CNN Based Inverse Tone-Mapping

Lynx L, Zixiao Z, Qinxin Lin, and Nishtha C.

***Abstract*—In the last several decades, High Dynamic Range (HDR) imaging has revolutionized the field of computer graphics and other areas such as photography, virtual reality, visual effects, and the video game industry. High Dynamic Range displays have seen technological advances and reached consumer levels. Hence there has been increasing commercial availability of HDR devices and HDR content. However, there is an enormous amount of legacy content to be displayed on HDR devices. To solve this problem, we propose an inverse tone mapping method using a Convolutional Neural Network with LDR based learning. It is difficult to train a CNN directly with HDR images because the loss function for HDR learning is problematic. The CNN used in the proposed method learns a transformation from various input LDR images to LDR images mapped by Reinhard's global operator.**

Keywords*—*HDR, SDR, inverse tone mapping, Neural networks, Deep Learning

I. INTRODUCTION

The original legacy televisions used a technology referred to as the Standard Dynamic range (SDR) which was only able to reproduce a limited amount of brightness and color perceivable by the human eye. The SDR video captures images, rendering and video using a conventional gamma curve, and therefore presenting a dynamic range that is considered standard, which allows for a maximum luminance of 100 cd/m2. HDR in the video industry can not only capture but also replicate a range of luminance values that a close to what the typical human eye can see in the real world.

Compared to conventional standard dynamic range (SDR) content, the visual quality of HDR content is much higher because it can provide life-like experience. For SDR, most of the color images are represented with 8 bits per pixel for each of the red, green and blue channels. Such a representation scheme can only make 256 different shades for each of the red, green and blue channel, which is less than what human eyes can perceive and inadequate to represent many scenes in the real world. Contrary to SDR, HDR can capture and store the luminance and color information of real-life scenes, both very dark and bright objects can be represented in the same image.

Most cameras today have HDR mode and one way to obtain HDR is to take multiple SDR images of different exposure over a short period of time and then synthesizing the photos. Even the slightest movement of subject or shaking of the camera can introduce blurriness and ghost artifacts [1]. A single shot HDR can be obtained using advanced HDR cameras which are expensive and will take time to reach consumer levels. This is where inverse tone mapping finds its application to convert a single SDR image to HDR image.

A plethora of research work exists to generate an HDR image from a single SDR image [2-6]. Traditional ways of inverse tone mapping expanded the dynamic range of an image by using a fixed function or parameterized function [2-3]. Akyuz et al. conducted a series of experiments using the linear operation to provide a decent SDR to HDR expansion [2]. Banterle uses the inverse of Reinhard to handle the saturated bright region of the image since the differences between HDR and LDR images are evident in bright regions [3].

The recent advances in machine learning have led to its widespread application in a variety of domains including image processing. Deep learning approaches are the current standards in solving image processing problems because they provide data-driven solutions instead of human reliance on heuristics. Latest works have demonstrated the efficiency of CNN in solving image processing problems due to their learning qualities and suitable architecture. Eilertsen et al. [7] aim to reconstruct saturated areas in input LDR images via a CNN using a new loss function calculated in the logarithmic domain, but their method is applicable only when a CRF used at the time of photographing is given. Endo et al. [6] have proposed a CNN based method that produces a set of differently exposed images from a single SDR image, to avoid the difficulty of designing loss functions for HDR images.

Thus, in this paper, we implement an inverse tone mapping method using a CNN with SDR based learning. Instead of learning a map from LDR images to SDR ones, the CNN used in the proposed method learns a transformation from various input SDR images to SDR images mapped by Reinhard's global operator [8]. The inverse tone mapping is done by applying the inverse transform of Reinhard's global operator to SDR images produced by CNNs.

The rest of this report is structured as follows: Section II discusses the methodology and proposed an approach that was used in this project. Section III has details about the CNN network, Section IV lists training and prediction technical implementation and Section V

reviews a final discussion on what future improvements can be carried out on the project.

II. METHODOLOGY

A. *Overview*

Here is an overview of the training approach and prediction approach.

