iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VI **Month of publication:** June 2020

DOI: http://doi.org/10.22214/ijraset.2020.6351

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

Heuristic Approach for Low Light Image Enhancement using Deep Learning Technique

Shrutika Ankam[1], Mansi Chougule[2], Ankita Chavan[3], Komal Dharam[4], Prof. Sonali Nalamwar[5]
[1, 2, 3, 4, 5]Department Of Computer Engineering, All India Shri Shivaji Memorial Society's College of Engineering, Pune, Maharashtra, India

***Abstract:** Due to low photon count and low Signal to Noise Ratio (SNR), low light imaging becomes more challenging. Images taken in a short exposure get affected by noise while images taken in a long exposure can be blurry. Different methods like image denoising, deblurring, and image enhancement are existing, but at extreme conditions their effectiveness is limited. A dataset that includes raw short exposure low light images and corresponding long exposure images is used for the development of a learning-based pipeline. DNN based approach operates on raw sensor data and works effectively. It outperforms the traditional image processing pipeline which shows poor results on such raw sensor data.*
***Keywords:** SNR, U-net, Black level, CNN, etc.*

I. INTRODUCTION

Imaging becomes more challenging in low light. High ISO is most commonly used for increasing brightness, but it also amplifies noise. Many post-processing techniques, such as scaling or histogram stretching, can be applied to reduce noise, but as the photon count is low, this does not resolve the low signal-to-noise ratio (SNR). There are many other alternative ways to increase SNR in low light, like opening the aperture, extending exposure time, and using flash. But each technique has its own drawbacks. For example, it can introduce blur or object motion, if the exposure time is increased.

Different techniques have been proposed by many researchers for deblurring, denoising, and enhancement of low-light images.[1,2,3] These techniques generally assume that images are captured with moderate levels of noise in dim environments. In extreme low-light imaging, the traditional camera processing pipeline fails and the image needs to be constructed again from the raw sensor data. When the environment is extremely dark with very little illumination at the camera. The exposure time is set to 1/30 second and the aperture is set to f/5.6. At high ISO ie. ISO 8,000, which is normally considered high, the camera produces an essentially black image, despite the high light sensitivity of the full-frame Sony sensor. The content of the scene is visible at extremely high ISO i.e. ISO 409,600, which is considered to be far beyond the reach of most of the cameras. But the image produced is dim, noisy, and even the colors are distorted. Even state-of-the-art denoising techniques [4] fail to eliminate such noise and do not address the color bias. Using a burst of images is an alternative approach [5,6], but in extreme low-light conditions burst alignment algorithms may fail.

The challenges of extreme low-light photography can be addressed by an image processing pipeline using a data-driven approach. More precisely, we will train deep neural networks to learn the image processing pipeline for raw data of low light images, which includes color transformations, image enhancement, noise reduction, and demosaicing. To avoid amplification of noise and error accumulation, the pipeline will be trained end-to-end.

II. OBJECTIVE

The convolutional neural network is trained to learn the image processing pipeline for low light raw data which includes methods like color transformations, demosaicing, noise reduction, and image enhancement. Different amplification ratios can be externally set to adjust the brightness level of the output image.

III. IMPLEMENTATION

A. Dataset

In this project, we used SID (See In The Dark) dataset[8]. It consists of total 211 raw short exposure images captured in Sony camera, each with corresponding long-exposure reference images. The exposure time of the short exposure (input) images are between 1/30 and 1/10 seconds(0.033s to 0.1s) and exposure time of the corresponding reference images is between 10 and 30 seconds. The resolution of each image is 4240 X 2832. For training, we used 161 images and for testing 50 images were used.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

B. Preprocessing

Before training, data preprocessing is important. On the short exposure images, we performed Black level reduction, amplification, and cropped the image by random flipping and transpose. We subtracted black level from the input image and packed the Bayer image to 4 channels. Also scaled the data by amplification ratio (e.g., 100, 250, or 300). Amplification ratio determines the brightness of the image. We cropped the image randomly into 512 X 512 patch size for training and applied random flipping and rotation for data augmentation.

The long exposure images were converted into a 16-bit linear image, we used the automatic white balance on the image as shot by the camera, and normalized the pixel values of the image. This long-exposure image is later fed to the network as a reference image for training the model.

