International Journal of Engineering Research and Advanced Technology (IJERAT)

DOI: 10.31695/IJERAT.2023.9.5.1

E-ISSN : 2454-6135

Volume.9, No. 5
May - 2023

Using Deep Learning Neural Networks to Enhance HDR Images

Ahmed Hasan khanjar

Computer Science Faculty of Basic Education,

Mustansiriya University

Baghdad, Iraq

ABSTRACT

In recent days, photography becomes main in our daily life style. Everyone like to upload images on facebook, instagram, or any social media site, or maybe want to photograph some personal images or for memory.

In other hand, the development of technology, especially in cell phones, let these mobiles equipped with cameras with a specific resolution. We define cameras with two parameters. First one is the width and the high of the image which can define the number of pixels inside the image, the bigger these pixels can be the accuracy and true scene can get. The second in the resolution of the pixel which means how many bits are used to define the colors of the image.

All natural images have a wide dynamic range because of containing a large range of intensity values. This high range cannot be collected using non-HDR cameras. So, mobile cameras might give very bad resolutions.

This search aims to use conventional neural networks to create an HDR images using a programmed manner so we can solve the problem of poor cameras problems, first of all we will create a map for an image using a dataset of static images has a different light intensity for the same scene (using differently exposed LDR images), and in the next task we will use the trained model to get the optimal tuning for the image to result the right color space. Here we could select a group of main parameters which are the number of learning tasks, the learning strategies, sensor data used, number of input exposures.

Keywords: Deep Learning Neural Networks, HDR images, light intensity.

1. INTRODUCTION

In recent days, in image processing tasks, High dynamic range (HDR) images are an important concept helps to create a very high-resolution image. Used in graphics, computer vision, or any other graphical technology. HDR means that we want to increase the intensity levels to a wide range. In another word, we want to increase the intensity levels captured from a real live image to create a new high resolution one.

Dynamic rang is defined as the ratio between the maximum and the minimum intensity values that can be transmitted in such a reliable method, or able to be reconstructed again, this task is done using any available computing system.
In another definition, we can define the dynamic range as the amount of data the camera can capture to the maximum exposure in a scene, that means we have a large scale of intensities ranging from the darkest part in the image (which are the dark parts of the image or darkness all over the image) to the lightest parts in the image that is captured, also known as HDR.
In a normal situations, we could find the effect of dynamic range if we take the pictures in sunlight where we have enough light that gives a good output image, in such an image, differences between lighting scales are noticeable, depending on the exposure of the camera, and the sky can be exaggerated, or it may Shadows appear under trees and some parts in the image will appear black.

In the following example, you can tell the difference between a photo taken without or with HDR:

Figure (1) the different between HDR and LDR images, we can see in the HDR image the color saturation between several classes

Dynamic range or HDR is measured in what we can call stops, each stop value equals twice or half the amount of light, as that defnition we can say that we can double the light of the image captured when we increase the exposure by one stop, if you shoot at a shutter speed of 1/100, then the exposure of one stop will be 1/50, Whereas it will be one stop darker than 1/200.

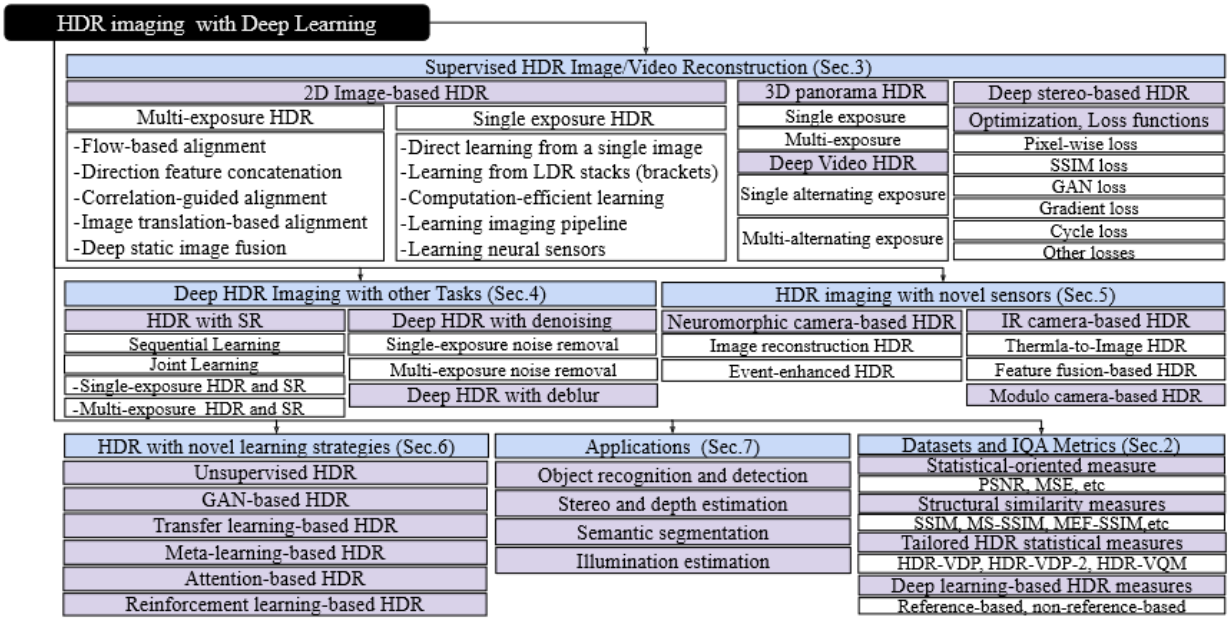
In addition to that, if we take a picture using one stop, that means the picture will give a scene where the lightest objects in the images are bright as the darkest part, the relation of the stops is nearly exponential, which means, if the camera has a sensor with two stops of dynamic range, the most light area of the scene will be much more brighter than the darkest point, bypassing this Borders will result in an image with distinct highlights or black shadows.
most of the recently built cameras have a large dynamic range, and a high-performance shooter, it's more common to see cameras with 12 to 14 stops, knowing that the eye Humanity has a dynamic range of 20 stops.
This HDR images needs a dataset of images for the same scene collected fron shadow to sunlight [1-2]. By using HDR images, we could be able to capture and store real world lighting, transmit it, or even process it for many tasks without need for the linearization of the signal or working with he clipped values [3]. If we want to capture a real conventional HDR image, we need to use special cameras [4-6]. But these cameras are so expensive for normal users. So, there is another solution to create a rendering tool that can create an HDR images from the normally captured images which are studied and used in several gaming and virtual reality tasks [3-4, 7].

