Real-World Mobile Image Denoising Dataset with Efficient Baselines: Supplemental Material

Roman Flepp[1] Andrey Ignatov[1,3] Radu Timofte[1,2,3] Luc Van Gool[1,3]

[1] Computer Vision Laboratory, ETH Zürich [2] Computer Vision Laboratory, University of Würzburg [3] AI Witchlabs Ltd., Zollikerberg

r.flepp@hotmail.com, {andrey, timofter, vangool}@vision.ee.ethz.ch

1. Additional Material

1.1. Model

To foster a better understanding of the SplitterNet architecture and its tensor shapes, we provide an additional visualization of its baseline block in Fig. 1. One should keep in mind that this figure only shows a two-step architecture while the SplitterNet configuration in this paper uses a 4 step setup.

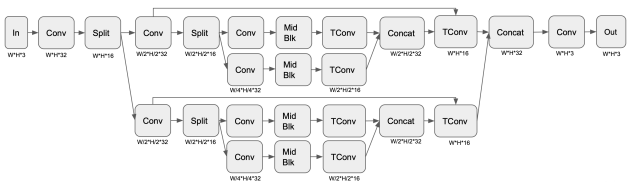

Figure 1. A simplified one-step view of the proposed architecture. The full model architecture consists of 4 such encoding and decoding steps, such that there are 16 middle blocks in total. The convolutions are all using 3×3 kernels. The width (W), height (H), and channel number of the output of a given component are written below it. The components without explicit dimensions take the same dimensions as the components below.

Within SplitterNet's design, the introduction of a slicing / splitting strategy and the creative use of skip connections mark significant changes away from traditional network structures. This architecture is particularly engineered to enhance image reconstruction capabilities by adopting a more segmented approach to processing image data.

The key innovation within SplitterNet lies in its overall architecture, where it differs from the conventional U-Net by dividing each feature map into halves along the channel dimension in each encoder step. This division results in two distinct tensors, effectively halving the original feature map's channels. This method enables separate processing paths for each tensor, providing a nuanced feature analysis that traditional, uniform processing methods might miss. This architecture enhances the network's capability to take apart and reconstruct complex image details by examining a broader range of input features.

Model	PSNR	SSIM	Runtime, *ms*
Identity Mapping	30.86	0.672	-
Model (1)	37.50	0.877	**17.5**
Model (2)	37.85	0.883	-
Model (3)	**38.00**	**0.883**	56.1
SplitterNet	37.93	0.882	27.7

Table 1. PSNR, SSIM and runtimes results obtained on the MIDD dataset for the ablation study of the SplitterNet. The runtime was estimated on the Exynos 2200 Mali GPU on images of size 720×480 px. Model (1) refers to the SplitterNet without splitting and using a constant channel number of 32. Model (2) refers to the SplitterNet with additional Layernorm in each encoding and decoding step. Model (3) refers to the SplitterNet without splitting and using an increasing channel number of 32 up to 512. The runtime is halved when using the splitting operation compared to not using it. Finally, the SplitterNet offers the best runtime to fidelity score tradeoff.

Further big changes included in the SplitterNet are its skip connections. Unlike the typical usage, which primarily aims at preserving spatial information, SplitterNet's skip connections serve a dual purpose. They not only maintain spatial continuity across the network but also bring back together features processed along parallel encoder paths during the decoding phase. By merging these split tensors with upscaled feature maps, the network ensures a thorough reintegration of spatial and feature details lost in downsampling. This mechanism significantly contributes to the improved reconstruction of images, making the most of the split-feature approach.

1.1.1 Ablation Study SplitterNet

A simple ablation study is provided as the SplitterNet includes a novel model architecture. We want to find the impact that splitting the tensors for each encoding step creates compared to two models where we do not use splitting. The first model uses a constant channel dimension, as the tensors inside the SplitterNet never exceed the channel number

