Learning to See in the Dark

Abhinav Mishra[1] Ajay Kumar Maurya[2] Aniket Kumar Gupta[3] Anita Verma[4] Vivek Shukla[5]
[1,2,3,4,5]Department of Computer Science Engineering
[1,2,3,4,5]Dr. Ambedkar Institute of Technology for Handicapped, Kanpur 208024- India

Abstract— Low light imaging is a challenge of cameras because of low photon counts and low signal to noise ratio. In case of low lighting taking an image becomes more challenging and complex. Generally all camera supports low light imaging but not all images result are good because they mostly uses traditional pipeline. Traditional Pipelining performs various processing such as Histogram Processing or scaling but it does not resolve the low SNR because of low photon counts. And generally short exposure images suffers from noise, while long exposure images may induce blur due to camera shake or Object motion and is often impractical. There are many different types of techniques like denoising, deblurring and image enhancement which are already proposed but all these techniques have a certain limits that the images are take in dim environment but they cannot give better result in extremely low light. In this paper we have proposed a learning based pipeline which uses Artificial intelligence to train his neurons to improve the learning, which can see in extremely low lighting. In this paper we are introducing a dataset of raw short-exposure low light images to correspond with images of long exposure time.

Keywords: SNR, CNN, CAN, Exposure Time, U-Net, Deep Neural Network

I. INTRODUCTION

It is a powerful machine-learning based image processing technique that allows regular cameras to take super-sharp pictures in very low light, without long exposures or the kinds of graininess associated with low-light photography. To try to solve this problem without inventing a new image sensor, researchers at Intel and the University of Illinois Urbana-Champlain taught an artificial intelligence algorithm how to take the data from darker images and reconstruct them so that they're brighter and clearer, according to research published this month and to be presented in June at an industry conference.

Computer Vision is a field in Artificial Intelligence which revolves around getting visual input and either making sense of the input received or manipulating the given input in some way to get the desired output. The model here uses an end to end trained fully-Convolutional Network which makes the use of a dataset of raw short-exposure night-time images, with corresponding long-exposure reference images. This makes obtaining results from extreme scenarios like night photography very easy and efficient as compared to traditional denoising and deblurring techniques.

II. METHODOLOGY

A. *Pipeline:*

After getting the raw data from an imaging sensor, the traditional image processing pipeline applies a sequence of modules such as white balance, demosaicing, denoising, sharpening, color space conversion, gamma correction, and others. These modules are often tuned for specific cameras. Jiang et al. [18] proposed to use a large collection of local, linear, and learned (L3) filters to approximate the complex nonlinear pipelines found in modern consumer imaging systems.

Yet neither the traditional pipeline nor the L3 pipeline successfully deal with fast low-light imaging, as they are not able to handle the extremely low SNR. Hasinoff et al. [14] described a burst imaging pipeline for smartphone cameras. This method can produce good results by aligning and blending multiple images, but introduces a certain level of complexity, for example due to the need for dense correspondence estimation, and may not easily extend to video capture, for example due to the use of lucky imaging.

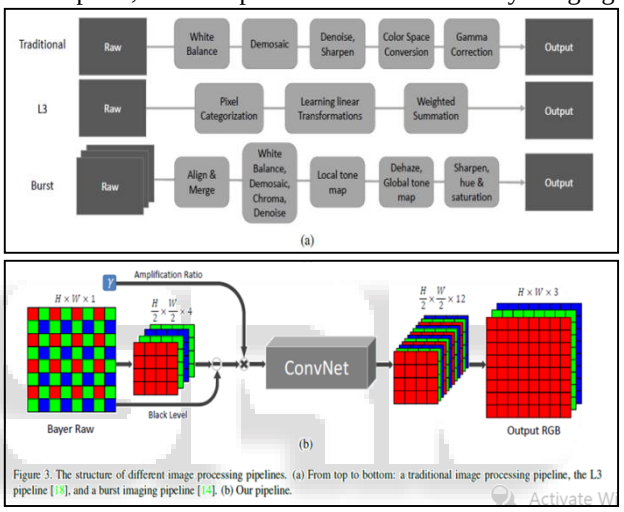

Fig. 1: illustrates the structure of the presented pipeline.