All of previous problems result the necessary of a method to create a HDR images using the low dynamic range images using several methods. One of these techniques is generating HDR image by fusing multiple LDR images. The main problem of this method is that we need to use a special software or hardware to get images in different exposure situations. So, it became hard to create the dataset we want to use. So, other studies headed to create HDR image from a single image [10-14].

In the development of new technologies, deep learning is one of the most advanced one used these days. This development gives us the DL-based HDR imaging, some of the studies used a Deep Neural Network (DNN) models which was developed from a wide range of structures starting from conventional neural networks (CNN's) [9-10,16] to generative adversarial networks (GAN's) [17-19]. Table (1) shows several techniques that are used to create HDR images.

Table(1) several techniques that are used to create HDR images.

DOI : 10.31695/IJERAT.2023.9.5.1



2. IMPORTANCE OF THE SEARCH

Not all the mobile phones has an HDR capability. In addition to that, if the mobile phone (or camera) is able to capture an HDR images, it will loose a lot of textures and color scale when it is quininized to save in the device storage unit.

Our search aims into two targets:

- Restoring and substitute the looses and errors that was effected by the quantize task.
- Reconstruct a new HDR image from low HDR image or from LDR image with several scales of resolution.
- Create a system that is able to work on low processing power to increase the ability to create HDR image on PC's or mobiles.

3. SEARCH METHODS

3.1. Digital imaging

As we said before, Natural scenes has a large range of intensity values. To create an HDR image we could control some performance concepts like exposure time of the shot. Camera pipeline structure is a non-linear system. If we have a pixel value at a location (I,j) can defined by the equation:

$$Z_{ij} = f(E_{ij}\Delta t)$$

Where Δt is the exposure time and E_{ij} is the irradiance value at the same position. We call f the function of the camera which is non-linear. Now, if we have the values of the specific pixel in different exposed images for the same scene we can get an estimated noisy image, using this method Debevec et al. (1997) create a weighted function $f^{-1}(Z_{ij})/\Delta t$ to estimate the real pixel value E_{ij} .

To use neural networks for HDR image creation, the input of the network must be pixels values of LDR image of static scene. This scene is sed then to predicate the irradiance values or we can map the HDR scene.

We cannot build an exact and accurate mapping operation, but only approximated one.

3. CNN ARCHITECTURE

There are many types of neural networks. CNN or convolutional neural network is one of these neural networks types, and it is one of the best neural networks that are used in image processing[1], these neural networks are based on the changing weight structure using a collection of filters that slides along the input features matrix to create a map structure between the input and the predicated output[2-3], most convolutional neural networks are working in the manner of translation/transformation, unlike static traditional neural networks that are working on the changing and building a feature selection model.[4] CNN neural networks are used in a wide range in image and video recognition, image reconstruction or any image processing tasks[5], other usage of these networks on the digital images are classification, image, medical image analysis. [6] these system advance to a new level using the brain-computer interfaces.

DOI : 10.31695/IJERAT.2023.9.5.1

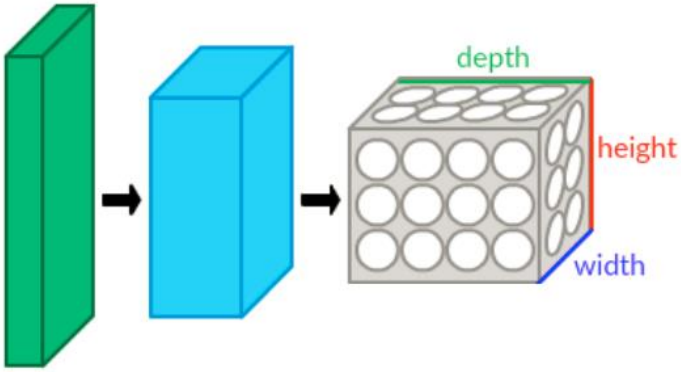


Figure (2) CNN network structure

CNNs are another structure of layered perceptron's neural networks. These networks are fully connected where, every neuron in a layer is connected to all neurons in the next layer. This full connected structure makes these networks to became able to act with high overloading. Another advantage of these networks that they are able to prevent overfitting using multiple methods like for example, changing the training parameters during training task (for example weight changing) or disconnection the non-valuable neurons using connections skipping or adding a drop out layer, etc. CNN's used filters to decrease the complexity of the pattern to get the most important features and discard the non-important ones.