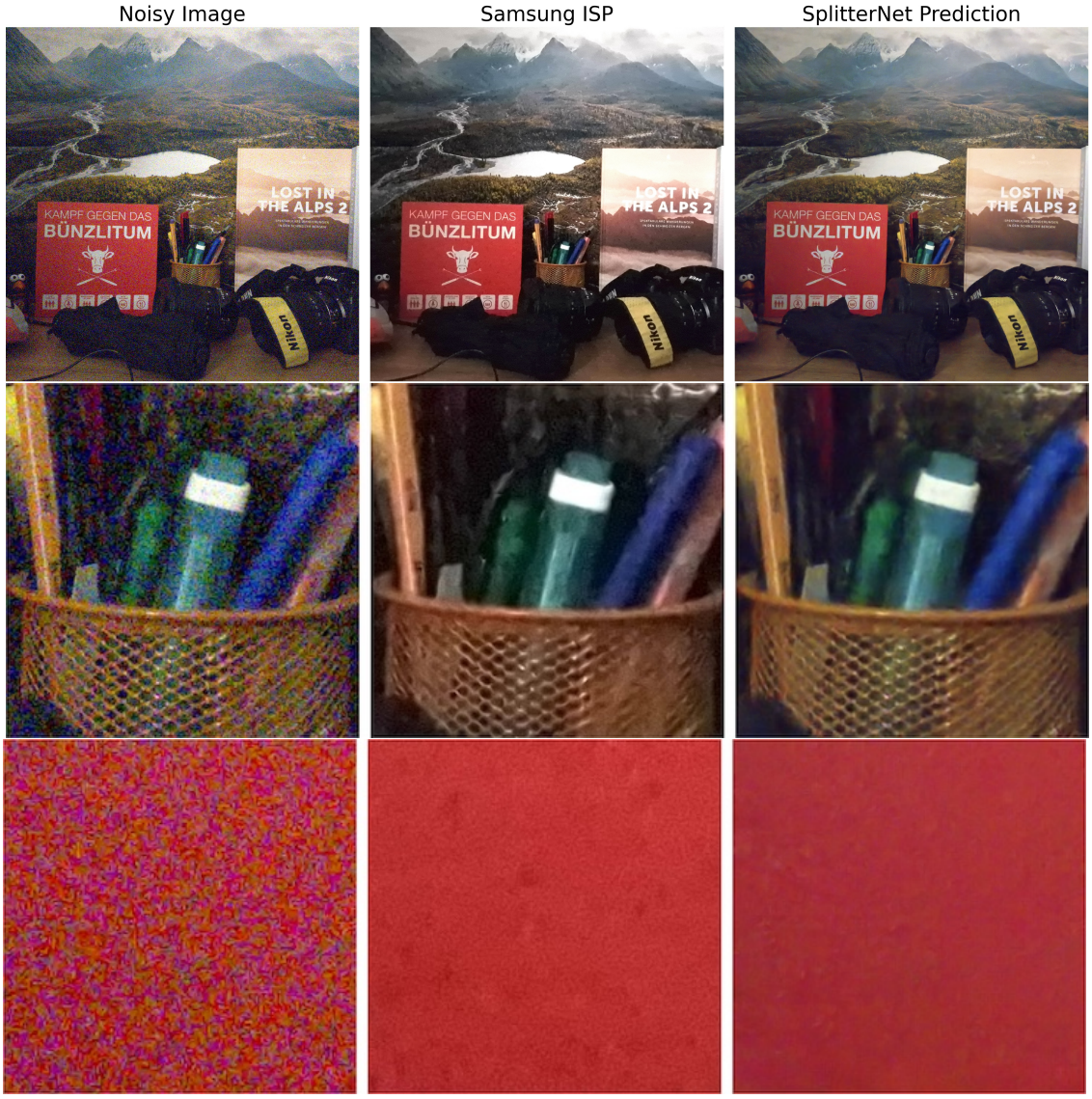

Figure 2. From left to right: processed raw photo, image obtained with S23 Ultra's ISP, and the photo denoised with the SplitterNet model. The Samsung ISP does not remove noise efficiently in flat regions and does not keep as many details as our SplitterNet.

of 32. The second model uses increasing channel dimensions, as the sum of channel numbers in each encoding step is increased. Furthermore, an analysis using Layernorm is presented as it has shown to be valuable in other denoising networks. It should be also noted that the LayerNorm op is currently not supported by current mobile ML frameworks, thus the resulting architecture would not be able to run on mobile devices and thus the runtime is not reported. In Tab. 1 we can see that:

1. The first Model offers the fastest runtime, as its channel number is constant, and fewer operations need to be performed compared to the SplitterNet. This comes with a price, the PSNR and SSIM scores are both significantly lower.