C. Model Design

1) *U-net Architecture:* The U-net architecture[7] is mostly used for semantic segmentation. It consists of two paths. The first path is the contraction path which is also called the encoder and is used to capture the context in the image. The encoder comprises a series of convolutional and max-pooling layers. The second path is the expansion path which is also called as the decoder and is used to enable precise localization using transposed convolutions. Therefore it is an end-to-end fully convolutional network (FCN), as it only contains Convolutional layers and does not contain any Dense layer because of which it can accept images of any size. During the downsampling, the model learns what features are present in the image but loses the information about precise localization. This lost information is gained back during upsampling, wherein the model restores the information about the localization factors.

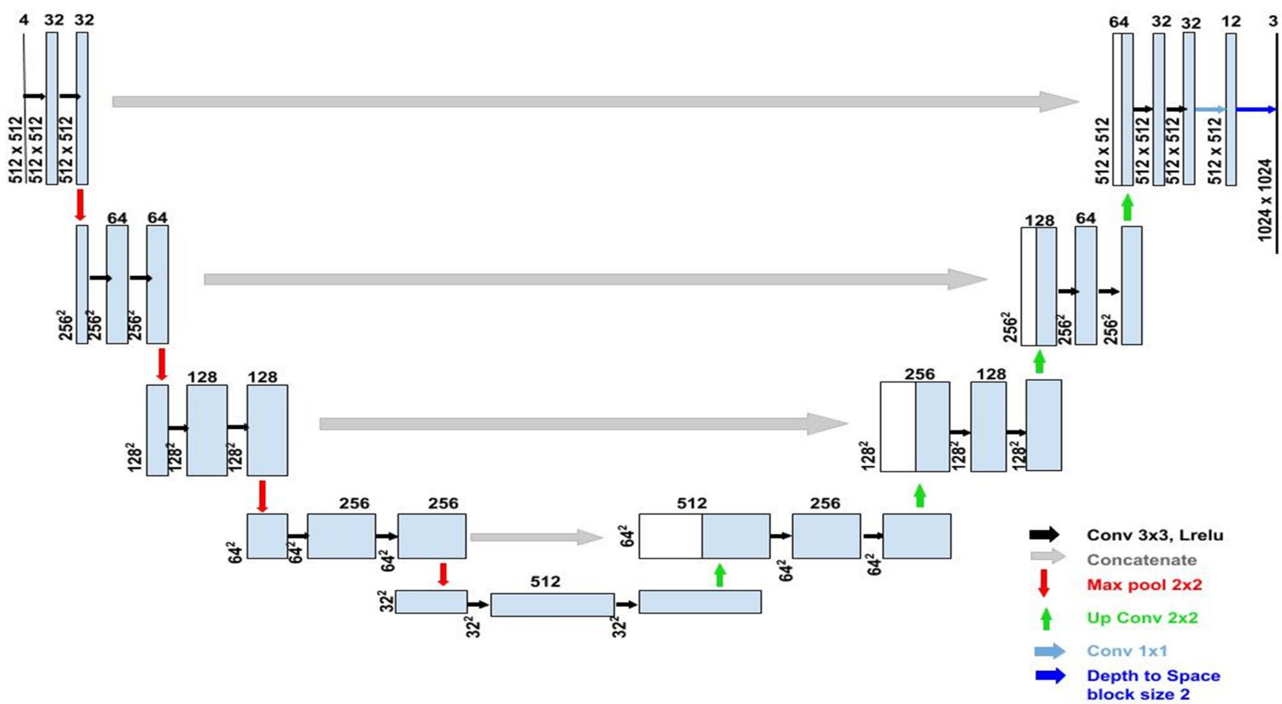


Fig. 1 U-net Architecture

As shown in figure 1. the preprocessed image patch of size 512 x512 is fed as input to the network. During downsampling at every level, there are two consecutive convolution layers followed by a max-pooling layer.

The convolution operation is performed by a 3x3 kernel, with the same padding followed by lrelu (Leaky Relu). And the max pooling operation is done by using a 2x2 kernel with the same padding and strides the same as kernel size. So the output of the max-pooling operation is half the size.

The left side is the contraction path and the right side is an expansion path. In the expansion, the network performs transposed convolution. The output of deconvolution is then concatenated with the corresponding layer from the contracting path. After this operation, the image gains information about the localization of features that were extracted during contraction. The number of feature channels is halved by this operation. This operation is again followed by 2 convolution layers.
At the end of the last convolution, the output is of the shape 512x512x12 which is then rearranged into blocks of spatial data to recover the original resolution.