4.1. Bypass layers

In CNN, first of all we must fit the model entered with the input of the tensor which has the form of $N \times M \times P \times r$ where N is the number of inputs and M is the heigh and P is the width and r is the gray scale. This input is passed through a convolution layer, the image becomes resulting a structure of feature-to-feature map, or an activation map, with the form $x \times y \times z \times s$ where x is the number of entries and y is the feature map heigh, z is the feature map width and s is the grayscale traits of the map.
The convolutional layer inside a CNN generally has the following properties:

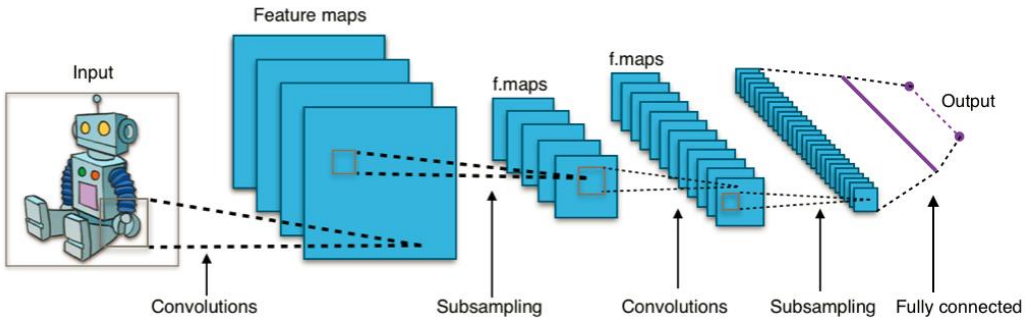


Figure (3) steps of CNN working
Bypass or kernel filters uses width and height parameters or what we named them as the ultra-parameters. In the same way we define the hyperparameters as the number of input channels and output channels. The depth of the network is equal to the number of output channels of its input.
another parameters for the convolutional process are, padding, stepping, and stretching.
thesse layers wrap the input and sends the result to the next layer. This can help in most working tasks when we need a large size inputs like video streams and high-resolution images. For the image processing task, It requires a large number of neurons, a fully connected layer of a (small) 100 x 100 image needs nearly 10,000 weights to be added to each neuron in the next layer. So, we can use the convolution to decrease the number of free parameters, allowing the network to be deeper.[15]
For example, if we have an image with a 5 x 5 tile area, which has the same weights, it requires just 25 parameters. uniform weights can help us to solve the vanishing gradients and pop-up gradient that can affect to the whole work of the network during back propagation in traditional neural networks. [16][17] . the spatial relationships between the separate features are processed during convolution task too.

4.2. Aggregation layer

Aggregation layers are used specially to reduce the dimensionality of data. This can be done by merging the output of clustered neurons in previous layer and compressing it into just one single neuron in the next layer. We can use the local grouping task too that can combine small batches, the most used size for the filters in this layer is 2" x 2". Global aggregation works using all neurons in the feature map. [18][19]

DOI : 10.31695/IJERAT.2023.9.5.1

This aggregation is named the pooling task too which could be either max pooling or average pooling. In maximum pooling we choose the biggest value under the filter area while we use the average value of all values under the filter mask in averaging pooling.

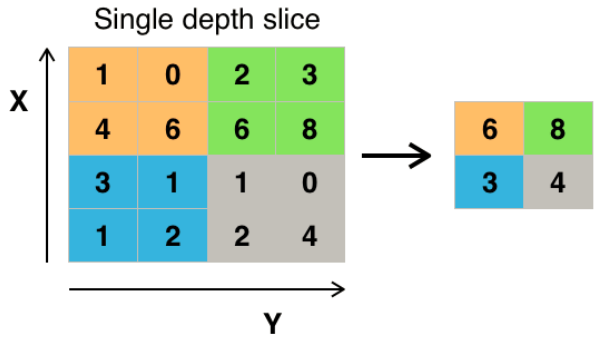


Figure (3) max pooling task in neural network

4.3. The fully connected layer:

As we define before, fully connected layer task is to connect each neuron in the input layer to all neurons in the next layer. This layer helps the neural network to share data and features between all neurons so some of them will learn more than others and this task can help us to dropout the non-important neurons.

4.4. Weights

when the input of the neural network is entered. Each neuron starts to compute the output by using a specific function. The connection between each neuron has a specific value that is named the weight. Because we have several connections, so the weight values must be in a matrix form. We have also the bias value that is added to the form studied. The simplest form that connects the input and the output of the network defined as:

$$y = w * x + b$$

where x, y are the input and the output, w is the weight matrix, and b is the bias.

