Low Light Noise Removal using CNN

Project report for IITB EE610 Image Processing 2021

Harsh Pal
18D070012[at]iitb.ac.in
Dept. of Electrical Engg.
IIT, Bombay
Mumbai, India

Anurag Kumar
18D070006[at]iitb.ac.in
Dept. of Electrical Engg.
IIT, Bombay
Mumbai, India

Advait Vinai Kumar
18D070003[at]iitb.ac.in
Dept. of Electrical Engg.
IIT, Bombay
Mumbai, India

Vinit Awale
18D070067[at]iitb.ac.in
Dept. of Electrical Engg.
IIT, Bombay
Mumbai, India

***Abstract*—With the aim of restoring high-quality image content from its degraded version, image restoration has numerous applications. Lately, convolutional neural networks (CNNs) have achieved dramatic improvements over conventional approaches for image restoration task. Existing CNN-based methods typically operate either on full-resolution or on progressively low-resolution representations. In the former case, spatially precise but contextually less robust results are achieved, while in the latter case, semantically reliable but spatially less accurate outputs are generated. In this project, we study an architecture with the collective goals of maintaining spatially precise high-resolution representations through the entire network, and receiving strong contextual information from the low-resolution representations. We understand the functioning of the architecture and suggest modifications which can be done to further improve the performance depending on various use-cases.**

***Index Terms*—Low Light Image Enhancement, MIRNET, Image denoising**

I. INTRODUCTION

The number of images produced in recent times is growing humongously due to presence of cameras everywhere on various devices. During image acquisition, degradation of varying severity often gets captured along with. It happens due to various reasons,some of them being physical limitations of cameras or inappropriate lighting conditions and surrounding environment. The art of recovering the original clean image from its corrupted form is done under the image restoration task.

In past decade, deep learning models have made revolutionary advancements for image restoration and enhancement, as they can learn strong and generalizable priors from large-scale data sources. Presence of huge amount of data which is growing exponentially and the increased computational capability has made this possible.

Existing CNN models generally follow one out of the two design frameworks: an encoder-decoder or a high-resolution , single-scale feature extraction followed by processing. The encoder-decoder models first sequentially map the input to a low-resolution representation, and then apply a gradual reverse mapping to the original resolution. Although these approaches learn a broad context by spatial-resolution reduction, the fine spatial details are often ignored, making it hard to restore them in the later stages. On the other side, the high-resolution, single-scale networks do not use any downsampling operation, and thus produce images with spatially more accurate details,but less effective in encoding contextual information due to their limited contextual field.

Image restoration is a position-sensitive process, where pixel-to-pixel correspondence from the input to the output is needed. Therefore, it is important to remove only the undesired degraded image content, while carefully preserving the desired fine spatial details like true edges and texture. Such functionality for segregating the degraded content from the true signal can be better incorporated into CNNs with the help of large context, e.g., by enlarging the receptive field.

Towards this goal, we explore a new multi-scale approach that maintains the original high-resolution features along the network hierarchy, thus minimizing the loss of precise spatial details. The multi-resolution parallel branches operate in a method which is complementary to the main high-resolution branch, thus providing us more precise and contextually enriched feature representations. Comprehensive experiments are performed on two real image benchmark datasets for different image processing tasks including image denoising, super-resolution and image enhancement, various tweaks and modifications are made in the standard structure to explore the possibility of improvement. Various Regularisation, types of Normalisation and Optimization approaches are tested and outcomes are recorded so that it can be modified in future as per the use-casew to achieve the desired task.

II. BACKGROUND AND PREVIOUS WORK

With the rapid growth of image content, there is a need to design effective image restoration and enhancement algorithms. The approach used in this project processes features at the original resolution in order to preserve spatial details, while effectively fuses contextual information from multiple parallel branches. Next, we briefly describe the representative methods for each of the studied problems:

A. Image Denoising :

Classic denoising methods are often based on modifying transform coefficients or averaging neighborhood pixels. Many patch-based algorithms that exploit redundancy in images are later developed. Recently, deep learning based approaches make significant advances in image denoising, yielding favorable results than those of the hand-crafted methods.

