HDR Pipeline on U-net

Xinya Xu, Wenhao Jiang, Leilin Wang

Abstract

HDR, stands for high dynamic range, is a technique used in photographic imaging and films, to reproduce a greater range of luminosity than what is possible with standard digital imaging or photographic techniques. Common HDR technology are often created by capturing and then combining several different, narrower range, exposures of the same subject matter, which contains limitation for shooting moving objects. The goal of this project is to use U-net to reconstruct images to HDR, which does not have limitation whether the object is moving or not. We used photo sets with same object and different exposure rate to train our model, then compare our output HDR images with other related results.

Introduction

If you try to photograph a high-contrast scene, even with the perfect exposure, there are certain scenes that will always tend to get blown-out highlights or flat shadows. It's nearly impossible to find a happy medium in these types of situations. The solution for this problem is high dynamic range. Dynamic range is difference between the lightest light and darkest dark you can capture in a photo. Once your subject exceeds the camera's dynamic range, the highlights tend to wash out to white, or the darks simply become big black blobs. To solve this problem, people use HDR techniques, which combines photos taken at different exposure levels and then merge them together with software to a new image with an unusually high dynamic range that couldn't be achieved in a single photograph.

The HDR technology nowadays contains one problem, as they cannot deal with moving objects. Since taking photos of different exposure level doesn't take place at the same time, moving objects may have different location or shape in each photo. So, when the HDR software merge those photos with moving objects to a new photo, it is likely to generate a photo with ghost effect. To solve the limitation of common HDR technology, we decided to use neural network methods to reconstruct shot photos to HDR images, which does not have limitation of the objects.

Related Work

Our projects and methods are mainly inspired by the paper: *Learning to See in the Dark* (2018) which contains previous approaches that address the similar problem of our project. The purpose of this paper is to use machine learning to develop pipelines for low-light image processing, based on end-to-end training of a fully conventional network. The project contains approaches and methods that we referenced on. Further output comparison between the model of our project and the model of Chen et al. (2018) is shown in the results section.

Datasets

For the dataset used in the project, we shot photos to generate our own data. The generated data contains 293 sets, with 9 images in each set. In each photo set, 9 photos shot at the same objects in the same environment, each of 4256×2848 pixel resolution and at 1 EV stop per each of them. Shot objects contains varies of types, includes lightest light, night view, still life objects and moving objects.

