 3D convolution of kernel size (k_x,k_y,k_z) followed by the activation function $ReLU_a$, blue arrows represent skip connections (which are detailed in Section 3.2). The red arrow indicates the 3D convolution that effectively fuse the input images. Downsampling and maxpooling are represented by red-orange layers, and upsampling are represented by blue layers.

the concatenation (I_1,I_2,I_3,I_1) in the depth dimension with I_i the expanded image generated by the i-th iTMO algorithm. This allows 3D-convolution with depth $k_z = 2$ to process all pairs (I_1,I_2); (I_1,I_3) and (I_2,I_3) as explained in Section 3.1. This convolution is done at the heart of the network (represented by the red arrow on Figure 2). After this convolution, we just need to upsample the feature maps to get back the original resolution. The final convolutions are done using maximum depth 3D convolutions to simulate 2D convolutions, as the depth dimension is not useful anymore. We compare this method to the classic 2D Convolution in the ablation study (Section 4.3). Therefore, the input tensor of the network is of size (Height, Width, Depth $= 4$, Channels $= 1$). These considerations determine the depths of all feature maps of the networks, and therefore the depth values k_z of all 3D convolutions. In the following, the depth values of 3D convolutions which are irrelevant (because they are already fixed by our previous construction choices) are denoted by a question mark ?.

Finally, we know that using convolution filters usually averages neighbour pixels, and therefore degrades the edges. First, to keep high frequencies as much as possible, we use skip connections. However, the feature maps before and after the fusion convolution have different depths, so we average the first feature maps over the depth dimension before merging them with the second ones. The aggregation of the feature maps is done by concatenating both feature maps, and then running a convolution with a kernel of $(1,1,?)$. Moreover, we modify the kernel sizes of the convolutions of the decoder part in a coarse-to-fine manner. Indeed, we decrease the kernel size from $(5,5,?)$ to $(1,1,?)$. This allows the final convolu-

tion to be a $(1,1,?)$ convolution, which better preserves the edges in the images. The impact of the kernel size decrease in the decoder part is discussed in the ablation study (Section 4.3).

3.3. Loss function

Our loss function is designed for comparing HDR images. We denote by $L(I_{HDR})$ the luminance of the ground truth HDR image, and by $\widehat{L}$ the output of the network. The loss function used for training consists of four parts: (i) a mean absolute error (MAE) $Y_c = \mathcal{L}_1(L(I_{HDR}), \widehat{L})$ for the actual values, (ii) a gradient-based error (gMAE) $Y_g = \mathcal{L}_1(g(L(I_{HDR})), g(\widehat{L}))$ – with g the gradient computation using Scharr filters – to emphasize the shapes, (iii) a perceptual loss Y_p to be more accurate on areas that are sensitive, and (iv) a dynamic range error Y_d.

The perceptual loss is based on VGG16 [SZ15]. The idea is to compute visible errors at different scales, and to that end we use activation maps from an already trained VGG network. We compute the mean absolute error between deep features of the target and the output images at the first four layers. The computation is done using a $\mathcal{L}_1$ difference.

Finally, we add a dynamic range error Y_d:

$$Y_d = \left| D(L(I_{HDR})) - D(\widehat{L}) \right| \text{ with } D(X) = \max(\log X) - \min(\log X)$$

This dynamic range error is to ensure the network produces an output with a dynamic range similar to that of the ground truth. The actual loss function is then a combination of those four components:

$$Y = \alpha Y_c + \beta Y_g + \delta Y_p + \varepsilon Y_d \tag{2}$$

Using the validation set, we found that the values $\alpha = 1$, $\beta = 0.3$, $\delta = 0.15$ and $\varepsilon = 1$ work best.

3.4. Post-processing

As our network only processes the luminance of images, we need some post-processing to at least add colour to the output image. Besides, as we input linear luminance images, we add a gamma correction to the output of the network to better match the ground truth image. We explain this process in this section.

In the following, we denote by I the coloured SDR input (and $L(I)$ its luminance); $\widehat{L}$ the output luminance of the network; and $\hat{I}$ the HDR output recoloured with our method. To recolor our HDR images, we use the luminance preserving formula proposed by Mantiuk et al. [MMTH09]:

$$\hat{I}(\gamma, s) = L_{exp}(\widehat{L}, \gamma) \left(\left(\frac{I}{L(I) + 10^{-5}} - 1 \right) \times s + 1 \right) \tag{3}$$

with γ the gamma factor, $s \in [0;1]$ the saturation factor and $L_{exp}(\widehat{L}, \gamma)$ the expected HDR luminance. Because this formula was designed to preserve luminance, we ensure that $L(\hat{I}(\gamma, s)) = L_{exp}(\widehat{L}, \gamma)$. We add the value 10^{-5} to avoid problems with luminances of zero. We correct the output luminance computed by the network using a gamma transformation $L_{exp}(\widehat{L}, \gamma) = \widehat{L}^{\gamma}$. The work of Mantiuk et al. also contains a method for automatic colour correction, however, their studies show that this method is not applicable to HDR images.

