The effect of power quality on photos taken by the camera module. Software image enhancement affected by noisy power supply.

Mykhailo Buleshnyi
Faculty of Applied Sciences
Ukrainian Catholic University
L'viv, Ukraine

Maksym Buleshnyi
Faculty of Applied Sciences
Ukrainian Catholic University
L'viv, Ukraine

Anna Yaremko
Faculty of Applied Sciences
Ukrainian Catholic University
L'viv, Ukraine

Iryna Kokhan
Faculty of Applied Sciences
Ukrainian Catholic University
L'viv, Ukraine

Denys Levchunets
Renesas Electronics
L'viv, Ukraine

***Abstract*—The objective of this project is to explore both hardware and software methodologies for enhancing image quality. Specifically, the focus of this report will be on the hardware aspect related to power supply. The investigation aims to understand the impact of a camera operating with a noisy power supply and seeks to implement software-based solutions to improve the overall image quality. For this project, we used a Raspberry Pi image sensor.**

I. Introduction

Different image sensors are used daily, and their creators aim to make them as good as possible. One major challenge in practical applications is the instability of the power supply, leading to substantial impairments in image quality. To address this issue, Low Dropout Regulators (LDO) are employed to convert variable power inputs into stable outputs, ensuring reliable and consistent sensor operation. However, it is important to note that LDOs have their limitations. In this work, we want to present how changing voltage, adding noise, and changing other conditions impact image quality. In such scenarios, we want to make image quality as good as possible using Machine Learning. We will make tests and prove performance boost using well-known metrics.

II. Testing setup

We spent much time creating the perfect environment to make our research accurate. Our test setup is based on a black box painted with black matte, light-absorbing paint (Fig. 2). We fixed the LED strip with yellow and white light. Created a special mount for the camera to keep it in the same place for all experiments. As a reference image, we have chosen the image shown in Fig. 1, which consists of different small tests (color rendering, display of small details, etc.). We used this image to track all possible changes in image representation. This image was printed on the matte printer using matte paper to get rid of reflections. We set up 1-second delays and captured multiple images to ensure the objectivity of our report. For the power supply, we used generators to make testing stable. For mixing the stable signal with the noise, we used a line injector created by Renesas Electronics.

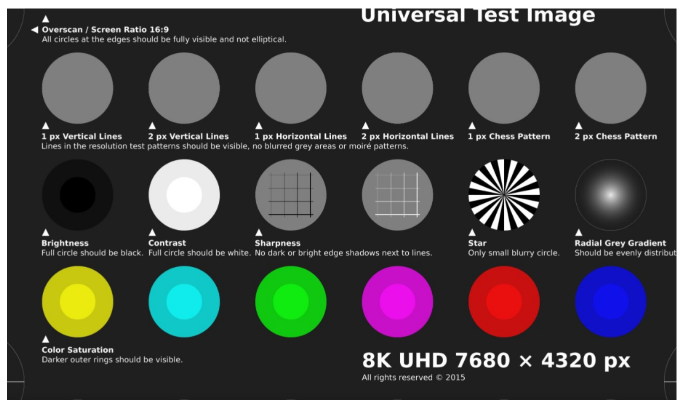


Fig. 1. Reference image used for tests.

Fig. 2. Created test setup.

III. Preparing cameras for tests

A. Low dropout regulator (LDO)

An LDO regulator is a linear regulator that can operate at a low difference between input/output voltage. It is needed for devices that require steady voltage despite noise, like camera modules. During this research, we will show images made without LDO that prove its necessity.

B. Preparation of camera modules

For this research, we had two image sensors. From one, we removed analog LDO (AP7331) [2]. The second one was modified so that it could easily be powered not by the development board (we have used the Raspberry Pi) but by a generator to control the noise level. The control points were soldered exactly to the analog LDO from 3.3V to 2.8V.

IV. Additional noise reduction

A. Convolutional Neural Network in our perspective

To improve image quality, especially by eliminating noise, we chose to use neural networks (NN). Building such a model takes a lot of time, so our approach involves using pre-trained models. Pre-training a neural network means first training it on one task or dataset and then using what it learned to train another model for a different task or dataset.

In general, the image-denoising model can be formulated mathematically as follows:

$$Y = X + N,$$

where Y is the noisy image, X is the clean image, and N is the noise to be removed. In most cases, the noise is Gaussian. Currently, there are many models to address this problem, such as DnCNN (Denoising convolutional neural networks), NRLN (Non-local recurrent network for image restoration), and others, but not all of them yield good performance.