B. Super-resolution (SR)

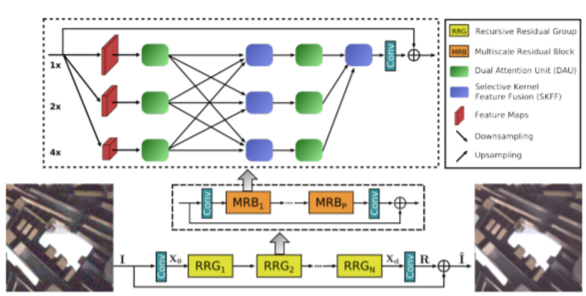
Currently, deep-learning techniques are actively being explored, as they give improved results over conventional algorithms. The data-driven SR approaches differ according to their architecture designs . In contrast to directly producing a latent HR image, recent SR networks employ the residual learning framework to learn the high-frequency image detail, which is later added to the input LR image to produce the final resolved result. Other networks designed to perform SR include dense connections, attention mechanisms , multi-branch learning , and generative adversarial networks (GANs).

C. Image Enhancement

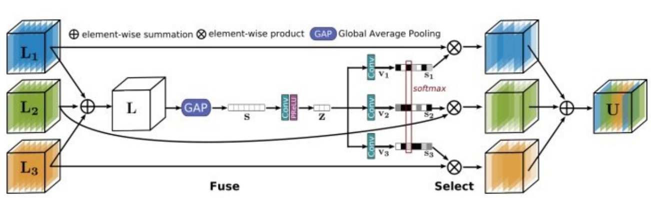
Often, cameras generate images that are less vivid with poor contrast. For image enhancement, histogram equalization is the most commonly used method. However, it frequently produces under- or over-enhanced images. Recently, CNNs have been successfully applied to general, as well as low-light, image enhancement problems. Some popular CNN methods employ Retinex-inspired networks, encoder-decoder networks , and GANs.

III. Data and Methodology

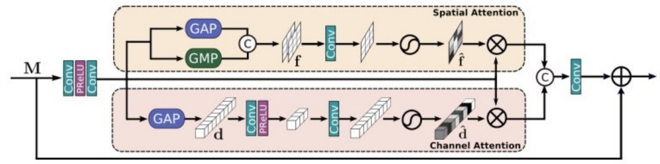
We used LOL Dataset for our experiments.The dataset is composed of 500 low-light and normal-light image pairs and divided into 485 training pairs and 15 testing pairs. Out of 485, we used 400 pairs for training and 85 for validation.
We used random cropping for data-augmentation and min-max standardization for pre-processing.



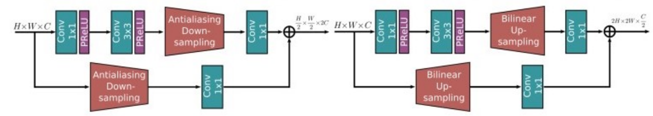
MIRNet Network, Source: References[1]



Selective Kernel Feature Fusion(SKFF), Source: References[1]



Dual Attention Unit, Source: References[1]



Upsampling and Downsampling in SKFF module, Source: References[1]

We use the standard MIR Net model as the broiler plate model for our experimentation. Due to the complexity of MIRNet and limited GPU usage in google colab we used 32 channels in each spatial scale in the MRB block instead of 64. We also reduced the original image size of $128 * 128$ to $64 * 64$ which decreases the training time significantly.

