

1 Introduction

This model conduct the LDR to HDR conversion image generation (translation) task with single image. Similarly to the pix2pix, every input data has the assigned target data. Basically this model make a same size of image which has high color range. The detail about the implementation are written in the README.md and in the python files.

2 Environment

- L1 loss for the cost
- Adam optimizer (epsilon = 0.002)
- Learning rate: 0.0005
- Tau value = 0.95

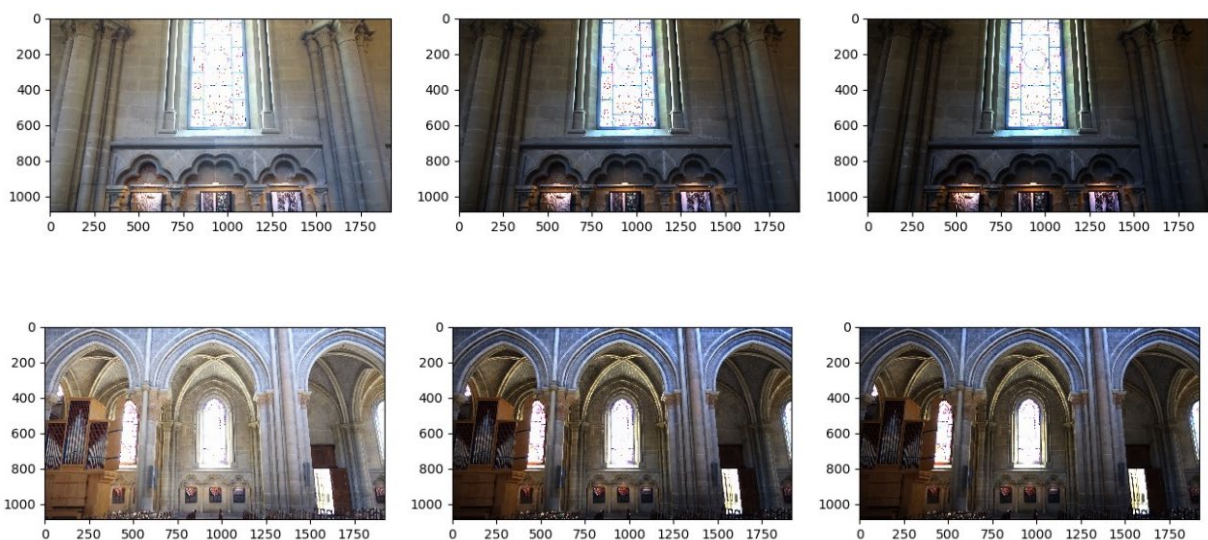
I could not follow the way that the author tried, but instead I put the LDR image at the training.

3 Result

After 1200 step,
Average training cost: 0.05
Average mPSNR = 13.4

```
Step: 00250    Cost: 0.211    mPSNR: 15.5
Step: 00300    Cost: 0.234    mPSNR: 12.4
Step: 00350    Cost: 0.214    mPSNR: 14.5
Step: 00400    Cost: 0.207    mPSNR: 14.1
Model
Step: 00450    Cost: 0.208    mPSNR: 14.2
Step: 00500    Cost: 0.211    mPSNR: 13.4
Step: 00550    Cost: 0.235    mPSNR: 13.1
Step: 00600    Cost: 0.203    mPSNR: 13.1
```