2. The second model performs very similarly to the SplitterNet without LayerNorm. These results are expected since LayerNorm usually helps in tasks where global image changes are introduced.

3. The third model offers slightly better PSNR and SSIM scores but a doubling of the runtime, which corresponds directly to the doubled mathematical complexity of not using the splitting operation.

Following the observations, it can be concluded that the proposed architecture offers superior performance when taking into account runtime as well as PSNR and SSIM scores.

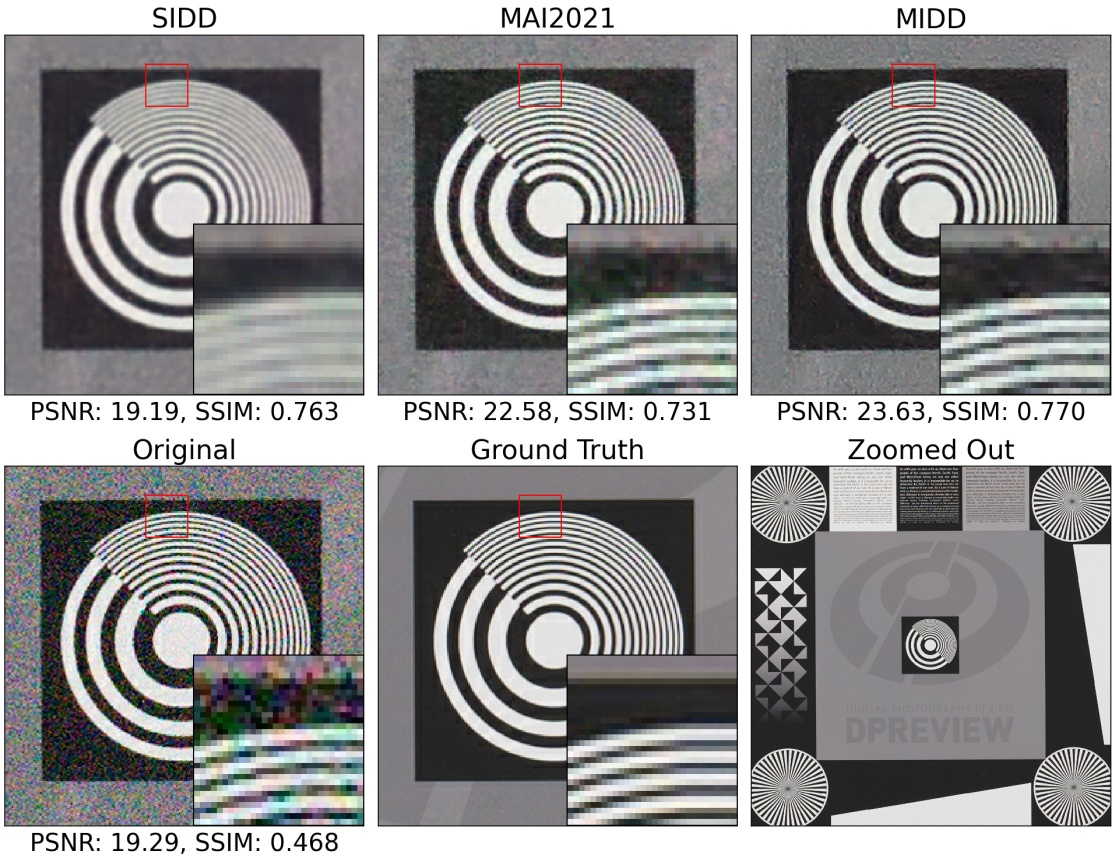

Figure 3. Visual results for the SplitterNet trained on the SIDD dataset for 50 epochs instead of 10 (top left). Still, the blur is not reduced and the scores are not significantly higher.