The weights and bias vector is defined in the network as a filter too, and many neurons can work with the same filter which can help to decrease the memory requirements while, in counter, every sensory domain object must have its own bias and weight filter.

4.5. Local contact in CNN architecture

one of the most important problems that we can face in neural networks is the dealing with high dimensions data. In our search, if we work with an image, we have a large amount of pixels which makes the connection between all neurons an insufficient method.
The main problem here that the CNN networks does not take all the input data if we are working in spatial domain. The CNN try to detect the connectivity between pixels to find the most important areas in the image. This adjacent is added to weights and we can let each neuron to deal with special part in the image.

5 SPATIAL ARRANGEMENT

there is three super parameters that controls the output size of each layer which are the depth of the network, pitch and fill size.
The depth of the output layer will define how many of neurons are connected to the input in same pixels area of the input layer, this task helps to detect different types of features in the same area in the image. So, there are some neurons are able to detect edges while others are able to detect colors for example.

In other hand, Stride controls how the filter could move all over the image, if we have a stride value of one, that means the filter will move one pixel each step and it will overlap when moving across image pixels. To avoid overlapping, noramally we can choose a value of stride equal or larger of 3.
Some times it is preferred to add padding of zeros around the edges of the image and the size of padding is the third hyperparameter. Some times we may fill the padding with the same values of the edges instead of zeros which is called the “same” padding.
The equation that fits the input given size is:

$$\frac{W - K + 2P}{S} + 1$$

DOI : 10.31695/IJERAT.2023.9.5.1

Where W is the input volume, K is the volume of the bypass neurons, S is the step and P is the padding size.
. In general, setting a zero fill to be P = (K − 1) / 2 when the step is S = 1 ensures that the input size and output size will have Spatially the same size.

6. ReLU LAYER

ReLU is an activation function that is used for corrected linear unit. It is used to delete the negative values and setting them to zero. It is defined as a function:

$$f(x) = \max(0, x)$$

This function helps to give the non linearity to the mapped decision function without affecting the fields of the convolution layers. There are another functions used for this task like the saturated hyperbolic function:

$$f(x) = \tanh(x)$$

And the sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}}$$

6.1.Fully connected layer

Which is the last classification layer. Here, the activation task is computed as an affinity transform, with matrix multiplication followed by bias compensation (addition of a vector for an acquired or constant bias term).

6.2.Calculating the number of filters

When the depth of the neural network increases, the map size of the network decreases.
Input layers use a small number of filters, and when we start passing into the higher layers, we can use more filters. We must try to balance computation task over these filters by letting the production of feature values over the pixel position constant all over the layers.
Here we must be sure that the total number of activations does not decrease from one layer to another.

6.3. Filter size

the filter size depends on the dataset used and the structure of its data. The filter size must be selected correctly without resulting the overflow problem. The max pooling filter size in normal CNN networks is selected usually with a size 2X2 and a step of 2. This means that we are down sampling the input to increase the computation tasks.

6.4. DropConnect

Drop Connect is a task that can be done in each connection, each output unit can be dropped with a probability of 1 − p. Thus each unit receives input from a defined subset of units coming from previous layer.
Drop Connect has a very important task to introduces dynamic variance within the model, the variance here can be found in the weights, the output vectors of the layer.

7. Related works:

First of all we must say that if we want to collect all the illumination range of an image we will need to create a exposure multiplexer. If we want to catch a static scene then we do not have a problem about the exposure as we are having a single image. But in the dynamical situation it is different as we need a special way to align the robust exposure, to solve this problem we could use several methods like multi-sensor imaging that was created by [Kronander et al . 2014; Tocci et al . 2011] or we could use varying the per-pixel exposure [Nayar and Mit-sunaga 2000] or gain algorithm.
These systems suffers of many problems like bulky and custon built systems, a problem of calibration for the scene, and results an increasing in image resolution.

7.1. Inverse tone-mapping:

It is a concept developed by Banterle et al. 2006 to define how to reconstruct an image with high quality using several functions types. The transformation functions gives multiple results varies from too bad ones to good ones. For example, Masia et al. 2009 gives a global pixel transformation with more input materials to use. Akyuz et al. 2007 gives the linear scaling method and Masia et al. 2009, 2017 study the non-linear method. All the se methods were unable to solve the problem of lost pixels and information in the image.
Another way used to create HDR images that depends on reconstruct of pixels by the pair-wise compression experiment on an HDR display used by Banterle et al. [2009]. Where Meylan et al. [2006] try to use a linear transformation with a highlight some different scaled regions of interest. The author used first of all a linear function to linearize the input and then used a boosting

DOI : 10.31695/IJERAT.2023.9.5.1

highlight with an expanded map created by using the median cut algorithm. This technique was used for video processing with a use of self automated calibration with cross-bilateral filtering of the expanded map. Remple et al.2007 used another type of filtering by using the gaussian filter to achieve a real-time performance. Some researchers used what named semi-manual methods like that method proposed by Didyk et al. 2008, working on splitting the image into its diffuse, reflection, and lights sources components. Because the human eyes work specially with light source and the reflection the author focused on them and left the diffuse with no change at all.