For Bayer arrays, we pack the input into four channels and correspondingly reduce the spatial resolution by a factor of two in each dimension. For X-Trans arrays (not shown in the figure), the raw data is arranged in 6×6 blocks; we pack it into 9 channels instead of 36 channels by exchanging adjacent elements. We subtract the black level and scale the data by the desired amplification ratio (e.g., x100 or x300). The packed and amplified data is fed into a fully-convolutional network. The output is a 12-channel image with half the spatial resolution. This half-sized output is processed by a sub-pixel layer to recover the original resolution [27]. After preliminary exploration, we have focused on two general structures for the fully-convolutional network that forms the core of our pipeline: a multi-scale context aggregation network (CAN) recently used for fast image processing [5] and a U-net [18]. Other work has explored residual connections [20, 24, 11], but we did not find these beneficial in our setting, possibly because our input and output are represented in different color spaces. Another consideration that affected our choice of architectures is memory consumption: we have chosen architectures that can process a full-resolution image (e.g., at 4240×2832 or 6000×4000 resolution) in GPU memory. We have therefore avoided fully-connected layers that require processing small image patches and reassembling them [26]. Our default

architecture is the U-net [15]. The amplification ratio determines the brightness of the output. In our pipeline, the amplification ratio is set externally and is provided as input to the pipeline, akin to the ISO setting in cameras. Figure 4 shows the effect of different amplification user can adjust the brightness of the output image by setting different amplification factors. At test time, the pipeline performs blind noise suppression and color transformation. The network outputs the processed image directly in sRGB space.

B. Current Implementation:

The model here uses an end to end trained fully-Convolutional Network which makes the use of a dataset of raw short-exposure night-time images, with corresponding long-exposure reference images. This makes obtaining results from extreme scenarios like night photography very easy and efficient as compared to traditional denoising and deblurring techniques.

C. The Learning Process:

1) How is the CNN trained?

The CNN is trained on two sets of images.

1) A dimly lit (almost dark) scene or short-exposure picture as an input.
2) A corresponding normal lighting scene or long-exposure picture of the same scene as target.

The neural net is trained on a dataset containing 5094 raw short-exposure images and their corresponding long-exposure images. So if you want to train the network, you will have to first click a photograph under normal lighting conditions which will be used as a target variable to obtain error by the network. Next, you will have to click a low exposure photograph of the same scene so that it looks dark. This will be given as an input to the network while training. The pair of these two photographs will produce an (input, output) pair for the network upon which it will be trained to be used on low-light test images.

Using the L1 loss and Adam optimizer [21], we train the networks. Our aim is we train one network for each camera. Under training, raw data of the short-exposed image and the ground truth is the corresponding long-exposure image in sRGB space (processed by libraw, a raw image processing library) is the input to the network. For both training and testing, the amplification ratio is set to be the exposure difference between the input and reference images (e.g., x150, x240, or x300) .For every iteration, we randomly crop or pick up a 512×512 patch for training and apply random flipping and rotation for data augmentation. After 2000 epochs, the learning rate is reduced 10-2 from the initial set to 10-4. Our Training proceeds near to 4000 epochs.

III. COMPONENTS OF SID MODEL

Computer Vision is a field in Artificial Intelligence which revolves around getting visual input and either making sense of the input received or manipulating the given input in some way to get the desired output. The paper that we are concerned about here works on the second use case. Thus, we need to train the Machine Learning Models by using some training models. The most widely used is Deep Neural Network.

A. Deep Neural Network

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers.[11][2] The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output. For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence the name "deep" networks.

DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network.[11]

Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures, unless they have been evaluated on the same data sets.

DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights", to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network didn't accurately recognize a particular pattern, an algorithm would adjust the weights.[That way the algorithm can make certain parameters more influential, until it determines the correct mathematical manipulation to fully process the data.

B. Convolutional Neural Networks

Similar to how a child learns to recognize objects, we need to show an algorithm millions of pictures before it is be able to generalize the input and make predictions for images it has never seen before. Computers 'see' in a different way than we do. Their world consists of only numbers. Every image can be represented as 2-dimensional arrays of numbers, known as pixels. Bute fact that they perceive images in a different way, doesn't mean we can't train them to recognize patterns, like we do. We just have to think of what an image is in a different way.

To teach an algorithm how to recognize objects in images, we use a specific type of Artificial Neural Network: a Convolutional Neural Network (CNN). Their name stems from one of the most important operations in the network: convolution. Convolution is one of the main building blocks of a CNN. The term convolution refers to the mathematical combination of two functions to produce a third function. It merges two sets of information.

