Image Enhancement For Unconstrained Environments

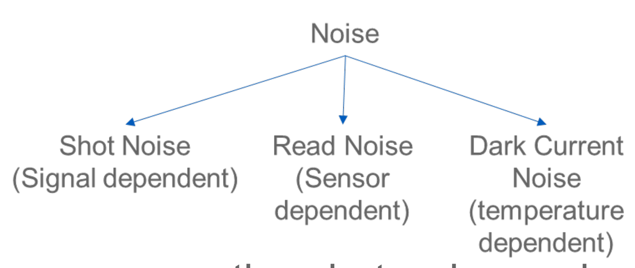
DNNs for Image denoising and exposure adjustment

Sougato Bagchi, MS; Nalini Ratha, PhD

Introduction

Our goal is to use DNN models to improve the quality of images that are dark and/or noisy, which are hard to use for applications like surveillance, autonomous driving, vision-based mapping, and others. We also aim to generate synthetic data that can help train the models better.

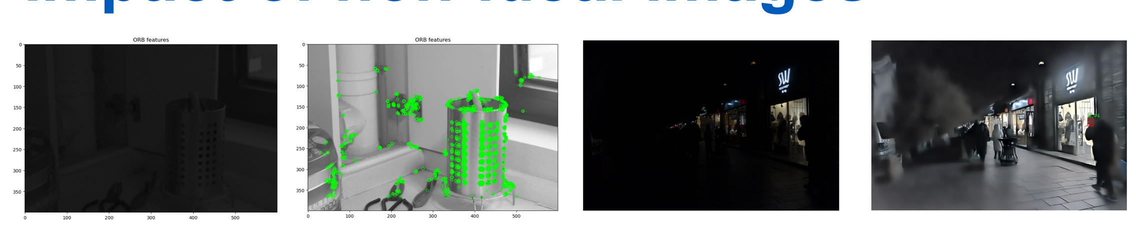
Types of Noise



For our purpose, the shot noise and read noise are deemed relevant where the latter is kept constant for simplification.

Impact of non-ideal images

A



(a) No feature detected in low light conditions (b) Features detected in bright light conditions (a) face detection algorithm completely failed (b) face detection algorithm able to detect at least 1 face

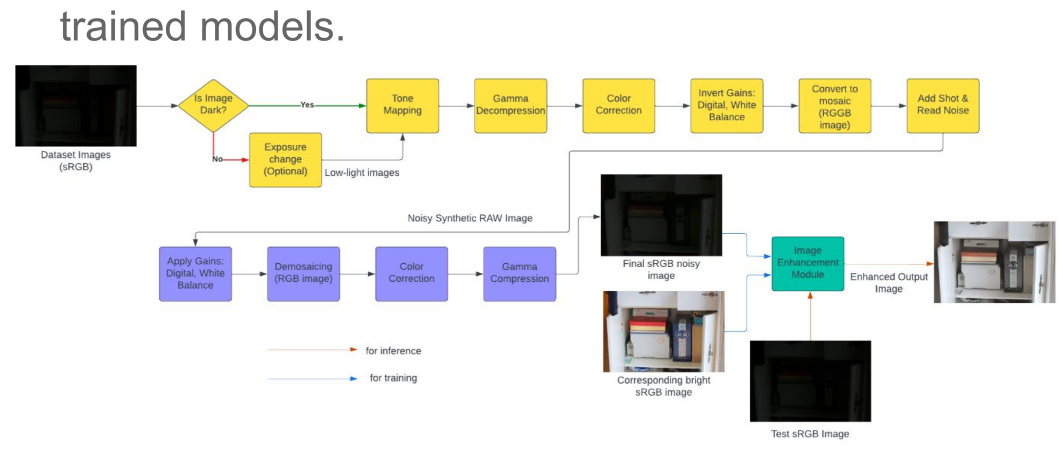
Most computer vision and image-processing applications are severely affected due to the poor quality of images.

Proposed Approach

Our architecture is divided into 2 main modules. The first one is to create synthetic image pairs and the second one is the neural network that trains on the first module's output.

- The first module is based on Brooks et al. [1]. It transforms the sRGB images from the dataset into their raw versions followed by adding noise and reverses the process to get the sRGB images with noise.
- The second module is the LLFlow [2] DNN network trained with paired data. Once the model is trained it is used for inferencing. We created different versions of the model by training them on different datasets, which was created by changing some of the steps in our first module. Then we compared the performance of our versions with the pre-trained models.