2) Mathematical Model

Some important formulae related to the model are as follows:

a) W x H x D : Input volume size
F : Filter size
S : Stride
P : Padding value
K : No. of filters
i). Volume size of the output of convolution layer = W_1 x H_1 x D_1
Where, $W_1 = (W - F + 2P)/S + 1$

$H_1 = (H - F + 2P)/S + 1$

$D_1 = K$

ii). Volume size of the output of pooling layer = W_1 x H_1 x D_1
Where, $W_1 = (W - F)/S + 1$

$H_1 = (H - F)/S + 1$

$D_1 = D$

b) Activation function LRelu, y = max(x*0.2, x)

c) L1 loss = $\Sigma|Ytrue - Ypredicted|$

d) Adam optimizer $w_t = w_{t-1} - \frac{\eta}{\sqrt{v_t+\epsilon}} m_t$

$m_t = \beta_1 * v_{t-1} + (1-\beta_1)(\Delta w_t)$, $v_t = \beta_2 * v_{t-1} + (1-\beta_2)(\Delta w_t)^2$

$m_t = \frac{m_t}{1-\beta_1^t}$, $v_t = \frac{v_t}{1-\beta_2^t}$} this is for bias correction

Where, w : weight
η : learning rate
m : 1st moment estimate
v : 2nd moment estimate
ϵ: adaptive learning rate
β_1: exponential decay rate for m
β_2: exponential decay rate for v
t : timestep

D. Training

We trained the network using u-net architecture[7] with the L1 loss function and adam optimizer[9]. Adam optimizer[9] is used to change the hyperparameters of the network (i.e. weights, bias, and learning rate) such that current loss is always less than previous loss. The activation function used is LRelu (Leaky Rectified Linear Unit). Input to the network is a raw short-exposure image and a corresponding long exposure image. The network is trained after the preprocessing. Every input image is cropped into a 512 x 512 patch size. Initially, the learning rate is set to 10^-4 and is reduced to 10^-5 after 2000 epochs. A total of 4000 epochs are performed. Tensorflow is used for all the computations.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VI June 2020- Available at www.ijraset.com