Pipeline Architecture Given an image $\mathbf{I} \in \Re^{H*W*3}$, the network first applies a convolutional layer to extract low-level features $X_o \in \Re^{H*W*C}$. Then, the feature maps X_o pass through N number of recursive residual groups, yielding deep features $X_d \in \Re^{H*W*C}$. We note that each RRG contains several multi-scale residual blocks. Next, we apply a convolution layer to deep features X_d and obtain a residual image $\mathbf{R} \in \Re^{H*W*3}$. Finally, the restored image is obtained as $I^{'} = I + R$.
In the primary stage we optimize the proposed network using the Charbonnier loss :

$$L(I^{'}, I^{*}) = \sqrt{||I^{'} - I^{*}||^2 + \epsilon^2}$$

where I^* denotes the ground-truth image, and ϵ is a constant which we empirically set to 10^{-3} for all the experiments.

In further attempts towards improvement we also try to analyse how SSIM loss optimization performs, SSIM loss function is given by :

$$SSIM(I^{'}, I^{*}) = S = [\frac{2\mu_1\mu_2}{\mu_1^2+\mu_2^2}] * [\frac{2\sigma_1\sigma_2}{\sigma_1^2+\sigma_2^2}] * [\frac{\sigma_{12}}{\sigma_1\sigma_2}]$$

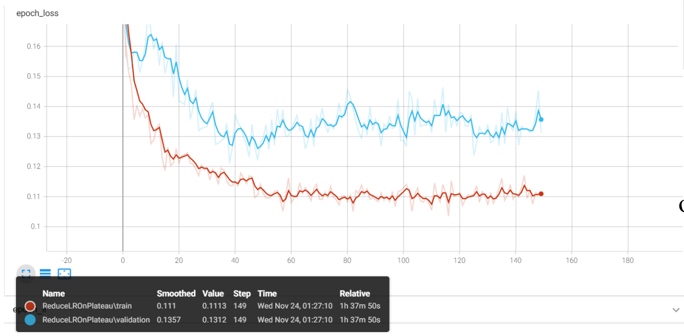
$$S = [\frac{2\mu_1\mu_2}{\mu_1^2+\mu_2^2}] * [\frac{2\sigma_{12}}{\sigma_1^2+\sigma_2^2}] = S_L S_V$$

Here , S_L contains local luminance and the second S_V local covariance.

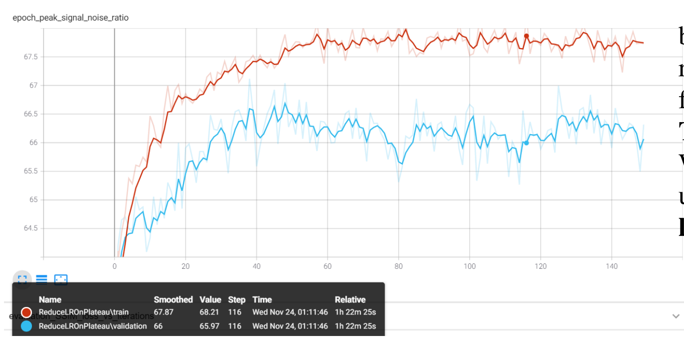
Multi-scale residual block, which is the fundamental building block of our method, containing several key elements: (a) parallel multi-resolution convolution streams for extracting (fine-to-coarse) semantically-richer and (coarse-tofine) spatially-precise feature representations, (b) information exchange across multi-resolution streams, (c) attention-based aggregation of features arriving from multiple streams, (d) dual-attention units to capture contextual information in both spatial and channel dimensions, and (e) residual resizing modules to perform downsampling and upsampling operations.

IV. Experiments and Results

First, we trained the original MIRNET architecture from scratch upto 150 epochs with batch size of 4. For this, the observed plot of Loss vs Epochs on the training and the validation dataset was as follows:



Loss vs Epochs for Modified MIRNET



PSNR vs Epochs for Modified MIRNET

A. *Results Obtained by the Original Architecture*

Fig. 1. Example of the corrected image

Fig. 2. Example of the corrected image

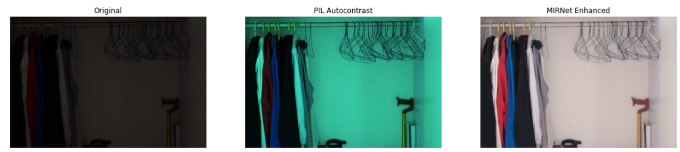


Fig. 3. Example of the corrected image

Fig. 4. Example of the corrected image

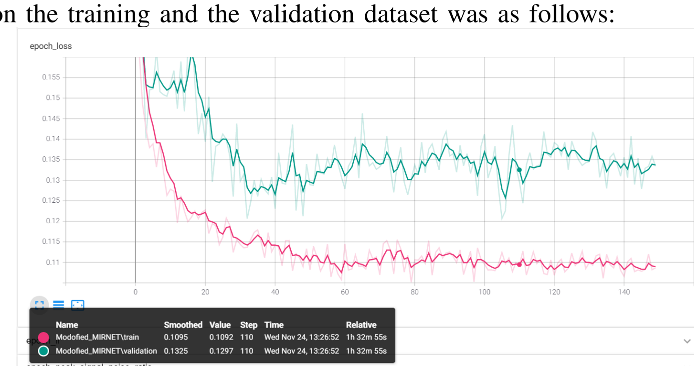
We performed total 3 experiments by modifying the setup of original MIRNET architecture

1) Modified model with Charbonnier loss
2) Modified model with L1 and L2 regularization and Charbonnier loss
3) Modified model with L1 and L2 regularization and SSIM(Structural Similarity Index) loss

For each experiments we trained upto 160 epochs with batch size of 4. We experimented with different Learning rate schedulers. **ReduceLROnPlateau** was observed to give faster convergence than **Exponential decay**, **Step decay** and **Time-based decay**.
We used Adam optimizer with learning rate of 0.0001. We used two loss function for different experiments, **Charbonnier loss** and **SSIM loss**(*using SSIM metric as loss function*)

B. *Modified (4 spatial scales) MIRNET with Charbonnier Loss*

For this experiment, the observed plot of Loss vs Epochs on the training and the validation dataset was as follows:



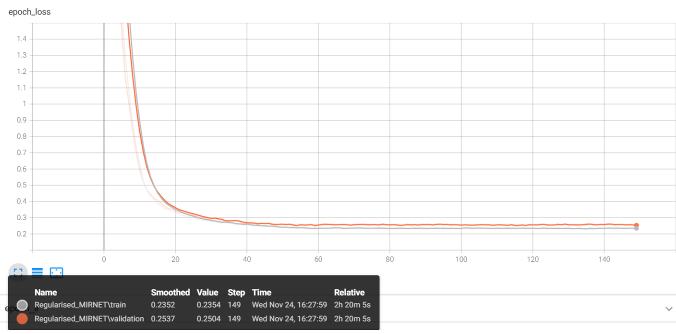


Fig. 5. Loss vs Epochs for Modified MIRNET with Regualrization

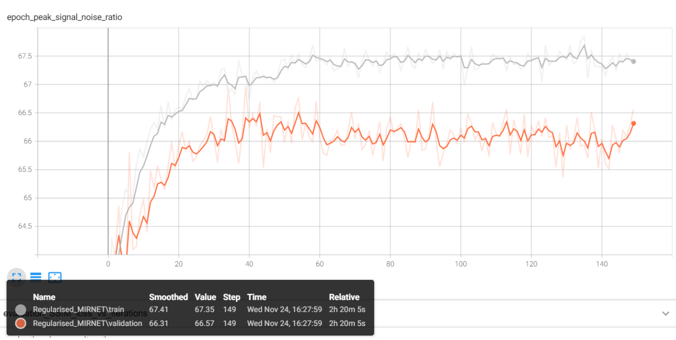
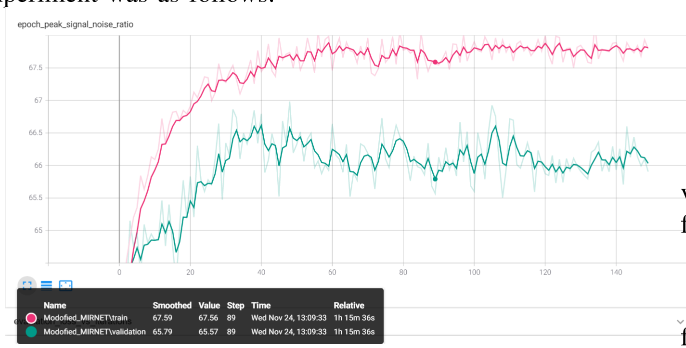