In contrast to regular image denoising, where the noise type and level are assumed to be known, blind denoising handles cases where the noise level or type is unknown. Various methods have been explored, such as using a single deep model for Gaussian denoising with different noise levels and tasks like JPEG compression and image super-resolution. However, these methods are mostly evaluated on synthetic or processed noise and face challenges when applied to real-world scenarios with more complex noise. Establishing practical noisy/clean image pairs for training deep blind models remains unsolved.

B. Solution and SCUNet

We used a pre-trained model largely developed by Kai Zhang based on scientific work with other colleagues [5]. The idea of using pre-trained models is quite common in the realm of neural networks. In essence, it involves beginning with a pre-trained model that has accumulated extensive knowledge from vast datasets, which we then adapt and refine to suit our particular task. This method is widely embraced and recognized as a best practice in the field of NNs. It is based on the use of a deep blind model. The working principle is that this model is similar to DRUNet (Dilated-Residual U-Net Deep Learning Network). Still, it adopts four swin-conv (SC) blocks instead of four residual convolution blocks at each scale down/upscale.

To break it down further, UNet [1] is a common neural network architecture used for image-related tasks. In SCUNet, we've infused the UNet backbone with these SC blocks. Now, let's unpack what these SC blocks do:

- SC blocks bring together a "swin transformer" (SwinT) block and a "residual convolutional" (RConv) block. This fusion is achieved through various operations, including convolutions, splitting, concatenation, and a residual connection.
- The SC block essentially takes an input, applies convolution, splits it into two groups (X1 and X2), processes each group separately through SwinT and RConv blocks, and concatenates and convolves the results to produce the final output.

Here's why this is important:

- Local and Non-local Modeling: SCUNet benefits from the best of both worlds – local details are captured by RConv, while SwinT focuses on understanding non-local relationships in the data.
- Multiscale UNet: The UNet structure in SCUNet operates at multiple scales, allowing for a comprehensive understanding of features at different levels.
- Efficient Information Fusion: The convolutions effectively blend information from SwinT and RConv blocks.
- Complexity Reduction: Operations like split and concatenation act like a clever shortcut, reducing the computational complexity and the number of parameters.

V. Super Resolution CNN

SRCNN is not a very deep model; it contains only 3 parts:

- patch extraction and representation,
- non-linear mapping,
- reconstruction.

We will use it to add details to the image after denoising to enhance image quality even more.

A. Patch Extraction

This is the first layer, which is used for extracting a set of feature maps from the low-resolution input image. This involves breaking the image into many tiny parts, which are then turned into high-dimensional vectors. Generally, the operation of this layer can be expressed as:

$$F_1(Y) = \max(0, W_1 * Y + B_1),$$

where Y – input image; W_1 – n_1 convolution with the kernel size of $c \cdot f_1 \cdot f_1$, where c – number of channels, f_1 – spatial size of the filter; B_1 – n_1-dimensional vector.

This step is essential for keeping important information that improves the overall quality and clarity of the final high-resolution image.

[1] Convolutional neural network developed for biomedical image segmentation at the Computer Science Department of the University of Freiburg.

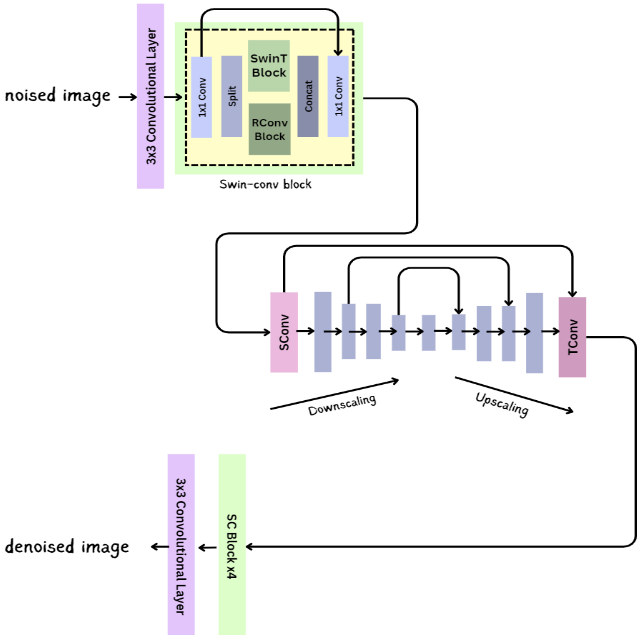


Fig. 3. The idea of this structure of the neural network is proposed by Kai Zhang and others [5].