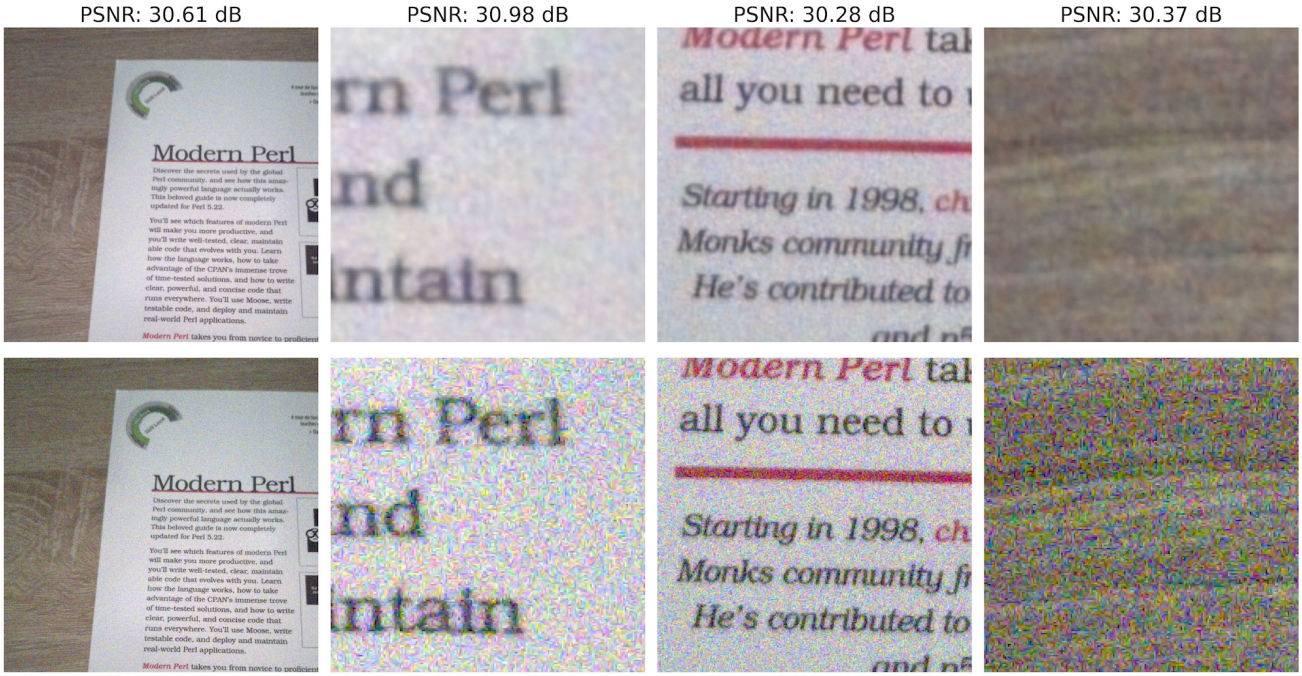

Figure 4. An example of using SplitterNet trained on the SIDD for prediction (top row). Again, a strong blur is introduced to the resulting images. For comparison, one can see the denoising performance of the SplitterNet trained on the MIDD on the same image in Fig. 6.