Dataset	Number of HDR images	Images size
HDR-Eye [NKHE15]	46	(1920×1080)
DEIMOS [KFP*11]	79	(4300×2900)
pfstools [MKMS07]	8	Variable
HDRPS [Fai07]	105	(4300×2900)
HDR-Real [LLC*20]	480*	Variable
Our dataset	496	$\sim (6000 \times 4000)$

Table 1: Characteristics of different HDR datasets. (*This is the number of original HDR images in the training set, but this dataset is composed of more than 19,000 crops of size (512×512))

The saturation factor s and the gamma factor γ can be modified to yield different results. To get the maximum of correlation between the ground truth and our modified output, we choose both factors by minimizing the mean square error between the images of our training dataset and the computed image from our network. As human beings are more sensitive to order of magnitude of luminance rather than absolute values, we compute those difference in the log-domain. Let T be the set of training images. This amounts to:

$$\gamma^* = \arg\min_{\gamma} \sum_{I \in T} \left\| \log_{10}(L(I_{HDR})) - \gamma \log_{10}(\widehat{L}) \right\|_2^2 \tag{4}$$

$$s^* = \arg\min_{s} \sum_{I \in T} \left\| I_{HDR} - \hat{I}(\gamma^*, s) \right\|_2^2 \tag{5}$$

After optimizing these formulas, we consider the parameters $\gamma^* = 2.659$ and $s^* = 1$.

3.5. Training dataset

Because we use supervised learning methods, we need an image dataset with HDR ground truth and SDR input images to train our network. We present our new training dataset in this section.

The training dataset is an essential component of a neural network. It is mandatory that we carefully select the right images with regards to the architecture and to our needs. The network we devise includes fewer parameters than the state-of-the-art networks. To train such a network, we need a sufficient number of images – which is less than other state-of-the-art networks –, but each image must contain as much information as possible. To do so, we need very high resolution images. These images may be found in currently available datasets, but they are not easily recoverable (see Table 1 for a comparison of the different existing HDR datasets). Therefore, we collect a new HDR image dataset.

This dataset is mandatory to train our network, but is an addition to other datasets: the same data augmentation techniques can be applied to yield several thousands of smaller images. As such, our dataset can be used for the same purposes as other datasets.

We have taken photographs with a Sony Alpha 7 III camera. We

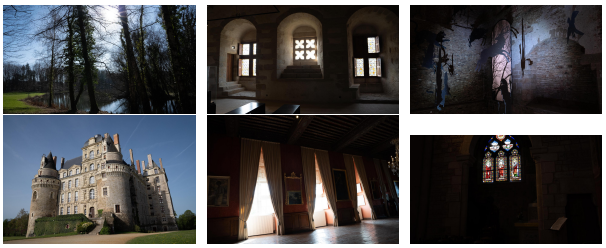


Figure 3: Examples of SDR images from the proposed dataset.

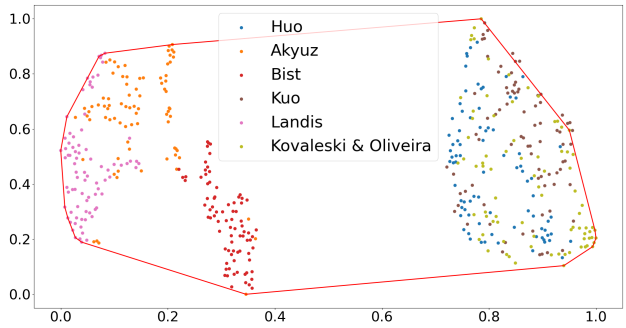


Figure 4: 2D projection of the scores of images obtained with the t-SNE algorithm. Each colour corresponds to a different iTMO. The red line is the maximum area convex hull, which encompasses points from Akyuz, Landis and Kovaleski & Oliveira iTMOs.

use the bracketting mode to take 3 exposures at -3, 0 and +3 exposure values. The ground truth HDR image is obtained through exposure fusion using Photoshop algorithm, as it provides images with less artifacts. For the SDR input, we choose the middle-exposed photograph as it contains information balanced between dark areas and bright areas. Besides, colours are usually more saturated in low light environments, and less saturated in high light environments. The middle-exposed shot provides the best colour quality among the three exposure photographs. Each element of our dataset is then composed of the fused HDR image and the middle-exposed image as the SDR image input. Some examples of images are shown on Figure 3. We have managed to take 496 HDR photographs. We have done no photometric calibration, so all the images are provided in relative luminance values. We then have to normalize all images to use them as training images: we divide all image RGB values by the maximal RGB value in the dataset. As all images were taken with the same camera, this ensures that every HDR image has values in $[0;1]$ while maintaining homogeneity.

For training our model, we split this dataset in three parts: 80 images for testing the model; 56 images for validation; and the remaining 360 images for training. The 360 training images are then flipped horizontally and vertically to yield an effective training database of 1,440 images. This dataset is available online[†].