B. Non-linear mapping

The non-linear mapping layer transforms the high-dimensional vectors obtained from the patch extraction step. This layer applies non-linear functions to capture complex patterns and relationships within the image data, allowing the model to learn intricate details. This mapping is defined with the following formula:

$$F_2(F_1) = \max(0, W_2 * F_1 + B_2),$$

where F_1 – output of the first layer; W_2 – n_2 filters of the size of $n_1 \cdot f_2 \cdot f_2$; B_2 – n_2-dimensional vector – corresponding bias vector.

This layer refines the feature maps, making them more representative of the high-resolution image.

C. Reconstruction

The reconstruction layer combines the refined feature maps to produce the final high-resolution image. This layer aggregates the information from the previous steps to recreate a detailed and clear image. Such convolutional layer is represented as:

$$F(F_2) = W_3 * F_2 + B_3,$$

where F_2 – output of the second layer; W_3 – c filters of the size of $n_2 \cdot f_2 \cdot f_2$ – set of linear filters; B_2 – c-dimensional vector.

This final step ensures the model outputs a high-resolution image with enhanced details and improved quality.

VI. Metrics and Analysis

A. Introduction to Metrics

An important stage of our work is assessing the image quality, which consists of three stages:

- Determining the quality of the obtained photo.
- Determining the quality of the noisy photo.
- Assessing the quality of the denoised photo using a neural network.

We use a set of metrics to analyze different image stages that allow us to approach this task comprehensively. Each of these metrics, such as MSE (Mean Squared Error), RMSE (Root Mean Squared Error), PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), UQI (Universal Image Quality Index), and MS-SSIM (Multi-Scale Structural Similarity Index) is designed to assess image quality, but they employ different approaches and algorithms for this.

- MSE calculates the average value of the squares of the differences between corresponding pixels in the noisy and original images. A high MSE value indicates a significant difference between pixels, suggesting poor image quality.
- RMSE is the square root of MSE and measures the average difference between pixels concerning the original image. RMSE values also indicate the level of distortions in the noisy image.
- PSNR is expressed as the ratio of the maximum possible signal power to the present noise in the image. Higher PSNR values indicate lower distortion levels and are considered an indication of better image quality.
- SSIM compares the structural similarity between the original and processed (noisy) images, considering brightness, contrast, and structural components. SSIM values range from -1 to 1, where 1 indicates perfect similarity.
- UQI measures the similarity of structures between the original and processed images, considering image quality and contrast. A high UQI value indicates better image quality.
- MS-SSIM enhances the evaluation of SSIM by considering multiple scales and accounting for structural properties at various levels. High MS-SSIM values indicate better image quality across different scales.

Different metrics may provide different image quality indicators when comparing noisy and original images. Typically, low values of MSE and RMSE and high values of PSNR, SSIM, UQI, and MS-SSIM indicate better image quality. However, it's important to consider the context and specific application nuances.

VII. Datasets

For our research, we collected datasets from image sensors for various noise levels(0 – 350 mV), constant power supply (2.7V – 3.3V), and light (from dark to bright: 30/70/150/400 mA). To be more precise, we captured a handful of images for every one of these combinations. In general, we have more than two hundred images. To collect data, we used our test set (Fig. 4).

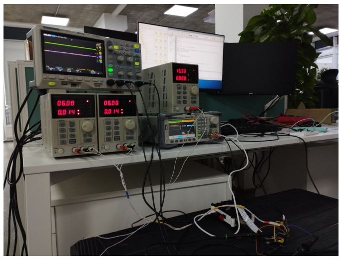


Fig. 4. Dataset collection with test setup.

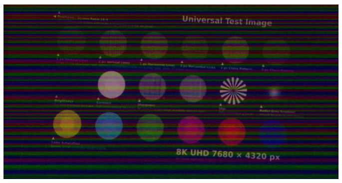


Fig. 5. Image with 10mV noise (No LDO)

VIII. Analysis

Firstly, we tried to give some noise straight to the camera module without LDO. We notice that the 2 mV noise of the power supply caused significant changes in the image. All images were covered by strange line patterns. The maximum noise we set was 10 mV (Fig. 5). There is no need to make a deep analysis of this dataset as it significantly changes from image to image. Also, it is not a very realistic scenario. But that means that an unstable power supply can completely spoil the image.