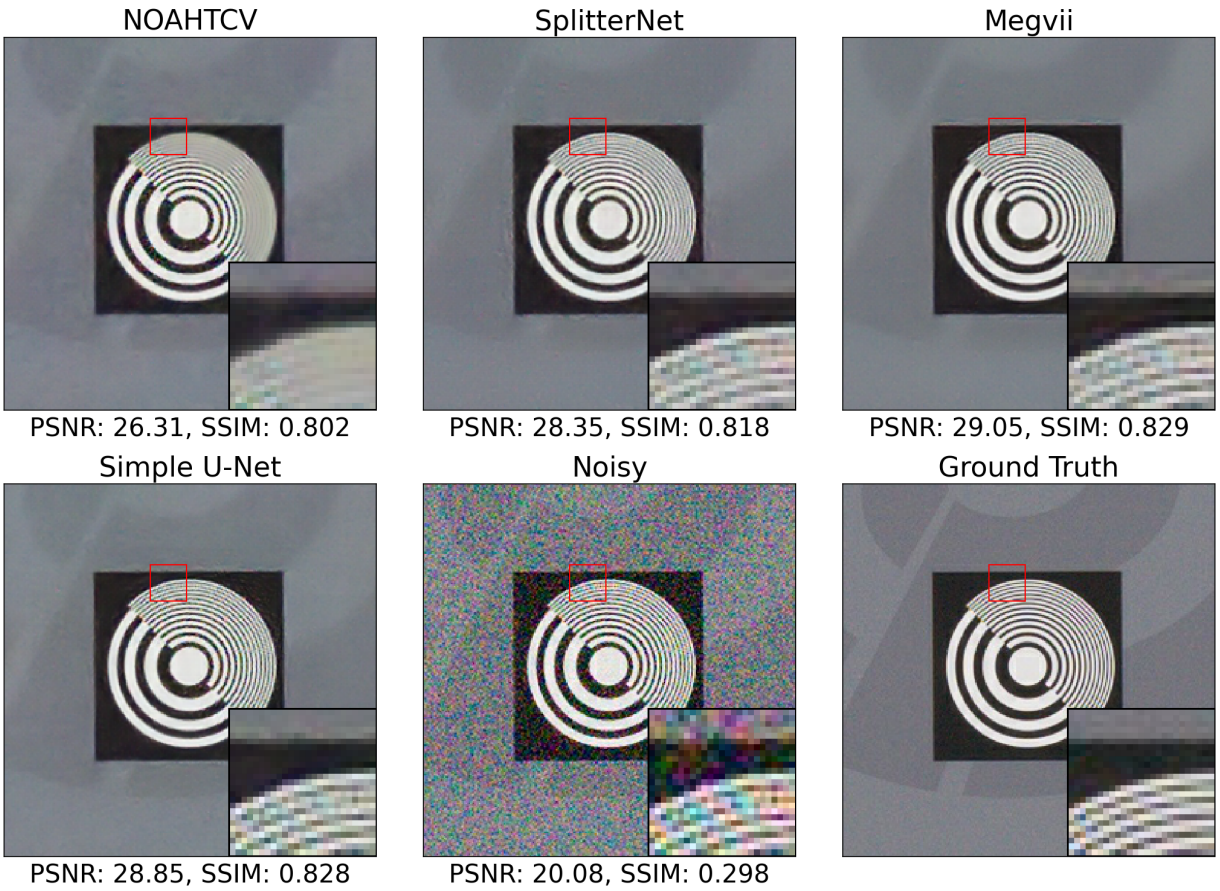

Figure 5. A comparison of mobile denoising model performance on a sample image from the independent DPreview dataset. SplitterNet performs similarly to the Megvii network, while the NOAHTCV solution offers the worst performance. A simple U-Net model offers a strong denoising performance, with a better line separation than the Megvii model, but weaker noise removal overall.

1.2. Visual Comparison Samsung S23 Ultra ISP

Comparing our denoising model with image denoising capabilities of current mobile Image Signal Processors (ISPs) presents an intriguing analysis. However, it is important to remember that these ISPs perform a wide range of other image enhancement techniques, therefore a direct comparison may not be entirely meaningful. Sample results for the Samsung S23 Ultra camera are demonstrated in Fig. 2. We should note that conducting a numerical performance comparison is unfortunately not possible in this case as image results obtained with different ISPs differ significantly in texture and color rendition (which affects the PSNR and SSIM scores).

1.3. Training on SIDD

From the dataset comparison in the results section, it was shown that the SplitterNet trained on the SIDD performs poorly and the resulting images are blurry. One could argue that the SplitterNet should be trained for more epochs on SIDD to provide sharper images. In Fig. 4, one can see the results for SplitterNet being trained for 50 epochs on SIDD. The resulting predictions are not significantly sharper and thus the blur is not due to a shorter training time.

1.4. Visual Model Performance

In Fig. 5, one can see the performance of different models on a sample DPreview image. All models were trained on the MIDD dataset. The Megvii network achieves the cleanest visual output, closely followed by a simple U-Net and SplitterNet. The NOAHTCV network introduces a strong blurring. One can conclude that the performance of the SplitterNet is closer to the Megvii network than to the NOAHTCV model.

1.5. Visual Examples SplitterNet

To further analyze and show the performance of the SplitterNet on real-world scenarios from our MIDD dataset, we provide additional examples of different lighting conditions and scenes in Fig. 6 and Fig. 7.

Figure 6. Qualitative evaluation of images denoised using the SplitterNet trained on the MIDD. The first and third rows, denoised images obtained with our model are presented as well as their PSNR scores. The original noisy input images are shown in the second and fourth rows. The first two rows show an indoor scene with strong noise. The model extracts the noise very well on the wooden surface, as seen in the fourth column, but introduces some artifacts when denoising the letters on the book surface. The second scene shows a daylight image with very little noise. Our model does not over-smooth edges and only minor changes are made by the model.