For testing purposes and comparison with other methods, we use the HDR-Real test dataset proposed by Liu et al. [LLC*20]. It contains more than 8,000 pairs of SDR/HDR images and is widely used in the state-of-the-art as a test set thanks to its number of images. These images were obtained from 480 original HDR images using augmentation techniques, namely cropping (with a crop window of 512×512) and varying exposure times and CRFs to create SDR from HDR.

3.6. Choice of input iTMO

3.6.1. Method

As we want to use already inverse tone-mapped images as input of our network, we now need to select the iTMOs that we can use. We start from a set of inverse tone mapping operators: the five operators implemented in the Matlab HDR toolbox [BADC17] and the style-aware tone expansion [BCMD16]. Those iTMOs are not based on learning algorithms, but rather handcrafted using photographic rules and common assumptions. Setting up the input tensor so that each pair of images is processed by the 3D convolution is feasible with three operators (as shown in Section 3.2). Among the six available iTMOs, we want to select three operators that are quite different, such that we have maximum performances with minimal network input size. The selection method is presented in this section.

First, we suppose that all of our operators are of similar output quality (with different strengths and weaknesses), and we use quality metrics to differentiate the operators on a chosen set of N images. We use our training dataset to do this, so we have $N = 360$. We process each inverse tone-mapped images of our set with six different metrics: HDR-VDP2, FSIM, MCS5, SI, PU-PSNR, PU-SSIM. These metrics are the ones used by Harmonic HDR-IQA [RDLC19], and PU-PSNR and PU-SSIM [AMS08]. This yields $N \times 6$ points in a 6D space. Then, we project those points in a 2D space using the t-SNE visualization algorithm, to make the analysis easier. From this, we compute the coverage of the space of each set of 3 iTMOs among the 6. To do so, we compute the convex hull which encloses all points from 3 of the iTMOs, and we finally choose the 3 iTMOs which have a convex hull of maximum area. The t-SNE plot, along with the convex hull with largest area, is represented in Figure 4. The three chosen iTMOs are Akyuz [AFR*07], Landis [Lan02] and Kovaleski and Oliveira [KO14].

3.6.2. Discussion

All of these operators have different strengths and weaknesses. Akyuz operator is a simple algorithm that sets the maximum luminance value to a constant. While it usually burns the high luminance areas, low- and middle-exposed areas of the HDR output are quite faithful to the SDR version.

Landis operator uses a power function to improve the luminance values of pixels above a certain threshold. This yields high-exposed pixels with appropriate value, but may introduce artifacts: as the SDR image is quantized, some blocks or bands may appear on smooth gradient areas.

Finally, the operator from Kovaleski & Oliveira uses joint bilateral filters to smooth out the areas to expand. This method reduces

[†] ftp://ftp.irisa.fr/local/percept/public/hdrlfnet/

the pixel values, but produces HDR images with less artifacts than Landis.

Note that the t-SNE algorithm which projects high dimension data points in smaller spaces is not very stable: small variations in the input data could modify the projection, and thus the chosen operators. We can however see on the Figure 4 that several iTMOs are very close to each other. As the areas of the bounding boxes are not very different to each others, this choice should not have a huge impact on the performances of our model.

4. Results

4.1. Implementation details

The network is written using the PyTorch framework and is available online[‡]. Using the dataset presented in Section 3.5, we train the network for 15 epochs, while reducing the learning rate each time the validation error increases. The number of epoch is quite low compared to other networks due to the small size of the network. Due to memory restrictions, we use a batchsize of 1.

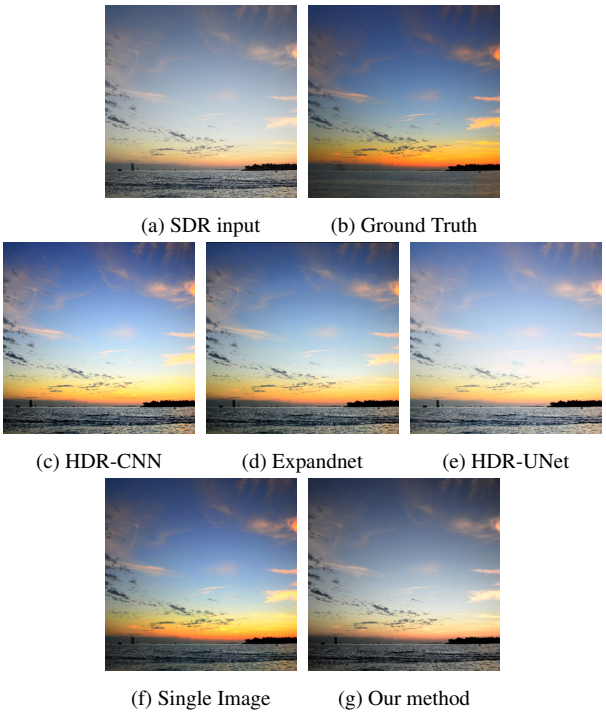
Besides, each epoch runs for about 40 minutes, for a total training time of approximately 10 hours. This is a much faster training than state-of-the-art training, which ranges from a few days to a full week.