7.2. Bit-depth extension:

In imaging systems, and when we want to catch an image using a camera we have first of all the quantization effect on the 8-bit standard image that may decrease the resolution of an image. To avoid this effect we can use a bit-depth extension, a method that uses a dithering methods, these methods depends on noise to hide the banding artifacts as Daly and Feng shows (2003).
Another method was given by Daly and Feng too, this method can used by applying the low-pass filters firstly and then using a quantization method to detect false contours. Another edge preserving filtering methods were used for the same purpose.

8. Search sources and constraints:

The main aim of this search is to create a model to predicate the pixel value of an LDR image captured using a normal camera and create the saturated one. Here we want to predicate the output pixel value by combination between the predicated pixels and the linearized input image. The out put pixel will be computed by using the equation:

$$\widehat{H}_{i,c} = (1 - \alpha_i) f^{-1}(D_{i,c}) + \alpha_i e^{\hat{y}_{i,c}}$$

Where $\widehat{H}_{i,c}$ is the reconstructed pixel which have the index I and the color channel c. $D_{i,c}$ is the input pixel value, $\hat{y}_{i,c}$ is a logarithmic output of the conventional neural network. And f^{-1} is the inverse camera curve used to transform the input used by the system to a linear form. Here we can use a linear ramp starting from a specific pixel value with a specified pixel threshold value τ and will end with the maximum pixel value. It is calculated with the equation:

$$\alpha_i = \frac{\max(0, max_c(D_{i,c}) - \tau)}{1 - \tau}$$

We can select $\tau = 0.95$ if the inputs are defined in the range between [0,1]. Selecting this value lets us to get more accurate in preventing banding between normal predicated pixels and their surrounding areas.

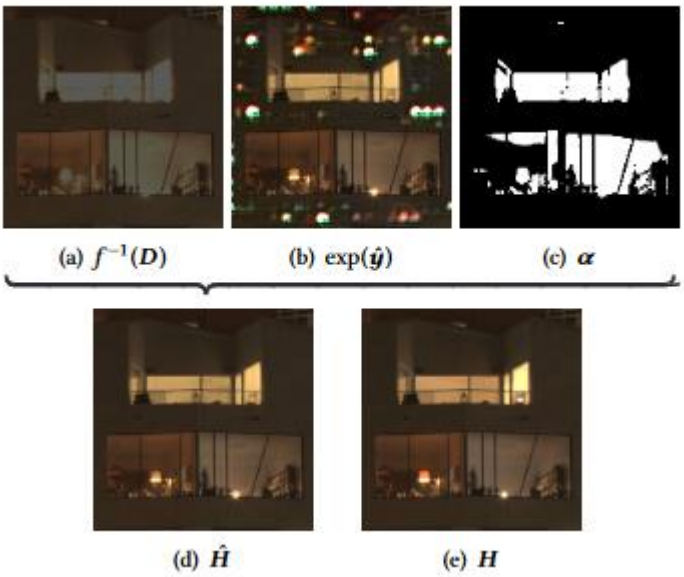


Figure (4) steps done to create an HDR image

When we work on the model, we found some problems that affect the whole model performance which are:

- The first problem is that each image has many small textures that are not able to be predicated in accurate way and may result a noise in the output image.
- When working with HDR images we will suffer of the color ranges of the images, that mean when we want to create HDR image from single LDR image many of color gradients will disappear causing a loss in the output image.
- Quantization of the image after taking a photo will cause too a loose in images details that decreases the resolution of the output image.

9. Encoding of HDR image using CNN

DOI : 10.31695/IJERAT.2023.9.5.1

The system depends on CNN neural network consists of a conventional layer and a max pooling layer with input size of $\frac{w}{32} \times \frac{H}{32} \times 512$. Where W, H are the sizes of the image. Our work depends on VGG16 deep learning network created by Simonyan and Zisserman 2014], which is without fully connected layer because here we care about the weights not a classification task. Our encoder will work on the LDR image so we will need a decoder to work on the HDR full range image. This decoder will work on the logarithmic domain to get all the colored range.
Here we will use a loss function compares the output that we have from the network with the real HDR image. For this case we need an up-sampling task with a filter of a size 4X4.
All the network layers are using ReLU activation function followed with normalization layer.

10. Training task

First of all we want to say that dynamic range will be different between LDR and HDR images. Figure () shows the different between histograms of HDR and LDR images. As we can see from the histograms, LDR images has a uniform distribution over all the image pixels except the noisy areas which are causing loses and shown as peaks in the histogram. Where in HDR images we have got non saturated pixels with decays long tail as a exponential function.