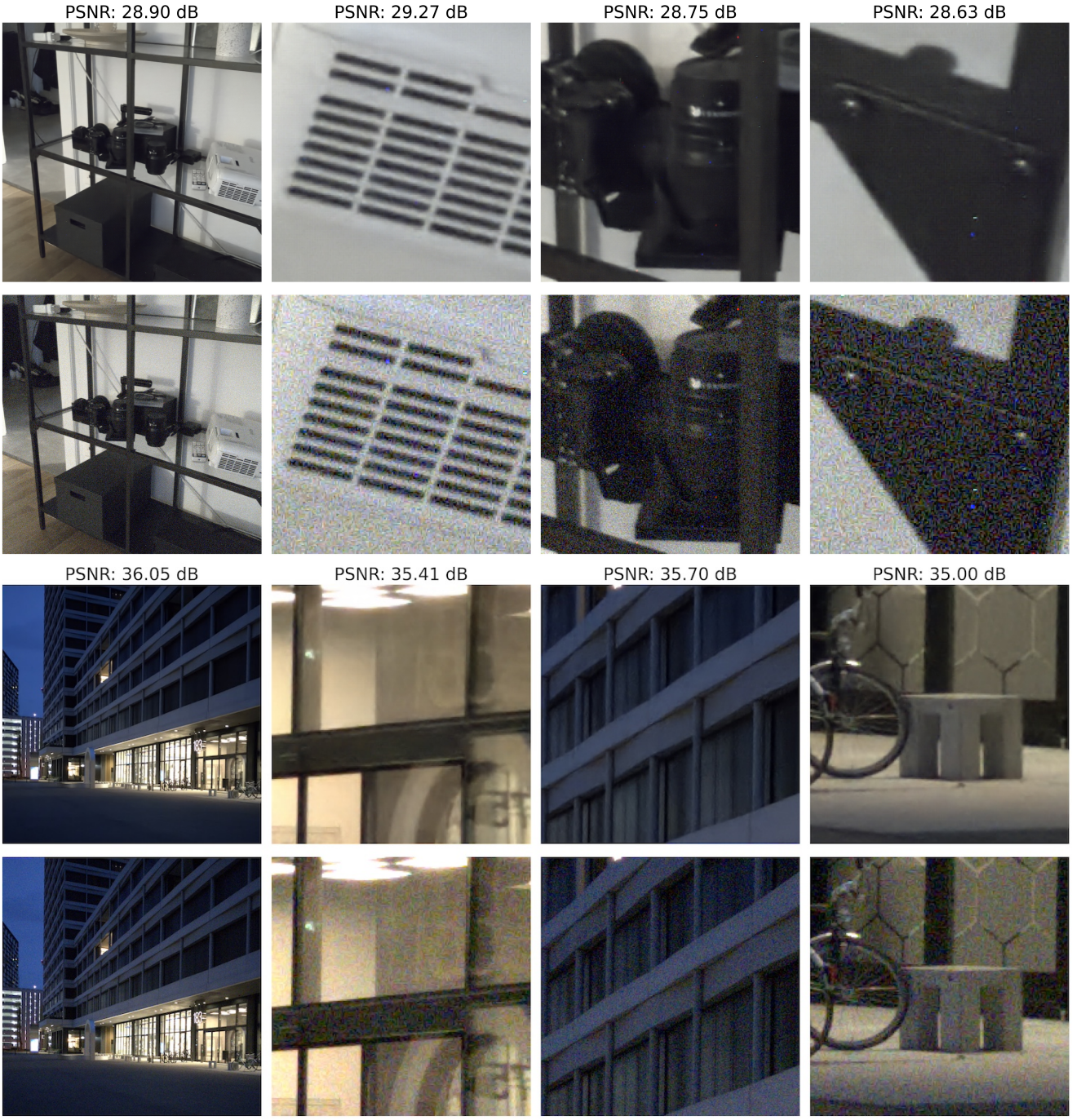

Figure 7. Two more images with strong noise from an indoor and an outdoor scene are shown. In both situations, the model performs well and extracts noise efficiently. On the top right image, a defective pixel is seen, which does not get extracted, which is the intended behavior.