B



Our proposed architecture, with the yellow-violet boxes belonging to the un-processing module and the latter green module, is the ESRGAN-based [7] network.

Datasets

LOL

(500 image pairs, public dataset)

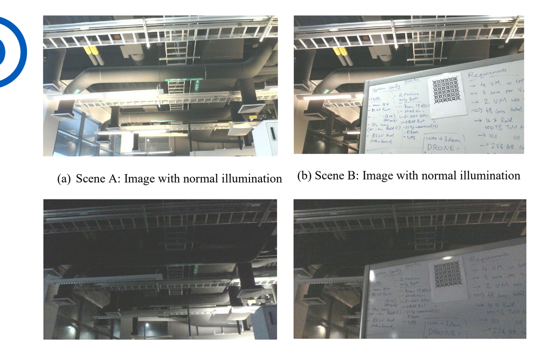
C

(a) Scene A: Image with normal exposure (b) Scene B: Image with normal exposure (c) Scene A: Image with low exposure (d) Scene B: Image with low exposure

PEPPER

(12 image pairs captured in our lab)

D



(a) Scene A: Image with normal illumination (b) Scene B: Image with normal illumination (c) Scene A: Image with low illumination (d) Scene B: Image with low illumination

VE-LOL-H [5]

(10,940 images, human face annotated)

E

(a) (b)

EarthCam [6]

(NYC skyline 2023)

F

(a) June 5th (b) June 6th 11:30pm (a) June 7th 12pm (b) June 7th 2pm

Experiments

- All models trained on the LOL dataset [3] (485 train – 15 validation)
- RAW means that we added shot and read noise to the raw versions of the dataset images, and then converted them back to sRGB format.
- no_ccm_wb_gain means that we added noise without using the color correction matrix, white balance, and digital gain steps.
- pre-trained means that it is the model that the authors originally trained on the unmodified LOL dataset.
- V2 means that we modified both the training and validation datasets in the same way.
- V3 means that we trained on the modified dataset but validated on the 15 unmodified LOL images for evaluation. It was used to make sure that our network does not overfit our modifications

Model type	PSNR (↑)	SSIM (↑)	LPIPS [4] (↓)
custom v2_RAW_noisy	μ = 20.21 σ = 2.640 σ^2 = 6.973	μ = 0.70 σ = 0.105 σ^2 = 0.011	μ = 0.37 σ = 0.081 σ^2 = 0.006
custom v3_RAW_noisy	μ = 20.43 **σ = 1.128** **σ^2 = 1.27**	**μ = 0.743** **σ = 0.072** **σ^2 = 0.005**	μ = 0.406 **σ = 0.070** **σ^2 = 0.005**
pretrained	**μ = 20.71** σ = 2.909 σ^2 = 8.46	μ = 0.70 σ = 0.182 σ^2 = 0.033	**μ = 0.496** σ = 0.311 σ^2 = 0.096
custom v2_ no_ccm_wb_gain	μ = 19.45 σ = 3.294 σ^2 = 10.851	μ = 0.71 σ = 0.096 σ^2 = 0.009	μ = 0.36 σ = 0.076 σ^2 = 0.006
custom v3_no_ccm_wb_gain	μ = 19.46 σ = 1.914 σ^2 = 3.666	μ = 0.72 σ = 0.052 σ^2 = 0.003	μ = 0.41 σ = 0.066 σ^2 = 0.004

Table 1: Based on the observations from this table we can say that the custom v3_RAW_noisy consistently performs better.

Model type	Eval dataset	PSNR (↑)	SSIM (↑)	LPIPS (↓)
custom v3_raw_noisy	LOL	19.23	**0.78**	**0.34**
custom v3_LOL_AWGN	LOL	**20.16**	0.77	0.45

Table 2: Comparing the performance of the v3 model trained in additive white Gaussian noise vs read + shot noise

Model type	Eval data	PSNR (↑)	SSIM (↑)	LPIPS (↓)
custom v3_raw_noisy	Pepper	**13.56**	0.53	**0.49**
pretrained	Pepper	12.45	**0.54**	0.57
custom v2_ no_ccm_wb_gain	Pepper	13.04	0.44	0.52
custom v3_RAW_noisy	EarthCam	11.71	0.53	**0.68**
pretrained	EarthCam	**13.31**	**0.63**	0.69
custom v2_ no_ccm_wb_gain	EarthCam	12.31	**0.63**	0.73

Table 3: The v3 model outperforms the pre-trained model on pepper data. Quantitative results on EarthCam data are inconclusive as we don't have a proper reference image.