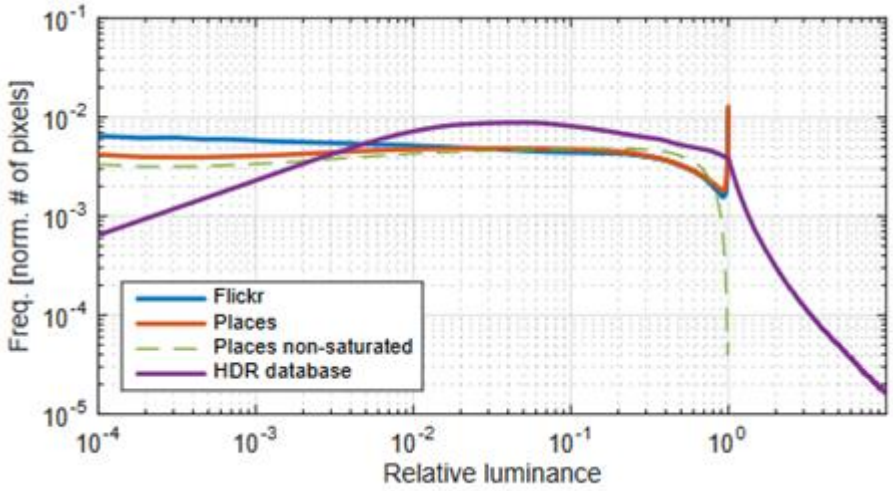


Figure (5) Histograms of two LDR images datasets compared with the pre-processed HDR dataset. As we can see, the probability has a large increase close to 1 which means a saturated information. The HDR image has a saturation information too represented in the tail of decreasing frequency

10.1. Training the dataset

First of all, when we want to train a neural network, we must initialize the weights of the network. If we start with zero value weights it needs a long time to get the output. So, instead we decide to use the weights of pre-trained network VGG16 with a large scale of images classification. Here we will use the ADAM optimizer with learning rate of $5X10^{-5}$ with a loss function defined using the equation:

$$\mathcal{L}_{IR}(\hat{y}, H) = \frac{\lambda}{N}\sum_i \left|\alpha_i(\log(I_i^{\hat{y}}) - \log(I_i^{y}))\right|^2 + \frac{1-\lambda}{3N}\sum_i \left|\alpha_i(\log(R_{i,c}^{\hat{y}}) - \log(R_{i,c}^{y}))\right|^2$$

Where N is the number of pixels and y is the LDR image and $\hat{y}$ is the predicated HDR image, we give y in the equation $y = \log(H + \epsilon)$ where ϵ was added to remove zero pixels values. I is the intensity value and R is The reflectance that stores information about details and colors. We can define the values as, $, c = Ii\ Ri, c$.
λ is a parameter defined by the user, it defines the relation between luminance and reflection.

10.2. Pre-training of simulated data:

HDR datasets are not available too much, because of the difficulties to capture them and the large size of the dataset that result. So we can use a normal HDR dataset that requires all the images not to contain saturated regions in the images.
For training, we calculate the histogram of the dataset at first, and then for threshold we use a constant value $\xi = {}^{50}/_{256^2}$ which means that if less than 50 pixel has the maximum value, we will use images in training with a size of 224X224 pixels resolution without resampling and an ADAM optimizer with learning rate of 2×10^{-5} with a patch size of 4.

11.Results:

After the training tasks and the reconstruction of the HDR images we will evaluate the whole system performance. We have evaluated our system performance using 95 HDR images which were reconstructed into 1024X768 pixels.

DOI : 10.31695/IJERAT.2023.9.5.1

First of all, without skipping connections between pixels, out CNN network works fine with low MSE error between input and output. With skipping connecting pixels the error was decreased by 24% and the resulted images were with more accurate details. Table (2) shows the MSE error of each method used in the study.

	direct	I/R	I	R
Reference	**0.999**	**0.890**	**0.712**	**0.178**
Without skip-connectio	**0.249**	**0.204**	**0.102**	**0.102**
Direct loss	**0.189**	**0.159**	**0.090**	**0.069**
I/R loss	**0.178**	**0.150**	**0.081**	**0.068**
Pre-train with I/R loss	**0.159**	**0.134**	**0.069**	**0.066**

Each row in the table shows the way used in image reconstruction. While each column shows the error.

