# HDR image generation by GAN

A Project by
1705501 JOYDEEP GHOSH
1705534 SAYAN BASU
1705494 ASHISH DASH
1705551 SHUBHAM PATRA

# THE IDEA

The whole idea of this project is to create an hdr image from a low dynamic range (ldr  image ) by GAN (Generative Adversarial Networks )

The whole process is developed with aim to reduce the cost of generating HDR content and further the research in the field of hdr imaging .

# Intro

**Traditional efforts to generate hdr images have been done by CNNs or other feed forward neural networks .**

➔ **Traditional**

Using an expensive optical sensor to capture images in native hdr .

➔ **New Approach**

Using CNN (convolutional ) networks to generate hdr images from ldr ones

➔ **Our Approach**

We are using an GAN to generate hdr images from ldr images.

# WHY we use GAN ?

Well the problem with previous approaches have been that they either require an expensive sensor to capture images accurately . or they require training a cnn to generate hdr images . the problem with the latter approach is they require different sub networks to be trained for generating different exposure images because each cnn only determines the mapping between input ldr image and  the image with exposure of T (T=1,2,3,..n).training and testing so many different networks increases complexity.

# Our approach!

Therefore we followed an approach of training a GAN (Generative Adversarial network) . we trained a GAN to generate images with a exposure difference of

EV+1 and EV-1. the image generated in a single step is used recursively to generate the image with subsequent exposure difference of +2 and -2 and so on .

Our GAN had four networks namely G+, G-, D+ ,D- or a generator and discriminator each for positive exposure and negative exposure .

Our approach has 4 networks as compared to traditional cnn approach of 2N

Networks where N= no of layers of exposure (N=1,2,3,4,5,6,...n)

# Some Information about GAN

## GAN

GAN or generative adversarial networks are form of neural network which is used for estimating generative models via adversarial process. It has two components a generator G and a discriminator D .

G captures the data distribution

D estimates the probability that a sample came from actual data  rather from G

The Training process for them is to maximise the probability of D making a mistake and therefore make the generator

So good that its output is indistinguishable from real data

# Training and Testing

- ➔ **Objective functions**
- ➔ Lambda = relative weight
  Gplus=argmin LSGAN(G)+ lambdaL1(G) for Iev+1 and I
- ➔ Gplus=argmin LSGAN(G)+ lambdaL1(G) for Iev-1 and I
- ➔ Where Iev+1 is image with relative exposure of +1 and Iev-1 is an image with relative exposure of -1.
- ➔ **Loss functions (x=ldr image,y=reference,z=noise)**
- ➔ LSGAN (G) = Ex,z [(D(G(x, z), x) – 1) ^2 ],
- ➔ LSGAN(D)=½*E x,y [(D(y, x) – 1)^ 2 ] +½* E x,z [(D(G(x, z), x)) 2 ],

## Training process

Our training process took 1 week on a nvidia gpu with cuda . we achieved a accuracy of 90-95 % .

we trained the generators to minimize L1 loss and defeat discriminator networks. The discriminator distinguishes the pair (reference, input) from the pair (estimated image, input) as the training progresses.

L1 loss=E x,y,z [||y − G(x, z)|| 1 ].

L1 loss is also known as mean absolute error or content loss since it's a pixel wise difference between two images . we also thought of using l2 loss but using l2 norm generates a blurred image.

Training In the first training phase, we used only L1 loss, and in the second training phase,

we additionally used GAN loss

# Training process

Requirement- > Tensorflow 2.2, python3.8,cuda , nvidia gpu.

# Training &Optimisation

We describe the training and optimisation process

Our training pipeline included at the first stage training the Discriminator to accurately distinguish between real and generated image.then using the content wise losses from the discriminator to train the generator . this process continues till the discriminator achieves a accuracy of 85% .

Then we proceed to the optimisation process"

In the first stage we use only l1 loss and in the second phase use l1 + Gan loss as well.

We use adam optimiser for this process with a learning rate of 0.0001 and b1 = 0.5 and b2=0.9

# Inference & Dataset

**we generated images I EV 1 and I EV −1 from the given LDR image**

**using G plus , G minus . In the next phase, we obtained I EV 2 ,I EV −2 by using I EV 1 and I EV −1 as the input of G plus and G minus , respectively.**

**We recursively repeated this process for creating a multi-exposure stack**

Dataset -> We use **Funt et al. HDR Dataset .**it is a data set of images of 105 scenes captured using a Nikon D700 digital still camera.The camera's auto-bracketing was used to capture up to 9 images of exposures with 1 EV (exposure value) difference between each in the sequence. The rate of capture was 5 frames per second. The exposure range was set to ensure that in each set there would be at least one image with maximum digital count less than 10321. During bracketing, the camera was set to allow it to adjust the shutter speed and/or the aperture setting automatically between frames in order to change the exposure by 1EV. In other words, the f-stop setting was not fixed. All images were recorded in Nikon's NEF raw data format.

# WHAT NEXT?

Once we have the multi exposure stack

We proceed to our next phase which is the image recovery phase

Once we have the multi exposure stack we then have to reconstruct the hdr image.