4.2. State-of-the-art comparison

In this section, we compare our method to other state-of-the art ones using objective metrics. The iTMOs we consider are HDR-CNN [EKDM17], ExpandNet [MBHD18], the Single Image Network [LLC*20] and HDRUNet [CLZ*21]. We show some examples of images obtained with our method and with existing models on Figures 5 and 6.

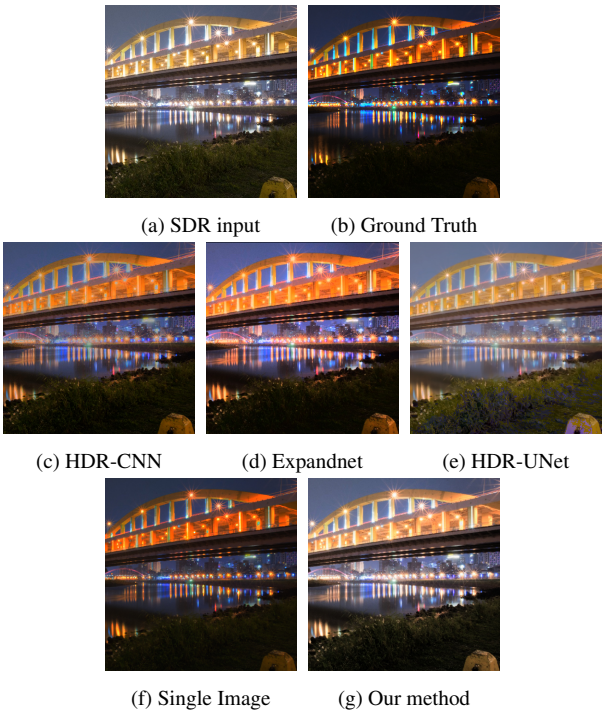
HDRCNN and the Single Image Network are presented in Section 2.2. We remind that HDRCNN contain about 30×10^6 trainable parameters, and the Single Image Network improves on HDR-CNN by using inpainting-like tasks during the training. It contains about 2×10^6 parameters. HDRUNet contains 1.6×10^6 parameters. The main idea behind HDRUNet is to split the network into three modules: a base network that performs most of the work, a condition network that computes spatially-variant transformations used to modify the deep features of the base network, and a weighting network that detects over-exposed areas to improve the reconstruction in those areas. All these networks are trained using their novel $tanh_{\mathcal{L}1}$ loss function. ExpandNet is a much lighter network with around 5×10^5 parameters. It also proposes a network with different modules, with each module working on a different scale: a global branch, a local branch and a dilation branch for mid-scale features. Each of those branches adds new information, which allows for a more faithful HDR reconstruction.

All of the networks are fully convolutional neural networks, meaning that the input can theoretically be of any size. However, for HDRCNN, due to how the deconvolutions are used, the input



(a) SDR input (b) Ground Truth (c) HDR-CNN (d) Expandnet (e) HDR-UNet (f) Single Image (g) Our method

Figure 5: Example of an image from HDR-Real dataset processed by different models. For ease of view, HDR images have been tone-mapped using the Drago algorithm [DMAC03].



(a) SDR input (b) Ground Truth (c) HDR-CNN (d) Expandnet (e) HDR-UNet (f) Single Image (g) Our method

Figure 6: Example of an image from HDR-Real dataset processed by different models. For ease of view, HDR images have been tone-mapped using the Drago algorithm [DMAC03].

[‡] ftp://ftp.irisa.fr/local/percept/public/hdrlfnet/

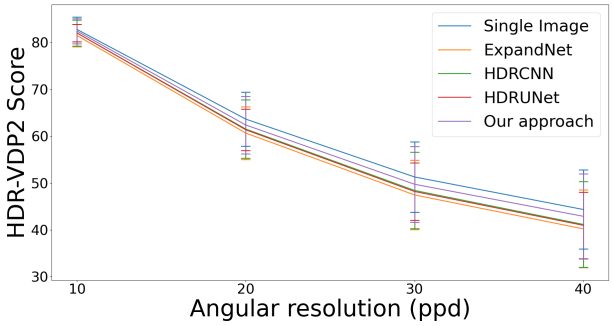


Figure 7: HDR-VDP2 score variation depending on the angular resolution on the HDR-Real dataset for four models.

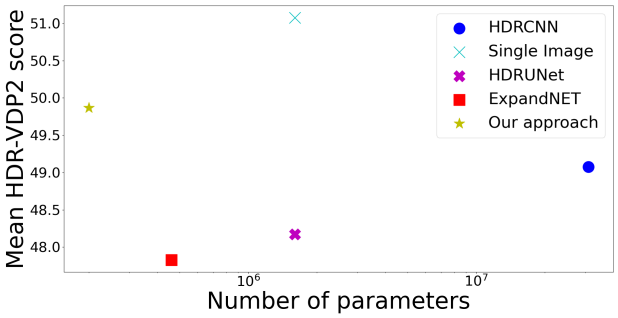


Figure 8: Mean HDR-VDP2 score against the number of trainable parameters of the tested models.

must have its height and width multiple of 32. This means that some of the images are either cropped or resized to fit this requirement.