Training

Step 1 - Input SDR image x from HDR image E by using a virtual camera with various CRFs.

Step 2 - Generate target SDR image y from HDR image by using Reinhard's global operator with parameter a = 0:18

Step 3 - Train a CNN to transform input LDR image x into target LDR image y i.e. input x SDR and output y SDR.

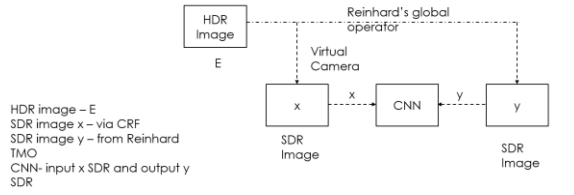


Figure 1: Training Model

Prediction

Step 1: x is SDR image taken with a digital camera.

Step 2: Transform x into Y by using the trained CNN. This transformation aims to estimate an LDR image generated by Reinhard's global operator

Step 3: Generate an HDR image Y' from Y by using inverse transform of reinhard using parameter a = 0.18

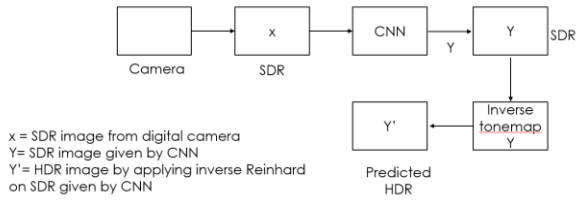


Figure 2: Prediction Model

B. *Dataset*

Many datasets are needed to develop an adaptive ITMO via deep learning. We use a HDR dataset from UBC comprising of 779 images with various scenes to be our training set. The scenes are classified as "Landscape", "People", "Building", "Animals" and "Objects". We use another HDR dataset from SFU comprising of 50 images to be test set. The scenes of test set are classified as "Nature", "Outdoor", "Indoor" and "People".

C. *Image Cropping*

The input of CNN is an image with a size of $512 * 512$ pixels, so we need to crop original HDR images in the training set and test set to image patches with $512 * 512$ pixels. For the training set, one image is cropped to four small patches. The position of the patch is determined randomly. The upper left corner coordinates of four image patches are (200, 300), (1000, 100), (512, 512) and (704, 284), respectively. After cropping, we have $4 * 779 = 3116$ HDR images. We select 2000 images as training images and 400 images as validation images. For the test set, one image is cropped to one small patch with $512 * 512$ pixels. The position of the patch is the center area of the image and the upper left corner coordinate of patch is (814, 454). In total, we have 50 HDR images with $512 * 512$ pixels as our test images.

D. *Virtual Camera*

Virtual camera model [7] can capture a number of random regions of the scene using a randomly selected camera calibration. The camera calibration incorporates parameters for exposure, camera curve, white balance and noise level [7]. In this paper, we generate SDR images from HDR images by using a virtual camera with various CRFs to be input images of CNN. This is corresponding to assuming that input SDR images are captured with various cameras [8]. This operation consists of the following steps:

a. Extract the world luminance component E of an HDR image.

b. Compute geometric mean $G(E)$ of luminance E by equation expressed in paper [8] we implement

$$G(E) = exp(\frac{1}{|P|} \sum_{(i,j) \in P} log(max(E_{i,j}, \epsilon)))$$

where P is the set of all the pixels and ϵ is a very small value that is close to 0 to avoid singularities at pixels $E_{i,j} = 0$.

c. Compute shutter speed Δt

$$\Delta t = a * 2^{v} / G(E)$$

where $a = 0.18$ is the parameter called "key value", which indicates subjectively if the scene is light, normal or dark. v is an uniform random number in range $[-4,4]$.

d. Compute exposure X from luminance E

$$X = \Delta t * E$$

e. Compute display luminance of an SDR image from exposure X by a virtual camera f. We use a parametric function in the form of a sigmoid to approximate different camera curves:

$$f(X) = min((1 + \eta) \frac{X^{\gamma}}{X^{\gamma} + \eta}, 1)$$

where η and γ are random numbers that following normal distributions. η has a mean 0.6 and variance 0.1, γ has a mean 0.9 and variance 0.1.

f. Determine the RGB values in SDR domain.