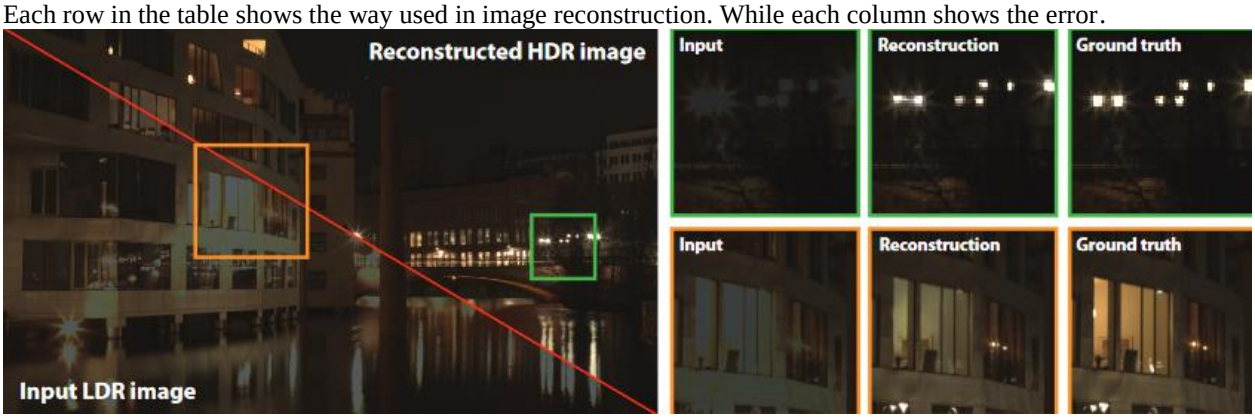


Figure (6) comparing between the reconstructed image with the real HDR image

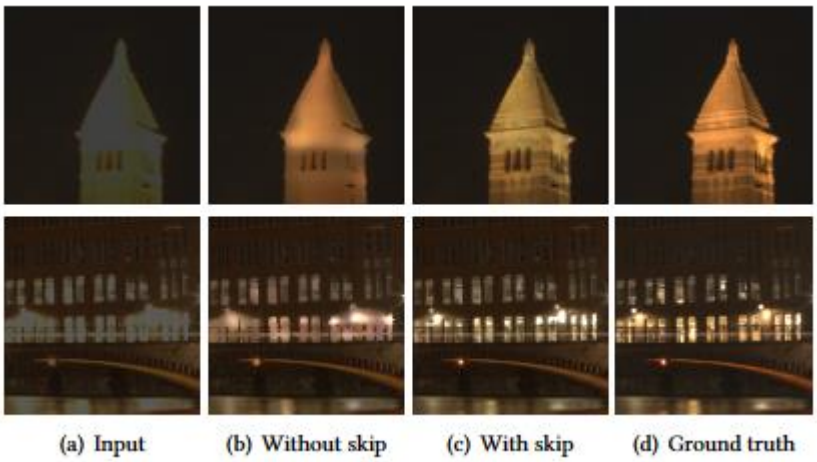


Figure (7) working without adding domain transformation skip-connections (b) and with it (c), as we can see the autoencoder was able to reconstruct high luminance

11.1. Training number of epochs effect

After building and evaluating the neural network we try to make sum changes on the neural network and calculate the results. It is clear that if we decrease the number of epochs that the neural network is working on will decrease the memory requirements and the time required to finish the training task.
The main network works on 10 epochs, when we decrease the number of epochs to 5 we increase the speed 30% and the results we have got were:

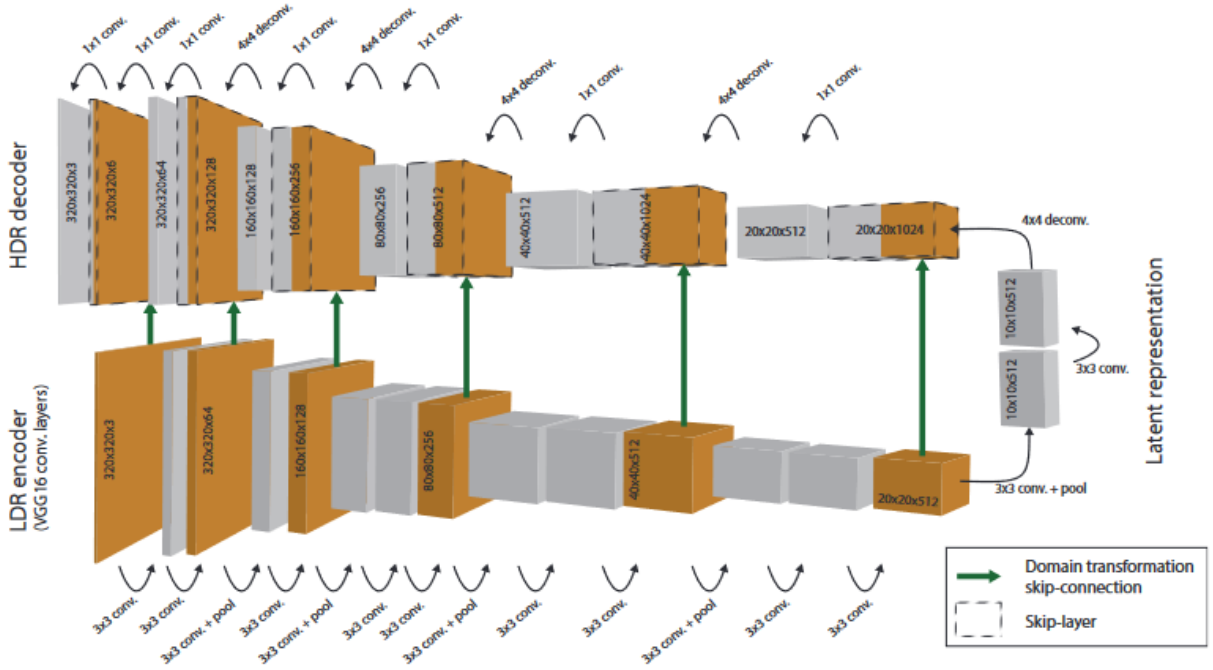
	direct	I/R	I	R
Without skip-connectio	**0.308**	**0.298**	**0.206**	**0.209**
Direct loss	**0.213**	**0.188**	**0.12**	**0.089**
I/R loss	**0.209**	**0.176**	**0.098**	**0.086**
Pre-train with I/R loss	**0.185**	**0.158**	**0.098**	**0.084**

The MSE error decrease to about 25% that mean we must train the network with more epochs to reach the maximum accuracy. We must say here that we can not increase the number of epochs to infinity value because the network will enter the overfitting situation that requires impossible memory requirements and unacceptable time to solve the problem.