C. U-net Network Structure:

We used U-net network architecture. u-net is convolutional network architecture and used for fast and precise

segmentation of images(raw images). Till now it has outperformed the prior best method.

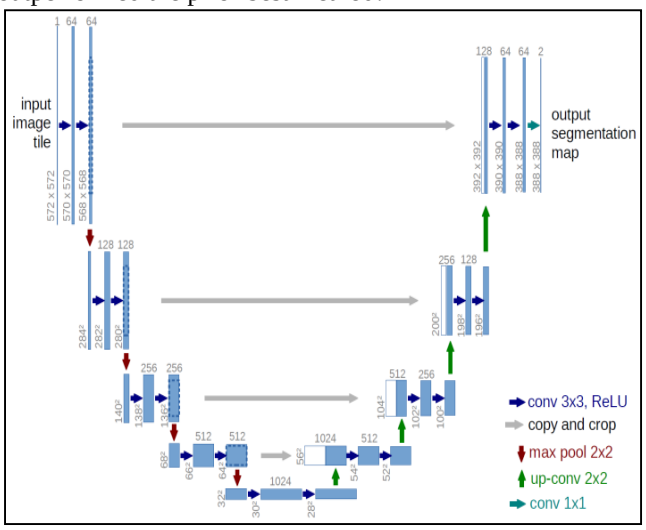

Fig. 2: U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

D. See-in-the-Dark Dataset:

The See-in-the-Dark (SID) dataset contains 5094 raw short exposure images, each with a corresponding long-exposure reference image. We keep multiple short-exposure images correspond to the same long-exposure reference image. For example: we apply the burst denoising methods on collected sequences of short-exposure images. Every image contains real imaging artefacts and useful for training and testing and image in the sequence is counted as a distinct low-light image, here we take 424 distinct long- exposure reference images in SID. Both indoor and outdoor images present in data set. Those images generally captured at night, under moonlight or street called as outdoor.

At the camera, illuminance in the outdoor scenes is generally between 0.2 lux and 5 lux. But indoor images are even darker than the outdoor. In indoor, images were captured in closed rooms with regular lights turned off .In case of the indoor, illuminance at the camera scenes is generally between 0.03 lux and 0.3 lux. The exposure time for the input images (raw images) was set between 1/40 and 1/20 seconds. The corresponding reference (ground truth) images were captured with 100 to 300 times longer exposure: i.e., 10 to 30 seconds. For the reference images, exposure times are necessarily long. Each and every scenes (images) in the dataset are static. The dataset is summarized in Table 1. Figure 2 a small sample is given. Here approximately 10%data set taken randomly from validation set and Figure (a)&(b) validation set and 20% randomly chosen from test sets. There might be possibilities that long-exposure reference images may still contain some noise, but the sagacity quality is very high for these images to serve as raw data(ground truth).So our target is aim to produce good images in low-light conditions or moon light or street light, rather than exhaustively removing all noise .

IV. SOFTWARE IMPLEMENTATION

The objective is to design a software system which is capable of learning the images in low light. Object detection refers to the capability of computer and software systems to locate objects in an image/scene and identify each object. Google's Pixel phones have already changed and improved smart phone photography dramatically, Night Sight is the next evolution of Google's computational photography, combining machine learning, clever algorithms, and up to four seconds of exposure to generate shockingly good low-light images.

A. Hardware Requirement

- Processor Intel i7 CPU
- Ram 128 GB (or above)
- Graphics Nvidia Titan X
- Pointing device Mouse
- Hard disk 256 GB (or above)

B. Software Requirement

Operating System Windows 10, Ubuntu , Required python (version 2.7) , Anaconda Jupyter Notebook.
libraries: Tensorflow (>=1.1) + Scipy + Numpy + Rawpy.

C. Future Scope

As we might have known that iPhone also uses Sony sensors we can build a model to support for iPhone RAW images and then create a mobile application.

D. Training

1) The dataset is split into training data and test data.(1865 IMAGES)
2) These raw images are splitted into color arrays.
3) A Random input patches of size 512*512 are selected and then the augmentation is done.
4) These augmentation patches of images are sent as input into neural network and trained for 400 epochs.
5) The network architecture used for training is U-Net.
6) The loss function used is mean squared error(MSE). – 1/N ((output_ image – label_ image)**2).
7) Optimization Function: Adam's optimizer with learning rate as 0.000.