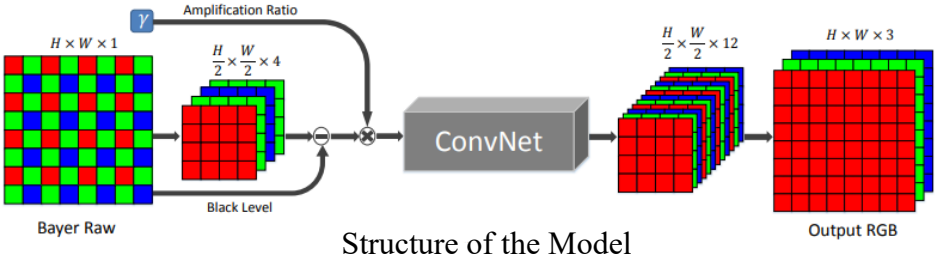
Example of one photo set

Methods

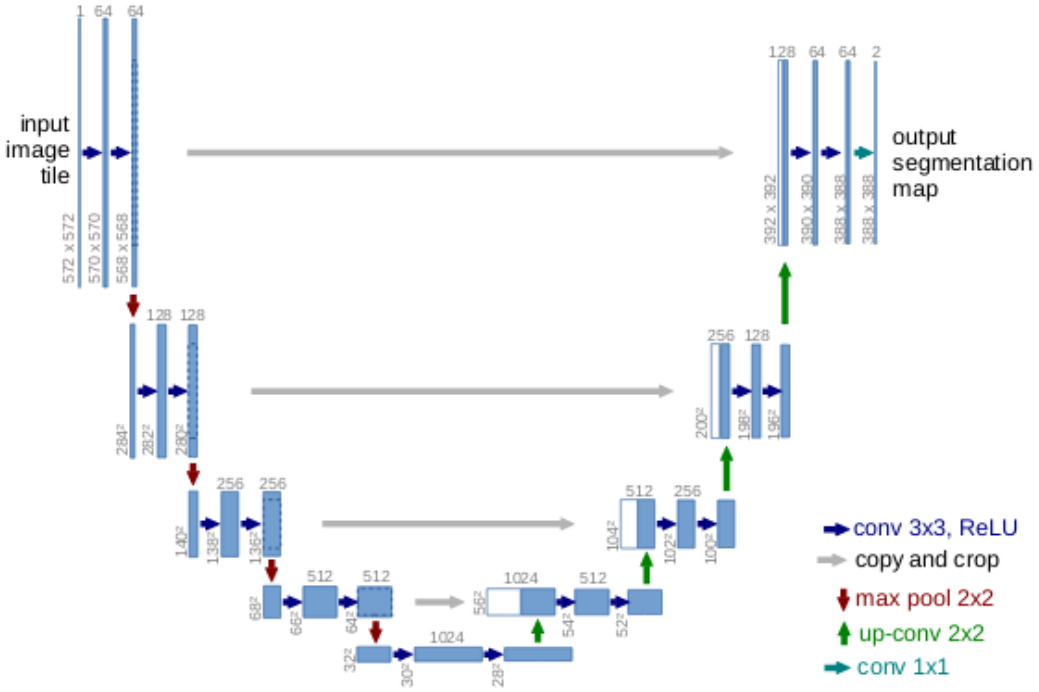
For 9 photos in each photoset, the photo with the lowest exposure is our input photo. Then we apply exposure fusion, a traditional image processing pipeline that takes the best bits from each image and seamlessly combines them to create a final 'Fused' image that is in low dynamic range, to generate a photo in perfect exposure, which are labeled data.

Generally, traditional methods would operate on normal sRGB images produced by traditional camera processing pipelines. Instead, we would operate on unprocessed raw sensor data, which contains more information for our future training.

The basic model structure is shown in the figure below. For the Bayer arrays, input is packed into four channels, and the spatial resolution is reduced by a factor of two in each dimension correspondingly. Then, the black level is subtracted, and the data will be scaled by a desired amplification ratio. This ratio is set to be the exposure difference between our input and the labelled data externally and is provided as input to the model. After processing, the packed and amplified data would be fed into a U-net, Convolutional Networks for Biomedical Image Segmentation. A U-net is very similar to a Fully Convolutional Network (FCN). One slight difference is that a U-net would have large number of feature channels in upsampling part, resulting in a more symmetric model architecture, as shown below. The output would be a 12-channel image with half of the spatial resolution. This half-sized output is processed by a sub-pixel layer to recover the original resolution.



Structure of the Model



Structure of the U-net

Training and Results

The loss function used is a simple L1 loss function, and the optimizer is Adam. When training, in each iteration, a 512*512 patch is cropped randomly for training. Then, random flipping and rotation would be applied for data augmentation. The learning rate is set initially to 10e-4, and will be reduced to 10e-5 after 2000 iterations. Total training would take 3000 epochs.

The results from our model are shown below, as comparison between the brightest photo in the original set and the result photo from our HDR model. Our model preserved more details in the highlight, and it avoids the problems of blown-out highlights and flat shadows while brightening the whole image. Noises in shadow and color bias were also largely removed, compared to the original photos in 4 EV.

Results from model. The columns indicate photos from different sets. From left to right: original 4EV, results from model of Chen et al. (2018), our model, respectively

Limitations and Discussion

As shown in the photo below, which is one of our resulting photos, it appears that some regions of the image are constructed poorly by our model. There are also some blur areas in several other photos. Perhaps more epochs or better model would be a solution to this problem.

Orange box indicates a reconstruction failure

Another limitation of the model is that the amplification ratio must be picked externally by hand. It can be an improvement if the model could determine a desired amplification ratio one its own for the input data.

Further, future work could be done to optimize the structure of our codes for improving total runtime. Currently, the model would take at least 24 hours to train on an AWS p2 instance; photo processing also takes a long time, which is not optimal for real-life application.

Reference

C. Chen, Q. Chen, J. Xu, V. Koltun (2018). *Leaning to See in the Dark*. Retrieved from: https://cchen156.github.io/paper/18CVPR_SID.pdf