The following figure shows the example of SDR images generated by virtual camera model with different CRFs.

Figure 3. SDR images generated by virtual camera

*E. **Reinhard's Global Operator***

Reinhard's global operator described in paper "Photographic Tone Reproduction" [9] is one of typical tone mapping operator. It presents a good trade-off between simplicity and flexibility for the expansion curve [3]. In this method, we generate target SDR images from HDR images by using Reinhard tone mapping to be ground truth of CNN. This tone mapping operation consists of the following steps:

a. Extract the world luminance component E of an HDR image.
b. Compute geometric mean $G(E)$ of E by using the same geometric mean equation in virtual camera. Note, this equation is different from geometric mean equation in paper "Photographic Tone Reproduction" [9].
c. Compute scaled luminance X

$$X = \frac{a}{G(E)} E$$

where $a = 0.18$ determines the brightness of an output SDR image.
d. Compute display luminance I by Reinhard tone mapping function

$$I = \frac{X}{1 + X}$$

e. Determine the RGB values in SDR domain.

The following figure shows the example of SDR images generated by Reinhard's global operator.

Figure 4. SDR images by Reinhard's global operator

*F. **Inverse Reinhard's Global Operator***

Typically, global TMOs have an easy inverse since they are invertible functions, while local TMOs do not [3]. Here we use inverse transform of Reinhard's global operator to reconstruct HDR images from SDR images generated by CNN. The SDR images generated CNN is an estimation of SDR images by Reinhard's global operator. The inverse of Reinhard tone mapping has the following steps:

a. Inverse gamma. By inverting a gamma curve of 2.2, we can obtain pixel values that are approximately proportional to the luminance in the original scene.
b. Extract the world luminance component E of an SDR image.
c. Normalize luminance E

$$L = \frac{E - min(E)}{max(E) - min(E)}$$

d. Expand luminance of SDR image by inverse function of Reinhard tone mapping.

$$L_{hdr} = \frac{L_{max} L_{white}^2}{2a} (L - 1 + \sqrt{(1 - L)^2 + \frac{4L}{L_{white}^2}})$$

where $a = 0.18$ is the same parameter as Reinhard's global operator, $L_{white} = 10$ is the smallest luminance value that will map to white, $L_{max} = 1000$ is the maximum luminance value expected for HDR image and it is chosen by user.
e. Determine RGB values in HDR domain.

III. NETWORK

The network architecture of the CNN used in our method is designed based on U-Net [10]. Figure 5 shows the details of the CNN. The input of the CNN is 24-bit SDR image in RGB color space. The size of the input is 512*512 pixels.

In the encoder which is the left part of the network, we applied convolutional blocks followed by a max pooling layer to encode the data into a feature representation.

Each convolutional block in the encoder consists of two convolutional layers with the same filter number. The numbers of the filter in each block are set to 32, 64, 128, 256, 512 and all filters are in the same size of 3×3. In the max-pooling layers, we chose the kernel size of 2×2 with a stride of 2 for downsampling.

The decoder, which is the right part of the network, consists of upsample and concatenation followed by regular convolution operations. It is connected with the encoder through the skip connection. The skip connection brings in the details of initial layers to boost the image reconstruction.