Then, we tried to vary the power supply on the image sensor to test how built-in LDO is coping with its work. Using contrast and dynamic range metrics, it is easy to notice that by adding a noised power supply, we lose the quality of the image; moreover, the image quality becomes very unstable from image to image (Fig. 6).

TABLE I
Metrics Comparison, with and without LDO.

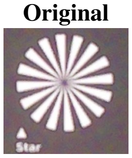
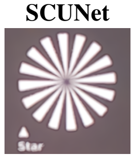
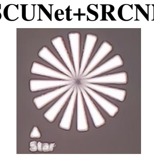
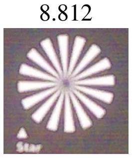
	Original	SCUNet	SCUNet+SRCNN
With LDO			
SSIM	0.331	0.374	0.413
UQI	0.532	0.631	0.574
PSNR	8.812	9.368	8.934
Without LDO			
SSIM	0.232	0.261	0.251
UQI	0.380	0.382	0.377
PSNR	7.505	7.527	7.475

Table I shows chip variants with LDO (voltage 3.3 mV, light 70 mA, and noise overlay 150 mV) and without it (voltage 2.8 mV, light 70 mA, and noise overlay 2 mV). In both cases, the metrics show that there is indeed more noise without LDO, and when using SCUNet, the quality improves. However, it can be noticed that when using both neural networks, the metrics may show slightly worse indicators. This is due to the fact that SCUNet can slightly blur the photo, and SRCNN can add sharpness, for which the metrics are not ready for evaluation.

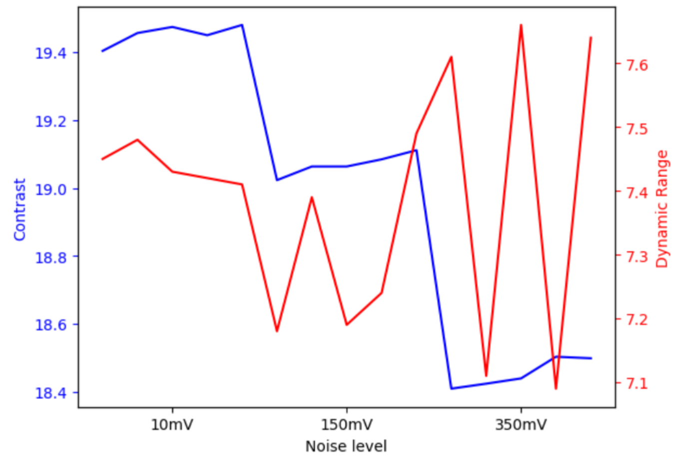


Fig. 6. Contrast and Dynamic Range dependence on noise level (5 images for each noise level) (with LDO)

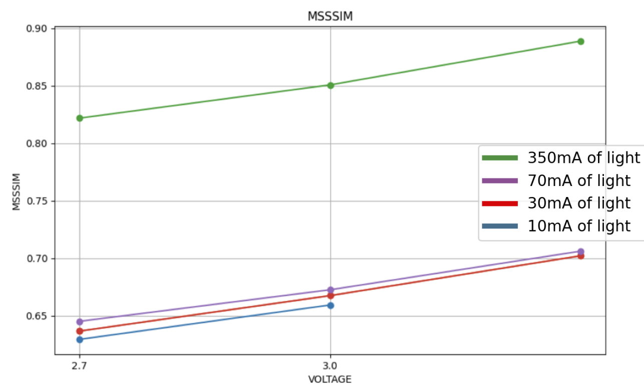


Fig. 7. Dependency of voltage with MS-SIM.

We also conducted research on how quality changes depending on the permanent voltage. For example, looking at the MS-SIM graph (Fig. 7), which demonstrates structural similarity, it's noticeable that as the voltage increases, the noise level decreases, and the structural similarity of the image to the 'noiseless' one increases. These graphs also indicate that the higher the light level, the higher the MS-SIM value. For instance, the blue line has a light level of 10, while the green one has 350, showing a significant difference.