The metric used in the state of the art for comparing models is HDR-VDP2 [MKRH11]. It is a metric with reference, and it takes as supplementary argument the angular resolution (measured in pixel per degree (ppd)). The angular resolution depends on the distance between the observer and the screen, and the resolution of the screen. This metric allows for a faithful simulation of the viewing experience on a specific screen. We represent on Figure 7 the impact of the angular resolution on the score. We notice that, although the actual scores change, the ranking of the model do not. For the rest of the evaluation, we use an angular resolution of 30 ppd.

As HDR-VDP2 only works on luminances, we use Harmonic HDR-IQA [RDLC19], which is sensitive to colours, as well as PU-PSNR and PU-SSIM [AMS08].

For this experiment, we test our network using the HDR-Real test dataset proposed by Liu et al. [LLC*20]. We train three versions of our network: (i) using the HDR-Real train, (ii) using our training dataset, and (iii) using our training dataset fine-tuned on HDR-Real train dataset. For fairness of comparison, we train these three versions for the same amount of time (10 hours). HDRCNN, HDRUNet and the Single Image Network are trained on the HDR-Real train set, while ExpandNet is used with the weights provided by the authors. We present the results of our evaluation in Table 2. We find out that, although our method do not perform the best, we manage to get second best on most of the metrics, except on Harmonic IQA. As Harmonic IQA assess the differences in colour between HDR images, this shows that our method does not reproduce colours as well as the other methods. However, the user study presented in Section 4.4 reveals that our method is preferred by observers. These results mean that although we are less faithful to the colours of the original image, our methods produces a more appealing picture than the state-of-the-art methods.

Note that our method performs best on HDR-Real test set when it is trained on our dataset, and not on HDR-Real train set. This comes from the nature of the datasets and our architecture: we managed to drastically reduce the number of trainable parameters. As explained in Section 3.5, the low number of parameters calls for fewer train images, but with richer content. HDR-Real images being only 512×512, our dataset is better suited to train our network than HDR-Real. We present on Figure 8 the mean HDR-VDP2 score as a function of the number of trainable parameters for our network, and networks of the state of the art. From this point of view, our model actually performs better than the state of the art.

4.3. Ablation study

In this section, we present the study conducted to assess the quality of the different components: the composed loss function and the usage of 3D convolutions, skip connections, and convolution kernel with varying sizes.

Each subsequent section presents a modified variant of our network. It is trained on our dataset and tested on HDR-Real. The average HDR-VDP2 score is presented on Table 3 for all variants, as well as for our proposed method (called final model in this section).

4.3.1. Composed loss function

Our loss function is composed of four different components. We train the same architecture with loss functions composed of only some of the original components detailed in Section 3.3. The mean absolute error (MAE) keeps the overall structure better among the four components, so we train networks with MAE and gradient loss (gMAE); MAE and dynamic range loss; and MAE and perceptual VGG loss. We also train a network with only MAE to compute the added value of each component. We use the same weighting as given in Equation (2) when training the different networks: for example, the version *MAE + gMAE* was trained using the loss function $Y = Y_c + 0.3Y_g$.

Surprisingly, our method performs better when trained with only MAE rather than MAE + gMAE or MAE + Perceptual loss. This can be explained as follows. In the final model, each weighting coefficient assigned to each component of the loss function has been carefully tuned with regards to the others. When using only two components of the loss function, the weighting coefficients should be different. Therefore, we can assume that the scores given in Table 3 are not optimal, except when training only with MAE (as no tuning is necessary because there is only one component). However, due to the large difference in score between the final model and the different loss function versions, we assume that our full loss

Model	Mean HDR-VDP2 (std) ↑	Mean Harmonic IQA (std) ↑	PU-PSNR (std) ↑	PU-SSIM (std) ↑
HDRCNN [EKDM17]	49.0745 (7.6484)	0.3554 (0.0997)	20.1330 (8.7622)	0.4325 (0.4499)
ExpandNet [MBHD18]	47.8268 (6.8777)	0.3685 (0.1003)	19.9602 (8.1942)	0.4043 (0.4643)
Single Image [LLC*20]	**51.0739** (6.9557)	0.3798 (0.1136)	**26.4531** (8.8608)	**0.5861** (0.4353)
HDRUNet [CLZ*21]	48.1709 (6.1163)	**0.4174** (0.0480)	18.0796 (7.3235)	0.3270 (0.4764)
Our Method trained on HDR-Real	28.3378 (5.8721)	0.2764 (0.1386)	12.4031 (6.4205)	0.0771 (0.3641)
Our Method fine-tuned on HDR-Real	41.3083 (8.7949)	0.3555 (0.1002)	19.7721 (8.9024)	0.3749 (0.4699)
Proposed Method	49.8686 (8.4689)	0.3597 (0.1053)	21.0415 (8.8968)	0.4647 (0.4407)

Table 2: Mean and standard deviation of HDR-VDP2 and Harmonic IQA for several models on the HDR-Real [LLC*20] test set. Scores in **bold** are the best; Scores underlined are the second best.