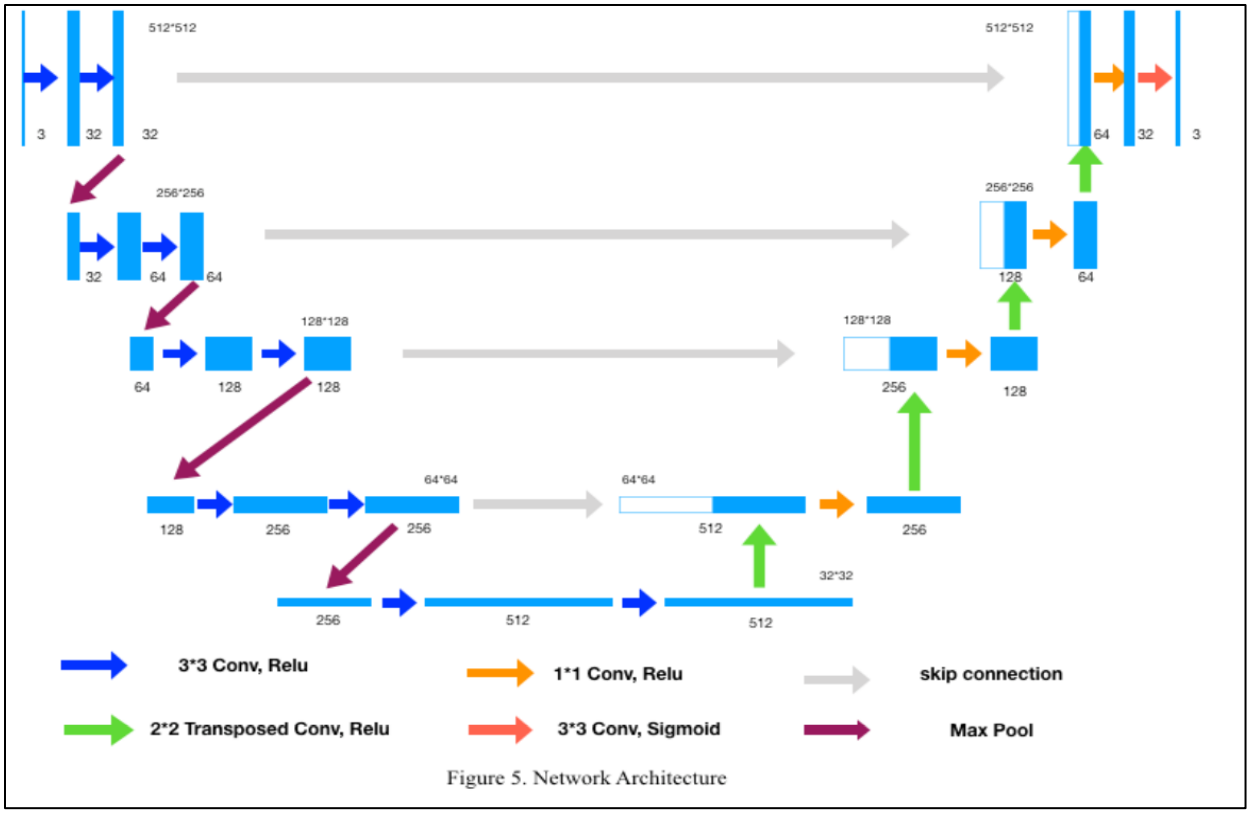


Figure 5. Network Architecture

We applied transposed convolution here to upsample the feature dimensions to meet the same size with the corresponding concatenation blocks from the encoder. In our method, the filters used in the transposed convolution layer are in the size of 2*2. From the first transposed convolution layer to the last one, the numbers of filter are 256, 128, 64 and 32. And 1×1 filter are applied in the convolutional layers except for the final one. To produce the output, we used three 3×3 filters in the final layer.

The rectified linear unit (ReLU) activation function [11] is applied in all convolutional layers and transposed layers except the final layer. For the final layer we used a Sigmoid function. To accelerate the training speed and improve the accuracy, we applied batch normalization [12] to the outputs of the ReLU functions.

IV. TRAINING AND PREDICTION

A. *Image Input*

During the training, image augmentation is used to allow for better generalization of the model. The augmentations used are as follows:

1) Image flips (both vertical and horizontal)
2) Image Shuffling (For each epoch)
3) Image Rotation (90-degree units)
4) Zoom Range (Up to ± 0.2 scale).

In addition to this, a smaller batch size of 2 is used to reduce overfitting by increasing weight update frequency.