Model Outputs

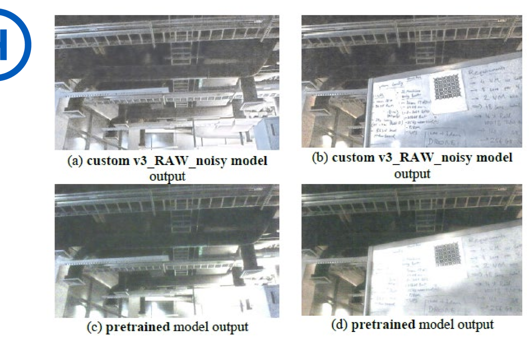
LOL

G

(a) Generated Image (b) Reference Image (c) Noisy Low Light Image (d) Paired Low light Image

PEPPER

H

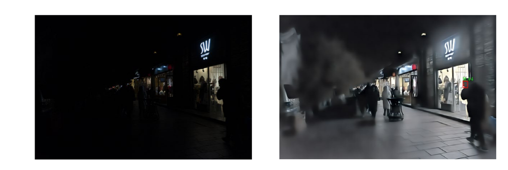


(a) custom v3_RAW_noisy model output (b) custom v3_RAW_noisy model output (c) pretrained model output (d) pretrained model output

The custom model maintains better exposure and preserves details on the image whiteboard.

VE-LOL-H

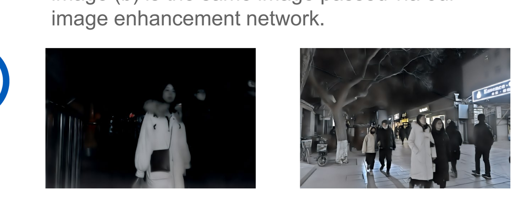
E



(a) (b)

Image (a) is from the VE-LOL-H dataset, where it's completely unprocessed whereas the right image (b) is the same image passed via our image enhancement network.

F

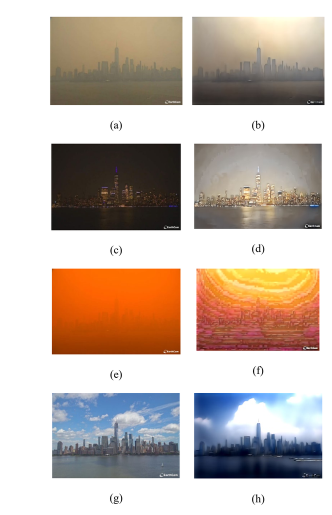


(a) (b)

Face detection using RetinaFace works much better with the enhanced images than with the original low-light ones. But for face recognition, we need the fine details that are lost after image enhancements.

EarthCam

G



(a) (b) (c) (d) (e) (f) (g) (h)

Here the left image is the reference, right image is the model output. The model works well for the pair (a)-(b). For the pair (c)-(d), the model brightens the image but also loses some building details. The model fails to process pair (e)-(f), which is cloudy and lacks details. The model also fails to handle the context for the last pair and creates unrealistic artifacts.

Conclusion

- Adding shot + read noise while training helped in improving the model's image enhancement performance.
- Datasets lack in modeling all the possible real-life circumstances
- Context awareness for a model is very important and our future research should be taking that into account. It will help the models to make image-specific enhancements.

References

1.T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet and J. T. Barron, "Unprocessing images for learned raw denoising.," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019*.

2. Y. Wang, R. Wan, W. Yang, H. Li, L.-P. Chau and A. Kot, "Low-light image enhancement with normalizing flow," in Proceedings of the AAAI Conference on Artificial Intelligence., 2022.

3.W. Chen, W. Wang, W. Yang and J. Liu, "Deep retinex decomposition for low-light enhancement," in arXiv preprint arXiv:1808.04560 (2018).

4.Z. Richard, P. Isola, A. A. Efros, E. Shechtman and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric.," in Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

5.J. Liu, D. Xu, W. Yang, M. Fan and H. Huang, "Benchmarking Low-Light Image Enhancement and Beyond," in *International Journal of Computer Vision (2021)*, 2020.

6.https://www.earthcam.com/usa/newyork/worldtradecenter

7.X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao2 and X. Tang, "Esrgan: Enhanced super-resolution generative adversarial networks.," in Proceedings of the European conference on computer vision (ECCV) workshops. 2018..

University at Buffalo The State University of New York

Department of Computer Science and Engineering
School of Engineering and Applied Sciences
cse.buffalo.edu