Model	Score
With 2DConv	42.6345
Without kernel size change	38.9088
Only MAE	36.3503
MAE + gMAE	34.9649
MAE + dynLoss	39.9745
MAE + VGGLoss	35.4325
Final model	49.8686

Table 3: Mean HDR-VDP2 score for different variants of the model.

function improves the output quality compared to the other tested loss functions. Besides, we notice that the dynamic loss is the most important component of the loss, as it effectively improves the quality according to the HDR-VDP2 scores. This is reflected in the relative weights of the loss: the weighting coefficient of the dynamic range loss component is much higher than the other weighting coefficients.

4.3.2. 2-by-2 processing

As explained in section 3.2, we use 3D convolution layers in our network. To assess the usefulness of the 3D convolution layers, we train the same architecture with 2D convolution layers only. The related HDR-VDP2 scores in Table 3 show that 3D convolutions contribute positively to the quality of the result. Visually, the output of the 2D Convolution network is similar to the output of our network, but we notice some discrepancies, especially in lowly lit areas. This may come from the fact that highly lit areas are represented by high values in the tensor, and those values overpower low light levels during the backpropagation. This effect is mitigated through the use of 3D Convolution, thanks to the specialization of the network.

4.3.3. Varying kernel size

To further improve the reconstruction of details, we use, in the second half of the network, convolutions with decreasing size from $(5,5,?)$ to $(1,1,?)$. We train the same architecture with fixed-sized convolution kernel of $(3,3,?)$. We notice some blur on those images, that comes from the convolution. Indeed, $(3,3,?)$ averages the values of the pixels in the neighbourhood, which leads to faded edges on the image, and to a worse HDR-VDP2 score.



Figure 9: Example of an image pair shown to one participant of the user study. Here, the left half was generated with HDRUNet and the right half with HDRCNN (all images have been tone-mapped using the Drago algorithm).

4.4. Subjective user study

We present in this section the user study we have conducted using our HDR SIM2 screen, to effectively compare the performance of our method with state-of-the art ones. We decided to perform a Two-Alternative Forced Choice (2AFC) experiment setting. To do so, we need to define the images pairs to be displayed on the HDR screen. For each participant, we randomly chose 10 images among our 80 test images and 5 versions of each of them: our method, HDRCNN, HDRUNet, ExpandNet and Single Image Network. The participants were presented with all possible image pairs created from the 10 test images and the 5 methods, for a total of 100 image pairs observed per session. In the following, we denote by $\langle I_M, I_{M'} \rangle$ or $\langle I_{M'}, I_M \rangle$ an image pair (with I_X half the image generated by the method X): one half is generated by the method M from the SDR image I_{SDR} and the other half by the method M' from the same SDR image I_{SDR} (see Figure 9). We then asked the participants to choose their preferred method among the two methods shown in the image pair displayed on the HDR screen. To respect the aspect ratio of the images, we randomly chose if we display the left sides or the right sides of each image in the pair. The displayed image pair is then composed of twice the same side of one image, as shown in Figure 9.

As it is difficult to train state-of-the-art networks with our dataset, we decided to use the pre-trained networks provided by the authors of the state-of-the-art methods. Therefore, our study

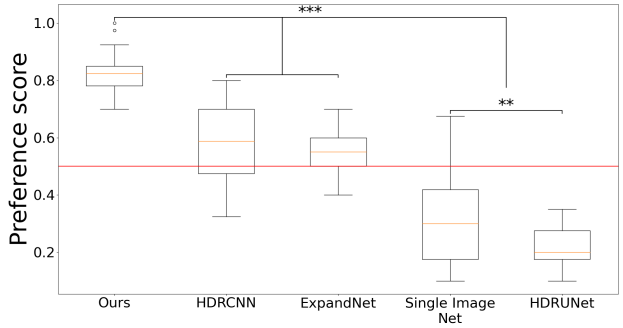


Figure 10: Preference of all participants for each method. The red line corresponds to a preference of 50%. The stars are attributed according to the p-values of the Tukey HSD test: nothing for $p > 0.05$, * for $0.05 \geq p > 0.005$, ** for $0.005 \geq p > 0.0005$, *** for $0.0005 \geq p$.

compares not only the architecture of these networks, but also their respective training datasets.