B. *Training Results*

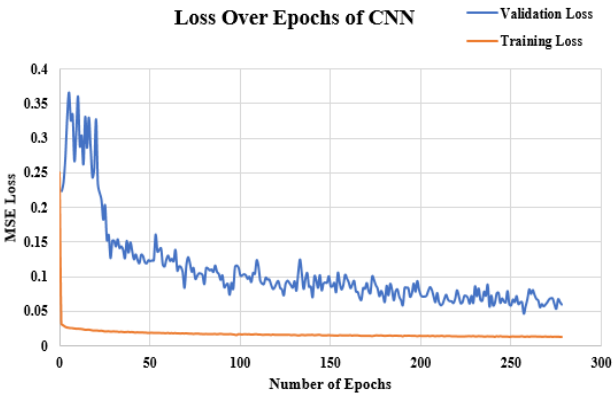


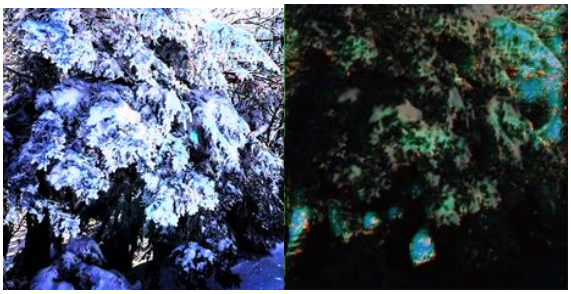
Figure 6 - Loss Over Epochs of CNN

The overall training result of the CNN showed a consistent improvement of the validation set up until approximately 300 epochs. From 300 epochs onwards, diminishing return is reached and eventually the model starts to overfit and the MSE loss climbs back up to 0.07 for the validation error. However, the training loss was able to reach 0.01 MSE at around 300 epochs. Thus, this potentially indicates overfitting of the model or the

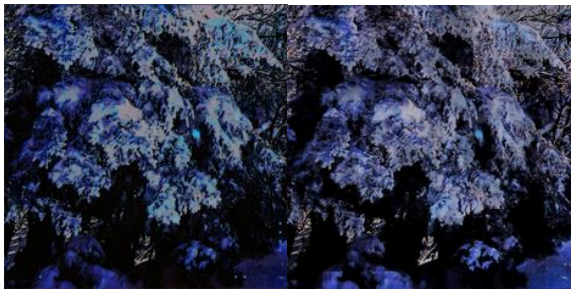
incapability to generalize well enough to the image set provided.

C. Image Prediction Results

Using the CNN, 400 validation images are sent through the model to create prediction images. Such images are generated per 10 epochs to observe the change in quality over the training.



SDR Virtual Camera 10 Epochs



100 Epochs 300 Epochs

Ground Truth

Figure 7: Prediction over Epochs

As can been seen in Figure 7 above, the model is able to learn to recreate missing details from the original validation SDR image overtime. Most of the artifacts disappears with the increasing epoch size as can be seen between 100 and 300 epochs. However, upon reaching 300 epochs where diminishing return is reached, there is still a slight discrepancy from the original ground truth in the luminance of each pixel.

The preliminary results were tested on two identical Sony BVM-X300 displays by viewing original HDR on one display and generated HDR on the other display. The results looked promising. As can be seen below in Figure 8, the CNN model was able to restore lost details in the clouds. In Figure 9, the stone wall saw a more natural coloring in addition to the restored details due to the CNN prediction. However, the model has not yet reached the optimal validation MSE as can be seen in the artifacts that exists in the image output. Thus, subjective testing has not yet been conducted at this point in time as we wait for an improved model to fix the artifacts.