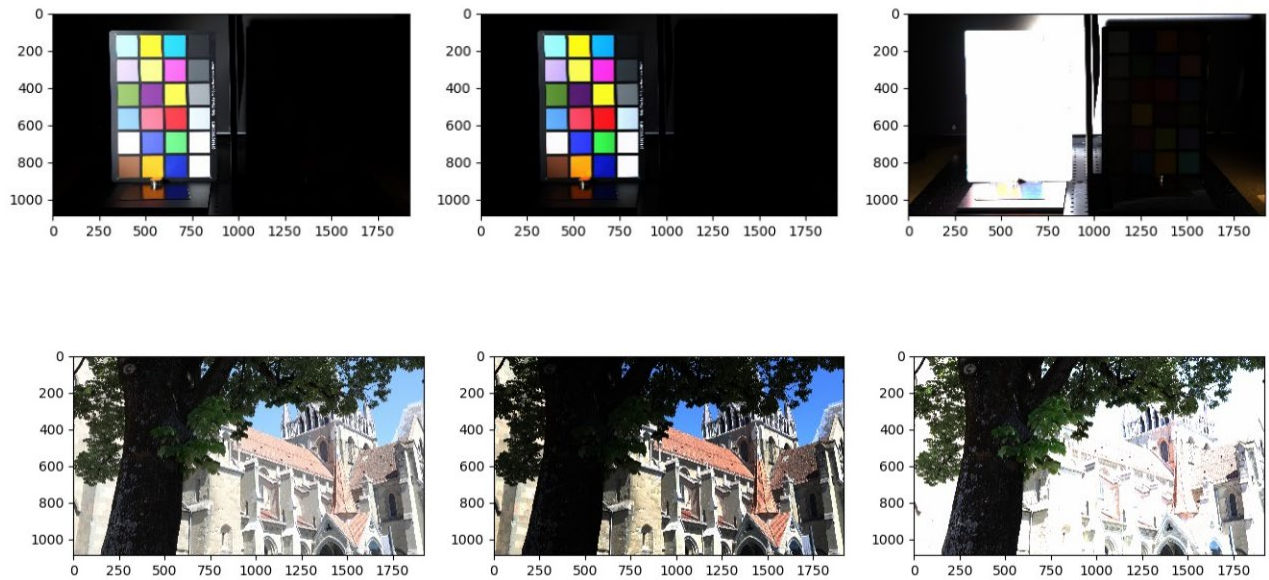
Recovered example (Left is input, middle one is output, and right one is the target)



(mPSNR 15.3, 14.7)

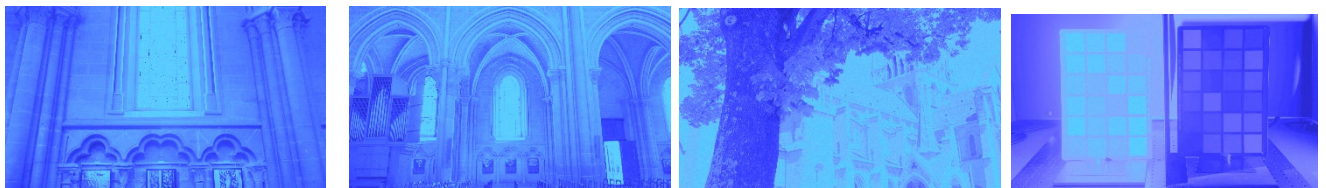
According to these results, we can see that the network well convert the LDR Image to HDR Image when the target image has fine difference of the contrast. However, when the target image has huge

contrast difference like data below, the output image not really follow the target and does not change that much.



(mPSNR 10.2, mPSNR 9.4)

When the image has a huge contrast, then the network does not follow the target that much. However, it seems the produced has more neat color scheme compare to the real target image. Technically the target image has wider color range (HDR), but the produced image seems more clear than real target image. In addition, this characteristic could be fixed by modifying the tau (threshold value).



This is the hdr vdp result of those Inputs, recovered, target image in order. According to the above results, there are some spots that has clearly different between produced image and target image, but the visualization does not catch them. Unfortunately, this code seems got a bug. Therefore, I attached the result files. (the single images in the 'data' directory is the recovered output)

The model does recover when the image has small difference because most of the data has similar distribution. The HDR image is more clear and has a neat color scheme compare to the LDR image. However, when the images such like below two comes, the network does not work as the target but make the similar neat and clear image with them. This seems the problem of data distribution, so if we prepare the data such like below ones more, then we can make the expected result.

Additionally, it seems there are some limits of the brightness that the network can produce, so there is a tendency that the network could not change the color value when the objects are extremely bright. This could be fixed by modifying the tau value.