Each participant can attend multiple sessions (a session corresponds to the evaluation of 100 images pairs). If a participant attends another session, we use 10 images that the participant has not seen yet. The user study involved 29 participants (23M, 6F; age: avg=36.3 $\pm$ 12.7, min = 23, max = 71), and 30 sessions, for a total of 3,000 image pairs observed. Among the 29 participants, all of them reported to have normal or corrected-to-normal vision and 10 of them reported to have experienced HDR imaging before. During the experiment, each participant was asked to choose the method they preferred in the image pair. We have collected for each participant 100 answers (left or right of the image pair), each of them corresponds to a tested iTMO. For each participant p, we compute the preference scores $x_p(M_1),\ldots,x_p(M_5)$ for each method M_i. If we denote by $IP(M)$ the set of image pairs that contains an image generated with the method M, we can compute $x_p(M)$ with

$$x_p(M) = \frac{\text{card}(\{\langle I_M, I_{M'}\rangle \in IP(M) \text{ s.t. } p \text{ preferred } I_M \text{ over } I_{M'}\})}{\text{card}(IP(M))}.$$

The method M performs well according to the participant p if $x_p(M) \geq 0.5$. These preference scores are represented in Figure 10.

We can see on the Figure 10 that our method is largely preferred on average to all other tested methods. To further study these preferences, we performed a one-way ANOVA test after asserting that our data (the computed $x_p(M)$) come from a normal distribution (using a Shapiro-Wilk test). We obtain a p-value $p \ll 0.05$, meaning that the average values are significantly different. To further discriminate the methods, we perform a post-hoc test using a Tukey HSD test. This statistical test allows to compare the mean of every group (in our case, the groups are the different iTMOs) two-by-two. The results of the Tukey HSD test provides, for each pair of methods, the probability p that the mean preference scores of the two considered methods are the same. We present the results of the Tukey HSD test in Figure 10 by grouping together the methods with close probabilities p.

Using the post hoc test, we notice that the participants found on average no significant differences between the images processed by HDRCNN and by ExpandNet (Average value of preference score of $\bar{x} = 0.57$). Using the same test, our method is preferred on average to all other methods (Average value of preference score of $\bar{x} = 0.83$). Our method, HDRCNN and ExpandNet all have an average preference score of above 0.5, meaning that those three methods are most of the time preferred by the participants. This study shows that our method performs well for human observers on our test dataset.

5. Conclusion

We have presented a new inverse tone mapping operator, called HDR-LFNet, along with its training dataset. Our architecture is lighter than the networks of the state-of-the-art thanks to methods aggregation, a technique inspired by other computer vision domains. To be fully effective, this lightweight architecture requires high resolution images to train. As there is no existing training HDR dataset with sufficient resolution, we also release a new dataset of high resolution HDR images that can be used by the community in complement or in place of existing datasets.

Objectives metrics show that our method is on-par with other methods, but the conducted user study shows that our method is preferred by observers. Along with the lower number of parameters, our method is a real improvement over the state-of-the-art. Our HDR-LFNet can be used in several applications, where resources for training and storage of the model are limited. This also should allow to transform an SDR image dataset with annotations (quality score, aesthetics score, saliency data) to an HDR image dataset with corrected annotations.

References

[AFR*07] AKYUZ, AHMET OGUZ, FLEMING, ROLAND, RIECKE, BERNHARD E, et al. "Do HDR displays support LDR content? A psychophysical evaluation". *ACM Transactions on Graphics (TOG)* 26.3 (2007), 38–es.

[AMS08] AYDIN, TUNÇ OZAN, MANTIUK, RAFAL, and SEIDEL, HANS-PETER. "Extending Quality Metrics to Full Dynamic Range Images". *Human Vision and Electronic Imaging XIII*. Proceedings of SPIE. San Jose, USA, Jan. 2008, 6806–10.

[BADC17] BANTERLE, FRANCESCO, ARTUSI, ALESSANDRO, DEBATTISTA, KURT, and CHALMERS, ALAN. *Advanced High Dynamic Range Imaging (2nd Edition)*. Natick, MA, USA: AK Peters (CRC Press), July 2017. ISBN: 9781498706940.

[BCMD16] BIST, CAMBODGE, COZOT, RÉMI, MADEC, GÉRARD, and DUCLOUX, XAVIER. "Style Aware Tone Expansion for HDR Displays." *Graphics Interface*. 2016, 57–63.

[CLZ*21] CHEN, XIANGYU, LIU, YIHAO, ZHANG, ZHENGWEN, et al. "HDRUnet: Single image hdr reconstruction with denoising and dequantization". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, 354–363.

[DM08] DEBEVEC, PAUL E and MALIK, JITENDRA. "Recovering high dynamic range radiance maps from photographs". *ACM SIGGRAPH 2008 classes*. 2008, 1–10.

[DMAC03] DRAGO, FRÉDÉRIC, MYSZKOWSKI, KAROL, ANNEN, THOMAS, and CHIBA, NORISHIGE. "Adaptive logarithmic mapping for displaying high contrast scenes". *Computer graphics forum*. Vol. 22. 3. Wiley Online Library. 2003, 419–426.

[EKDM17] EILERTSEN, GABRIEL, KRONANDER, JOEL, DENES, GYORGY, and MANTIUK Rafaand Unger, JONAS. "HDR image reconstruction from a single exposure using deep CNNs". *ACM Transactions on Graphics (TOG)* 36.6 (2017).