Fig. 6. PSNR vs Epochs for Modified MIRNET with Regualrization

Loss vs Epochs for Modified MIRNET

We have used PSNR (Peak Signal to Noise Ratio) as metric in all of the experiments. The plot of PSNR vs epochs for this experiment was as follows:



PSNR vs Epochs for Modified MIRNET

From these plots we clearly see that the model is overfitting. Hence we decided to add regularization to the weights in MRB layer.

1) Modified MIRNET with L1 and L2 Regularization with Charbonnier Loss: For this experiment, the observed plot of Loss vs Epochs on the training and the validation dataset was as follows:

The plot of PSNR vs epochs for this experiment was as follows:

2) Modified MIRNET with L1 and L2 Regularization with SSIM Loss: For this experiment, the observed plot of Loss

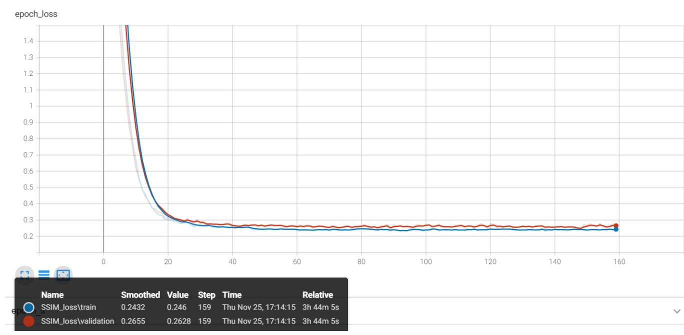


Fig. 7. Loss vs Epochs for Modified MIRNET with Regualrization and using SSIM Loss

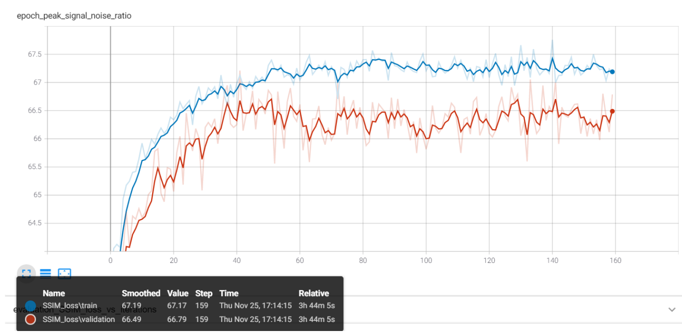


Fig. 8. PSNR vs Epochs for Modified MIRNET with Regualrization and using SSIM Loss

vs Epochs on the training and the validation dataset was as follows:

The plot of PSNR vs epochs for this experiment was as follows:

Experiment	**SSIM**	**PSNR**	**Charbonnier Loss**
Original Model(3 spatial scales)	0.1878	66.3162	**0.1312**
Modified(4 spatial scales)	0.2053	67.7935	0.1334
Modified+L1 and L2	0.1696	66.5710	0.1404
Modified+L1 and L2 +SSIM	**0.2158**	**67.7935**	**0.1304**

.