The reconstruction requires the radiance map . once we have the radiance map we use tone mapping to extract image in the most dynamic range .

a)input ldr image

b)The high dynamic range radiance map, displayed by linearly mapping its entire dynamic range into the dynamic range of the display device

c) radiance map by linear mapping lower 0.1 % of the dynamic range

d)false colour map showing relative radiance values

e ) rendering of the radiance map using adaptive histogram compression

f) rendering of the radiance map using histogram compressionand also simulating various properties of the human visual system, such as glare, contrast sensitivity, and scotopic retinal response

# Image reconstruction

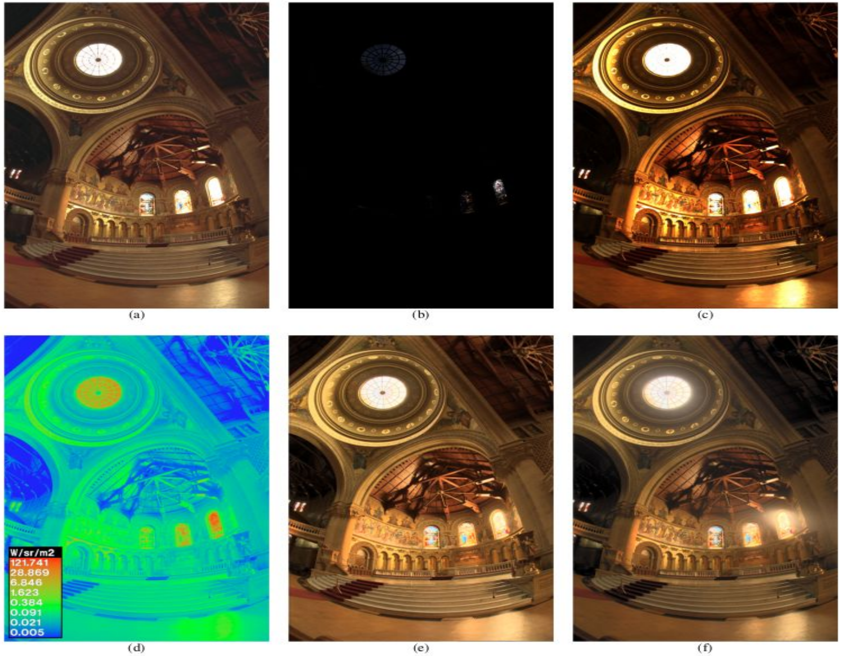


Figure 8: **(a)** *An actual photograph, taken with conventional print film at two seconds and scanned to PhotoCD.* **(b)** *The high dynamic range radiance map, displayed by linearly mapping its entire dynamic range into the dynamic range of the display device.* **(c)** *The radiance map, displayed by linearly mapping the lower 0.1% of its dynamic range to the display device.* **(d)** *A false-color image showing relative radiance values for a grayscale version of the radiance map, indicating that the map contains over five orders of magnitude of useful dynamic range.* **(e)** *A rendering of the radiance map using adaptive histogram compression.* **(f)** *A rendering of the radiance map using histogram compression and also simulating various properties of the human visual system, such as glare, contrast sensitivity, and scotopic retinal response. Images (e) and (f) were generated by a method described in [23]. Images (d-f) courtesy of Gregory Ward Larson.*

# The recovery process

We need to have the film response function to build the radiance map.

**The film response curve denotes the relationship between pixel values z and Exposure X**

**Film response function:**

In the first stage of the process, the film response to variations

In exposure X (which is E t , the product of the irradiance E the

film receives and the exposure time t ) is a non-linear function,

called the “characteristic curve” of the film.

# IMPORTANT TERMS

- EV value: this function calculates a list containing the Ev values of respective images .This function is crucial to create a Relative Ev value scale
- Weights: this weighting function is to create weight map from pixels
- Sample pixels : This function samples random pixels from training image stack to input into compute response curve functions

# Important terms

- ➔ Compute Response Curve : This function is used to calculate the Response curve of the camera used for data collection which is crucial to build radiance map
- ➔ Radiance Map : This Function is used to Create Radiance Map from the Response Curve function which is crucial for the reconstruction of images

# FILM RESPONSE FUNCTION

Lets assume the film response function to be f

E=irradiance value and t= exposure time

Let z be the pixel values and X exposure

z=f(X) ->(1)

X=E * T -> (2)

By 1 & 2

z=f(E * T)

f^-1(z)= E * T

By taking logn both sides

ln(f^-1(z))=ln(E)+ln(T)

g=ln(E)+ln(T) where g= ln(f^-1(z))

# Constructing radiance map

Once the response curve g is recovered, it can be used to quickly convert pixel values to relative radiance values, assuming the exposure  t j is known. Note that the curve can be used to determine radiance values in any image(s) acquired by the imaging process associated with g , not just the images used to recover the response function.

- $$Ln(E)=g(Z) - lnT$$

- E is the radiance map.

# Tone Mapping

We are finally in the last stage of our process. This stage is known as Tone Mapping Since standard display devices cannot display high dynamic range images therefore the Radiance map has to be fitted in the dynamic range of the display devices. Traditional devices can display only 8 bit data per colour channel that is 0-255. Whereas hdr image can be between 10-32 bits per colour channel . therefore having much more colour depth than traditional image

Tone mapping is a technique to map one set of colours to another to approximate the appearance of high dynamic range images in a medium that has a limited dynamic range like lcd monitors,projectors etc.

# THANK YOU

# Source Code-

https://github.com/joydeep2899/HDR-using-GAN/tree/joydeep