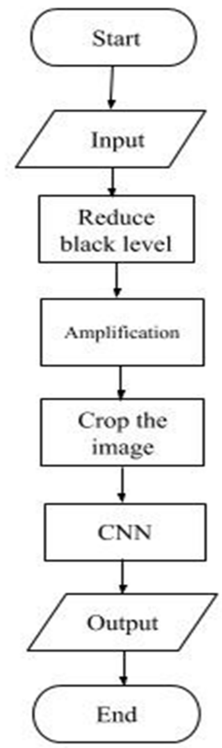


Figure 2. Flowchart

E. Testing

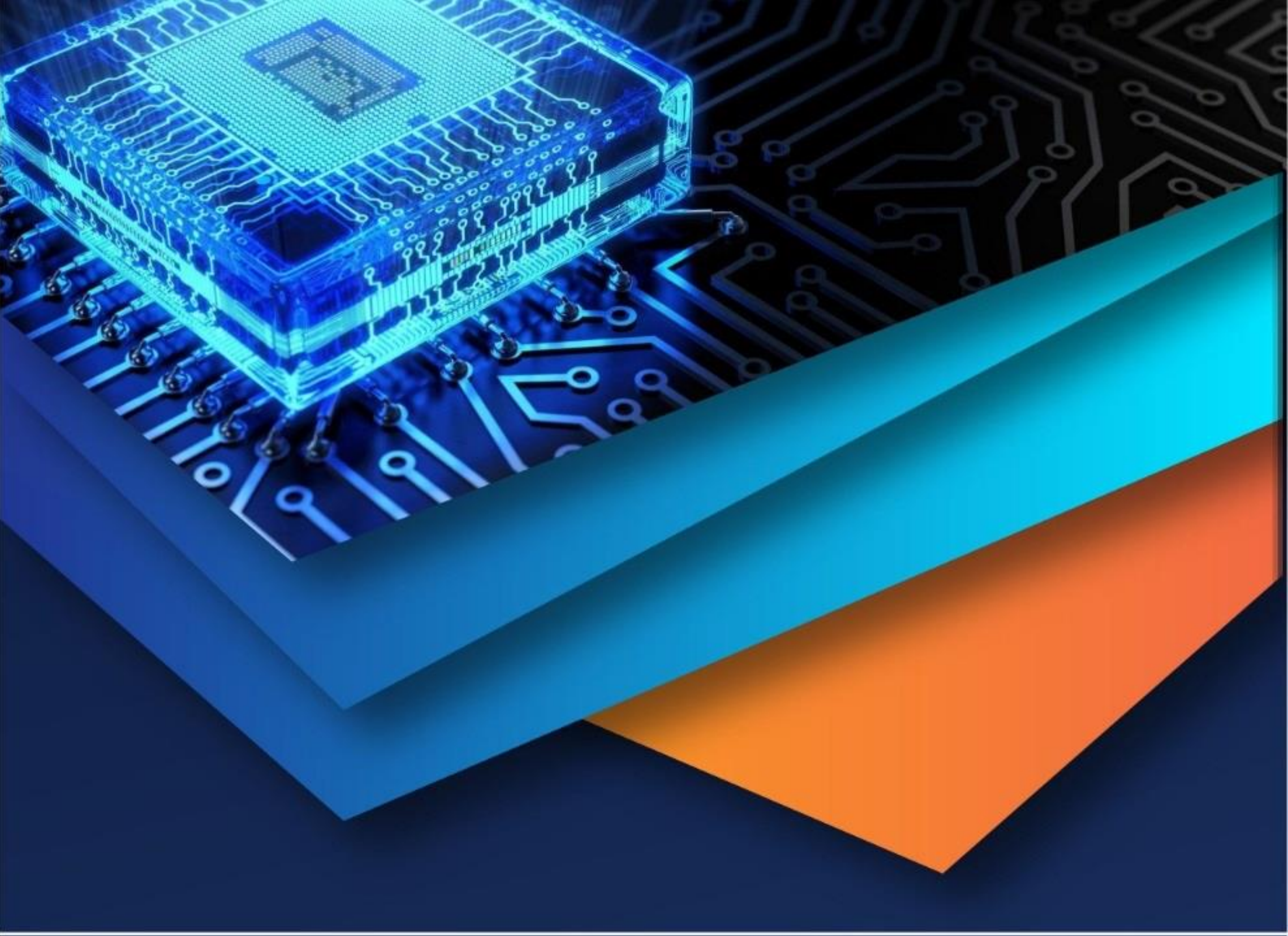
The trained model was tested on 50 raw images from the SID test set. The loss was calculated using the L1 Loss function. The loss of the model was calculated to be approximately 3 0.5%. And the accuracy turns out to be approximately 97 0.5%. The trained model was deployed on a website developed using the flask framework of python. Flask is a lightweight WSGI web application framework. Front-end of the application is based on HTML and CSS. And the backend is supported by the flask.

IV. RESULTS

A data-driven heuristic approach to develop a model that is used for extreme low light imaging purposes was used. A pipeline based on end-to-end training of a Fully Convolutional network was implemented that improves upon traditional processing of low light images. The low light raw images from the See In The Dark dataset were enhanced in a way that the problems faced during low light imaging such as high noise, incorrect color generation, and induced blur were successfully tackled. The system can enhance the quality of the low light images to a high extent as opposed to the traditional processing pipeline. The output images were generated with successful noise suppression and correct color transformation on SID data.

REFERENCES

[1] 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.
[2] X. Zhang, P. Shen, L. Luo, L. Zhang, and J. Song. Enhancement and noise reduction of very low light level images. In ICPR, 2012.
[3] T. Remez, O. Litany, R. Giryes, and A. M. Bronstein. Deep convolutional denoising of low-light images.arXiv:1701.01687, 2017.
[4] T. Plötz and S. Roth. Benchmarking denoising algorithms with real photographs. In CVPR, 2017.
[5] 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
[6] Z. Liu, L. Yuan, X. Tang, M. Uyttendaele, and J. Sun. Fast burst images denoising. ACM Transactions on Graphics,33(6), 2014.
[7] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In MIC-CAI, 2015.
[8] Chen Chen, Qifeng Chen, Jia Xu, Vladlen Koltun. Learning To See In The Dark. In CVPR,2018.
[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In ICLR, 2015.



10.22214/IJRASET

45.98

IMPACT FACTOR:
7.129

IMPACT FACTOR:
7.429