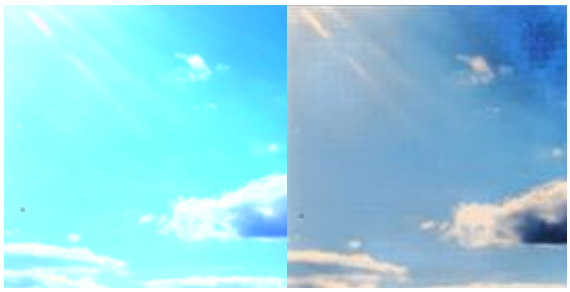


Figure 8: Blue Sky SDR to HDR

Figure 9: Stone Wall and Door Frame SDR to HDR

V. FUTURE IMPROVEMENTS

It can be noticed that many of the produced images have unwanted artifacts caused by the model not generalizing well enough. However, there are also concerning artifacts such as dotted patterns that indicates neuron outputs of 0 as can be seen in Figure 11 below. This can indicate potential dead neurons that is an inherent problem in ReLU layers implementation in a network architecture. Given the limited dataset size, it is likely that some images retain these 0 outputs no matter the number of epochs.

Figure 10 - City Landscape CNN Output

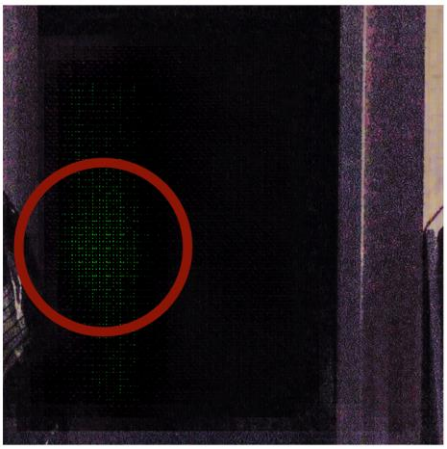


Figure 11 - Dark Doorway CNN Output

In order to solve this issue, further investigation would be needed in testing the use of "Leaky" ReLUs or ELUs to see if these dotted artifacts diminish. In addition, a larger dataset may be needed to help with better generalization of the model and combat the large error difference between the validation and training error.

Reference

[1] Jang, Hanbyol & Bang, Kihun & Jang, Jinseong & Hwang, Dosik. (2018). Inverse Tone Mapping Operator Using Sequential Deep Neural Networks Based on the Human Visual System. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2870295.

[2] A. O. Akyuz, R. Fleming, B. E. Riecke, E. Reinhard and H. H. ¨ Blthoff, " Do HDR Displays Support LDR Content? A Psychophysical Evaluation," ACM Trans. Graph (SIGGRAPH), vol. 26, no. 3, 2007.

[3] F. Banterle, P. Ledda, K. Debattista and A. Chalmer, "Expanding Low Dynamic Range Videos for High Dynamic Range Applications", *Proceedings of the 24th Spring Conference on Computer Graphics*, pp. 33-41. 2008.

[4] Y. Kinoshita, S. Shiota, and H. Kiya, "Fast inverse tone mapping with reinhard's global operator," in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017, pp. 1972–1976.

[5] Y. Kinoshita, S. Shiota and H. Kiya, ""Fast inverse tone mapping based on reinhard's global operator with estimated parameters," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 100, no. 11, pp. 2248–2255, 2017.

[6] Y. Endo, Y. Kanamori, and J. Mitani, "Deep reverse tone mapping," ACM Transactions on Graphics (Proc. of SIGGRAPH ASIA 2017), vol. 36, no. 6, pp. 177:1–177:10, Nov. 2017.

[7] G. Eilertsen, J. Kronander, G. Denes, R. Mantiuk, and J. Unger, "HDR image reconstruction from a single exposure using deep CNNs," ACM Transactions on Graphics (TOG), vol. 36, no. 6, pp. 178:1–178:15, 2017.

[8] Y. Kinoshita, H. Kiya, "Deep Inverse Tone Mapping Using LDR Based Learning for Estimating HDR Images with Absolute Luminance", arXiv: 1903.01277 [eess.IV], Feb. 2019.

[9] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic Tone Mapping Reproduction for Digital Images", *ACM Transactions on Graphics (TOG),* vol.21, no.3, pp. 267-276, 2002.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", arXiv, 2015.

[11] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp. 315–323, 2011.

[12] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv, 2015.