DOI : 10.31695/IJERAT.2023.9.5.1

11.2. Changing the CNN structure effect:

Figure (8) shows the structure of VGG16 CNN network used, but what about if we decrease the layers used in the network?
As we can see in the figure, the structure contains of a number of convolution and deconvolution layers changing the size of the trained pixels. We work on deleting the middle layers and try to re-calculate the results.



	direct	I/R	I	R
Without skip-connectio	**0.456**	**0.312**	**0.305**	**0.308**
Direct loss	**0.345**	**0.234**	**0.205**	**0.123**
I/R loss	**0.314**	**0.258**	**0.158**	**0.118**
Pre-train with I/R loss	**0.256**	**0.204**	**0.156**	**0.148**

As we can see from the table above, the error increases too much because the neural network used skips many important pixels when traying to fit the pixels between layers. This skipping task results a high loose of image structure in the network including edges and color illumination.
The speed of working increased about 48% from the normal task but the resulted images were too bad as a HDR structure. Increasing the number of layers by creating a new neural network will increase the accuracy of the network resulting a very high accuracy HDR images but they need, again, a large amount of memory and processing unit (even if the work done on a server with multiple processors).

12.Conclusions:

HDR images are a recommendation for each one, these images has a larger intensity, luminance, and color range. But not all devices are able to introduce this image. So our task was to find a method to create a HDR images using low scale LDR images by training a neural network. This neural network will train from the HDR dataset and then apply the training result on multiple types of images.

The trained network works especially on color ranges and illumination in addition to small structures and edges. The work returns images in a good resolution and less in accuracy from real HDR images with nearly 20% at maximum which is a good result.

REFERENCES

[1] K. Debattista, P. Ledda, F. Banterle, Exapanding low dynamic range videos for high dynamic range , SCCG, 2008.
[2] P. Callet, F. DufauxR.K. Mantiuk, M. Mrak (Eds.), high dynamic range video, academic press. 2016.

DOI : 10.31695/IJERAT.2023.9.5.1

[3] J. Llach, S. Bhagavathy, and J. f. Zhai. Multi-Scale Probabilistic Dithering for Suppressing Banding Artifacts in Digital Images. IEEE International Conference, 2007.
[4] Y. Deng, C. Dong, C. Change L., X. Tang. Compression Artifacts Reduction by a Deep Convolutional Network. The IEEE International Conference on Computer Vision (ICCV '15), 2004.
[5] X. Feng, S. J. Daly, spatiotemporal microdither based on models of the equivalent input noise of the visual system. 2003.
[6] P. E. Debevec , J. Malik. Recovering High Dynamic Range Radiance Maps from Photographs, 1997.
[7] M. D. Grossberg, S. K. Nayar, the space of camera response functions?. the IEEE Conference on Computer Vision and Pattern Recognition,2003.
[8] M. Hein, R. Mantiuk, P. Didyk, and H.P. Seidel. Enhancement of Bright Video Features for HDR Displays. Computer Graphics Forum,2008.
[9] Y. Kanamori, Y. Endo, J. Mitani. Deep Reverse Tone Mapping. ACM Trans. Graph, 2017.
[10] A. Gilchrist, A. Jacobsen. Perception of lightness and illumination in a world of one reflectance,1984.
[11] X. Glorot , Y. Bengio . Understanding the difficulty of training deep feedforward neural networks. the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS '10), Vol. 9. 249–256, 1996.
[12] F. Huszár, J. Caballero , C. Ledig, L. Theis, , A. Cunningham, A. Acosta, A. Aitken,
Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint arXiv:1609.04802 , 2016.
[13] I. Goodfellow, J. Pouget-Abadie, Generative Adversarial Nets. In Advances in Neural Information Processing Systems 27. 2672–2680, 2014.
[14] S. Hajisharif, J. Unger , Adaptive dualISO HDR reconstruction. EURASIP Journal on Image and Video Processing 2015, 1, 41, 2015.
[15] X. Zhang, S. Ren, K. He, and J. Sun . Deep Residual Learning for Image Recognition. ,2016.
[16] G. E. Hinton , R. R. Salakhutdinov , Reducing the dimensionality of data with neural networks. Science 313, 5786, 504–507, 2006.
[17] S. Ioffe ,C. Szegedy . Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,2016.
[18] S. J. Daly, X. Feng. Decontouring: prevention and removal of false contour artifacts. Proceedings of SPIE, Vol. 5292. 130–149, 2004.
[19] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization. CoRR abs/1412.6980. http://arxiv.org/abs/1412.6980 , 2014.
[20] R. P. Kovaleski, M. M. Oliveira. High-quality brightness enhancement functions for real-time reverse tone mapping. The Visual Computer 25, 5 (2009), 539–547, 2009.
[21] E. Simo-Serra, S. Iizuka, H. Ishikawa. Let There Be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification, 2016.

DOI : 10.31695/IJERAT.2023.9.5.1