Table 1

3) Visualising the Results Obtained: Now, we compare the results obtained by training the model in the above mentioned experiments. The resulting images are as follows:

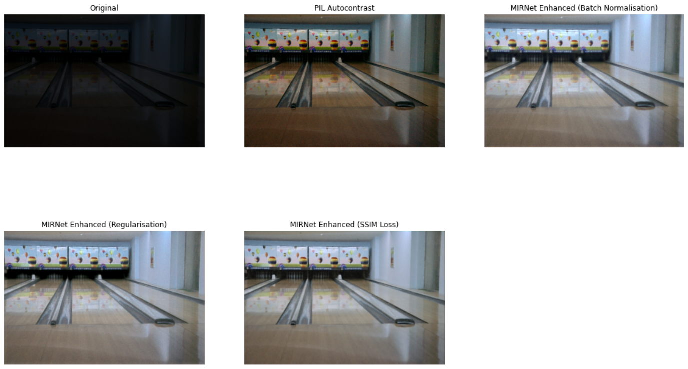


Fig. 9. Example of the corrected image

Fig. 10. Example of the corrected image

Fig. 11. Example of the corrected image

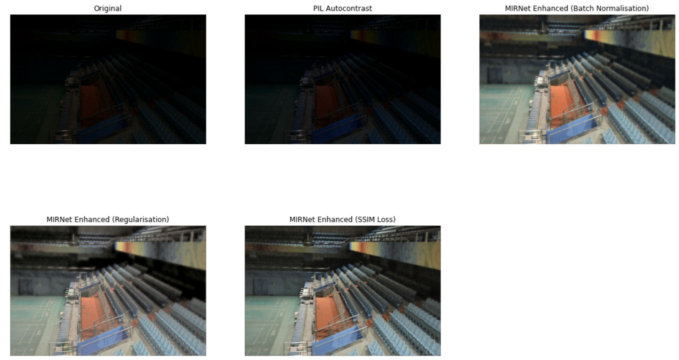


Fig. 12. Example of the corrected image

V. Learning, Conclusions, and Future Work

A. *Learning*

Image-enhancement is one of the use-cases where batch normalization gives negative results as visible in the inference results. SSIM metric can be used as a loss function and it gives better results better PSNR value than it alternative Charnnobier loss. L1 and L2 regularization gives significant improvement in the convergence of the model. This can be observed in the Loss vs Epoch curve. Further, in PSNR vs Epoch, the gap between training and validation is also reduced.

B. *Conclusion*

The experiment with modified model, SSIM loss and regularization gives the best result. Hence SSIM loss function selection was one of the essential output of this project. The approach of different spatial scales in MRB unit helps retain precise spatial details however negative affect the contextualised representations. MIRNet enables to learn relationhip between features within each branch of network as well as across multiple scale branches, unlike prior models which learns across multiple branches only. The choice of number of spatial scales, however, does not affect much atleast within 150 epochs.

C. *Future Work*

We can implement **Weight Normalization** in the bottleneck MRB units for better convergence, since **Batch normalization** failed to give positve results in this use-case. We can also use a variation of the Haar Wavelets (Wavelet Transform) for better **multiresolution analysis**. We can use custom loss functions with **attention parameters** of different spatial scales as we know that different types of images have different proportion of high level and low level features, corresponding to different scales. We can perform numerous experiment to find the optimal number of MRB units, number of spatial scales and other model architecture hyperparameters to find the best architecture.

Contribution of Team Members

Harsh Pal: Model modifications and debugging, loss function analysis and Regularization method selections

Anurag Kumar: Hyperparameter tuning, loss function analysis and report preparation

Advait Kumar: Hyperparameter tuning, loss function analysis and report preparation

Vinit Awale: Loss function analysis and Hyperparameter tuning. Responsible for running codes for every experiment.

REFERENCES

[1] Learning Enriched Features for Real Image Restoration and Enhancement, arXiv:2003.06792v2, [cs.CV] 8 Jul 2020
[2] On the Effects of Batch and Weight Normalization in Generative Adversarial Networks, arXiv:1704.03971v4 [stat.ML] 4 Dec 2017
[3] Loss Functions for Image Restoration with Neural Networks, arXiv:1511.08861v3 [cs.CV] 20 Apr 2018
[4] Structural Similarity Index SSIMplified, arXiv:1503.06680 [cs.CV] May 2015
[5] https://keras.io/examples/vision/mirnet/