[Fai07] FAIRCHILD, MARK D. "The HDR photographic survey". *Color and imaging conference*. Vol. 2007. 1. Society for Imaging Science and Technology. 2007, 233–238.

[HZD*20] HAN, JIN, ZHOU, CHU, DUAN, PEIQI, et al. "Neuromorphic camera guided high dynamic range imaging". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 1730–1739.

[Ker14] KERVRANN, CHARLES. "PEWA: Patch-based exponentially weighted aggregation for image denoising". *Advances in Neural Information Processing Systems* 27 (2014), 2150–2158.

[KFP*11] KLMA, MILO, FLIEGEL, KAREL, PÁTA, PETR, et al. "DEIMOS–An Open Source Image Database." *Radioengineering* 20.4 (2011).

[KO14] KOVALESKI, RAFAEL P and OLIVEIRA, MANUEL M. "High-quality reverse tone mapping for a wide range of exposures". *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE. 2014, 49–56.

[KSL*16] KONG, SHU, SHEN, XIAOHUI, LIN, ZHE, et al. "Photo aesthetics ranking network with attributes and content adaptation". *European Conference on Computer Vision*. Springer. 2016, 662–679.

[Lan02] LANDIS, HAYDEN. "Production-ready global illumination". *Siggraph course notes* 16.2002 (2002), 11.

[LLC*20] LIU, YU-LUN, LAI, WEI-SHENG, CHEN, YU-SHENG, et al. "Single-Image HDR Reconstruction by Learning to Reverse the Camera Pipeline". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 1651–1660.

[MBD21] MARNERIDES, DEMETRIS, BASHFORD-ROGERS, THOMAS, and DEBATTISTA, KURT. "Deep HDR Hallucination for Inverse Tone Mapping". *Sensors* 21.12 (2021), 4032.

[MBHD18] MARNERIDES, DEMETRIS, BASHFORD-ROGERS, THOMAS, HATCHETT, JONATHAN, and DEBATTISTA, KURT. "Expandnet: A deep convolutional neural network for high dynamic range expansion from low dynamic range content". *Computer Graphics Forum*. Vol. 37. 2. Wiley Online Library. 2018, 37–49.

[MIPW20] METZLER, CHRISTOPHER A., IKOMA, HAYATO, PENG, YIFAN, and WETZSTEIN, GORDON. "Deep Optics for Single-shot High-dynamic-range Imaging". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 1375–1385.

[MKMS07] MANTIUK, RAFAL, KRAWCZYK, GRZEGORZ, MANTIUK, RADOSLAW, and SEIDEL, HANS-PETER. "High-dynamic range imaging pipeline: perception-motivated representation of visual content". *Human Vision and Electronic Imaging XII*. Vol. 6492. International Society for Optics and Photonics. 2007, 649212.

[MKRH11] MANTIUK, RAFAL, KIM, KIL JOONG, REMPEL, ALLAN G, and HEIDRICH, WOLFGANG. "HDR-VDP-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions". *ACM Transactions on graphics (TOG)* 30.4 (2011), 1–14.

[MKV08] MERTENS, TOM, KAUTZ, JAN, and VAN REETH, FRANK. "Exposure Fusion: A Simple and Practical Alternative to High Dynamic Range Photography". *Computer Graphics Forum* 28 (Sept. 2008), 161–171. DOI: 10.1111/j.1467-8659.2008.01171.x.

[MMTH09] MANTIUK, RADOSLAW, MANTIUK, RAFAL, TOMASZEWSKA, ANNA, and HEIDRICH, WOLFGANG. "Color correction for tone mapping". *Computer Graphics Forum*. Vol. 28. 2. Wiley Online Library. 2009, 193–202.

[MNL13] MAI, LONG, NIU, YUZHEN, and LIU, FENG. "Saliency aggregation: A data-driven approach". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, 1131–1138.

[NKHE15] NEMOTO, HIROMI, KORSHUNOV, PAVEL, HANHART, PHILIPPE, and EBRAHIMI, TOURADJ. "Visual attention in LDR and HDR images". *9th International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*. CONF. 2015.

[RDLC19] ROUSSELOT, MAXIME, DUCLOUX, XAVIER, LE MEUR, OLIVIER, and COZOT, RÉMI. "Quality metric aggregation for HDR/WCG images". *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2019, 3786–3790.

[SRK20] SANTOS, MARCEL SANTANA, REN, TSANG ING, and KALANTARI, NIMA KHADEMI. "Single Image HDR Reconstruction Using a CNN with Masked Features and Perceptual Loss". *ACM Transactions on graphics*. Vol. 39. 4. July 2020.

[STF*20] SUN, QILIN, TSENG, ETHAN, FU, QIANG, et al. "Learning Rank-1 Diffractive Optics for Single-shot High Dynamic Range Imaging". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, 1386–1396.

[SZ15] SIMONYAN, KAREN and ZISSERMAN, ANDREW. "Very Deep Convolutional Networks for Large-Scale Image Recognition". *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by BENGIO, YOSHUA and LECUN, YANN. 2015.