ACKNOWLEDGEMENT

First of all, I would love to give my regards to the almighty GOD, my parents who are more than GOD for me and all family member for making me a human being and for making me able to make my dream comes true. It gives me an immense feeling of great pleasure and most sincere regards and sense of gratitude to my guide for his brilliant and valuable guidance for completing this project. I am very much grateful to my guide "Mr. Vivek Shukla" for his valuable suggestion, helping hand and guidance experienced to me. I am thankful to my guide for his guidance and the encouragement me for time to time.

REFERENCES

[1] Agostinelli, M. R. Anderson, and H. Lee. Adaptive multicolumn deep neural networks with application to robust image denoising. In NIPS, 2013. 2

[2] J. Anaya and A. Barbu. RENOIR – A dataset for real lowlight image noise reduction. arXiv:1409.8230, 2014. 2
[3] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In CVPR, 2012. 2
[4] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In ICCV, 2017. 6
[5] Q. Chen, J. Xu, and V. Koltun. Fast image processing with fully-convolutional networks. In ICCV, 2017. 4, 7
[6] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 2017. 2
[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Transactions on Image Processing, 16(8), 2007. 2, 5
[8] X. Dong, G. Wang, Y. Pang, W. Li, J. Wen, W. Meng, and Y. Lu. Fast efficient algorithm for enhancement of low lighting video. In IEEE International Conference on Multimedia and Expo, 2011. 2
[9] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing, 15(12), 2006. 2
[10] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand. Deep joint demosaicking and denoising. ACM Transactions on Graphics, 35(6), 2016. 2, 7
[11] Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D.Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014. 7
[12] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In CVPR, 2014. 2
[13] X. Guo, Y. Li, and H. Ling. LIME: Low-light image enhancement via illumination map estimation. IEEE Transactions on Image Processing, 26(2), 2017. 2
[14] S.W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. ACM Transactions on Graphics, 35(6), 2016. 1, 2, 4, 5
[15] K. Hirakawa and T.W. Parks. Joint demosaicing and denoising. IEEE Transactions on Image Processing, 15(8), 2006. 2
[16] Z. Hu, S. Cho, J. Wang, and M.-H. Yang. Deblurring lowlight images with light streaks. In CVPR, 2014. 1
[17] Jain and H. S. Seung. Natural image denoising with convolutional networks. In NIPS, 2008. 2
[18] H. Jiang, Q. Tian, J. E. Farrell, and B. A. Wandell. Learning the image processing pipeline. IEEE Transactions on Image Processing, 26(10), 2017. 4
[19] N. Joshi and M. F. Cohen. Seeing Mt. Rainier: Lucky imaging for multi-image denoising, sharpening, and haze removal. In ICCP, 2010. 2
[20] J. Kim, J. K. Lee, and K. M. Lee. Accurate image superresolution using very deep convolutional networks. In CVPR, 2016. 4
[21] P. Kingma and J. Ba. Adam: A method for stochastic optimization. In ICLR, 2015. 5
[22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4), 1989. 4
[23] C. Liu and W. T. Freeman. A high-quality video denoising algorithm based on reliable motion estimation. In ECCV, 2010. 2
[24] Z. Liu, L. Yuan, X. Tang, M. Uyttendaele, and J. Sun. Fast burst images denoising. ACM Transactions on Graphics, 33(6), 2014. 1, 2, 5
[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015. 4
[26] K. G. Lore, A. Akintayo, and S. Sarkar. LLNet: A deep autoencoder approach to natural low-light image enhancement. Pattern Recognition, 61, 2017. 2, 5
[27] Łoza, D. R. Bull, P. R. Hill, and A. M. Achim. Automatic contrast enhancement of low-light images based on local statistics of wavelet coefficients. Digital Signal Processing, 23(6), 2013. 2
[28] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In ICCV, 2009. 2
[29] H. Malm, M. Oskarsson, E. Warrant, P. Clarberg, J. Hasselgren, and C. Lejdfors. Adaptive enhancement and noise reduction in very low light-level video. In ICCV, 2007. 2
[30] S. Park, S. Yu, B. Moon, S. Ko, and J. Paik. Low-light image enhancement using variational optimization-based Retinex model. IEEE Transactions on Consumer Electronics, 63(2), 2017. 2