The provided graphs (Fig. 8) show various image quality metrics, with noise level on the Y-axis and light + voltage value on the X-axis. Each graph can be divided into three sec-

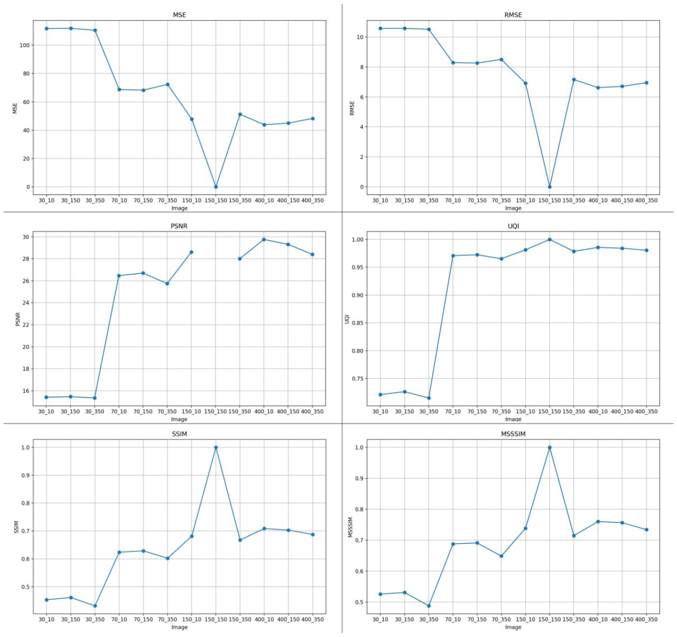


Fig. 8. Dependency of noise on voltage and light – all metrics

tions, representing different light levels (low – 30mA, medium – 70mA, etc.). Within each section, the graphs illustrate how the noise changes as voltage increases.

In general, noise decreases as voltage increases across all light levels. This trend is particularly noticeable in the medium and high light settings. Metrics like PSNR, UQI, and SSIM show significant improvement with higher voltage, indicating better image quality and less noise.

The best performance, with the least noise, is observed at medium to high light levels with higher voltage. This conclusion is supported by the improved values of PSNR, UQI, and SSIM, along with the decreased values of MSE and RMSE.

In summary, increasing the noise level worsens photo quality, whereas increasing voltage and light levels enhance quality, based on our research and analysis of results using various metrics.

IX. Conclusion

Certainly, there are numerous hardware methods to enhance image quality. However, this report specifically concentrates on power supply and its significance. It is evident that the power quality of the image sensor plays a crucial role in influencing various image characteristics, including image quality, color rendering, stability, noise level, and contrast. Our findings highlight the critical importance of power quality, particularly the use of a high-quality Low Dropout Regulator (LDO), in maintaining and improving image sensor performance.

In addition, advanced software techniques, particularly convolutional neural networks such as SCUNet, have been shown to be effective in improving image quality, as evidenced by the improved performance in our analysis. Our comprehensive approach, which combines both hardware and software algorithms, emphasizes the need for a balanced strategy to achieve optimal image quality. The code of the project is available at https://github.com/mikl123/POK-PROJECT.

References

[1] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *Cornell University*, 2014. [Online]. Available: https://arxiv.org/abs/1501.00092

[2] D. inc.", "300ma, low quiescent current, fast transient low dropout linear regulator," 2021. [Online]. Available: https://www.diodes.com/assets/Datasheets/AP7331.pdf

[3] N. Singh, T. Thilagavathy, R. T. Lakshmipriya, and O. Umamaheswari, "Some studies on detection and filtering algorithms for the removal of random valued impulse noise," *IET Journals - The Institution of Engineering and Technology*, July 2017. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-ipr.2017.0346

[4] T.Eude, H. Cherifi, and A. Mayache, "A comparison of image quality models and metrics based on human visual sensitivity," *IEEE Xplore*, November 1988. [Online]. Available: https://www.researchgate.net/publication/3774448_A_comparison_of_image_quality_models_and_metrics_based_on_human_visual_sensitivity

[5] K. Zhang, Y. Li, J. Liang, J. Cao, Y. Zhang, H. Tang, D.-P. Fan, R. Timofte, and L. V. Gool, "Practical blind image denoising via swin-conv-unet and data synthesis," *Machine Intelligence Research*, 2023. [Online]. Available: https://link.springer.com/article/10.1007/s11633-023-1466-0