

Supplementary Material for: Physically Plausible Spectral Reconstruction from RGB Images

Yi-Tun Lin and Graham D. Finlayson

A. Spectral Reconstruction on Real RGB Data

Most of the image-based spectral reconstruction (SR) methods are trained with *simulated* RGB data (including, the experiments we present in the main paper), due to the difficulties of getting registered ground-truth hyperspectral and RGB images. But, presumably, we wish to use the SR models on real world data, where RGB images are taken by actual cameras.

In this supplementary material we aim to show some visual evidence that the physical plausibility of SR is crucial for maintaining color fidelity in real-world applications. More ambitiously, we wish to compare the actual ‘end-of-pipe’ output of a camera system (*i.e.* the processed image shown to the end users) with the prediction given by spectral reconstruction. In detail, with a model of the processing pipeline in hand (we shall introduce this model later), we (1) take a camera raw image, (2) recover the corresponding hyperspectral images by SR, (3) reintegrate the spectra with the given camera spectral sensitivities and finally (4) apply the pipeline model to this reintegrated raw image to generate an approximated end-of-pipe image.

We get these ‘real RGB images’ from the INTEL-TAU image database [5], which is by far the largest open-source database for training and evaluating the algorithms of color constancy (*i.e.* illumination estimation). This database is very useful because it provides with each raw image:

- the spectral sensitivities of the camera used
- the expected end-of-pipe rendered image
- the ground-truth white point color (WP)
- the color correction matrix (CCM) which maps the raw RGBs to sRGB colors.

We are going to make use of all the above information in our demonstration. Example images from this database are given in Figure 1.

A.1. Training

We re-trained two SR models for comparison: the original¹ HSCNN-R and the proposed HSCNN-R^{pd} model (this second model is guaranteed to recover spectra that are colorimetrically accurate and also robust to variation in scene exposure). The purpose of this re-training is that we are to apply these models on real RGB data where the camera’s spectral sensitivities are different from the CIE 1964 color matching functions [3] we used in the main paper.

¹The HSCNN-R model [9] ranked the 2nd place in 2018 NTIRE Challenge on Spectral Reconstruction from RGB Images [2]

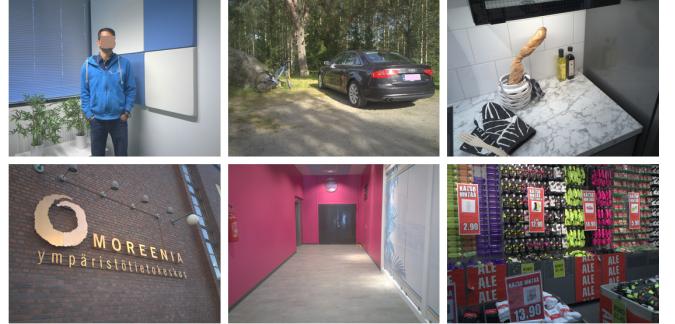


Figure 1: Example images in INTEL-TAU database [5]

Following the same training process as in the main paper, we randomly selected 100 ground-truth hyperspectral images from the ICVL dataset [1] for training and 50 for validation (the spatial dimension of these images is around 1300×1392). The only difference is that now the corresponding raw RGB images were simulated by the spectral sensitivities of SONY IMX135 (one of the three cameras used in INTEL-TAU). That is, the two SR models were trained to map SONY IMX135 raw RGBs to hyperspectral image output.

A.2. Reconstruction

The two trained SR models were used to reconstruct the hyperspectral information from 6 selected raw RGB images from the INTEL-TAU dataset [5], all of which were taken by SONY IMX135. The spatial dimension of these images is 2448×3264 .

Then, the reconstructed hyperspectral images were again reintegrated into raw RGB images with the spectral sensitivities of SONY IMX135. At this stage, the proposed HSCNN-R^{pd} is expected to give the exact same RGBs as the input raw RGB images, whereas HSCNN-R can generate different RGBs. The goal of this supplementary test is to visually demonstrate how different (*how wrong*) this recovery can be from a colorimetric point of view.

A.3. Color Fidelity Test

From page 3 onward of this supplementary document, we are going to show several visual comparisons and quantitative error maps between the ground-truth and the SR predicted end-of-pipe RGB images. In this section we detail the image rendering process and the process of calculating the quantitative errors.

A.3.1 Visual comparisons on end-of-pipe images

In the processing pipeline of a camera, the raw image might undergo, but not limited to, the following processes before being shown to the end users: black level and saturation correction, white balancing, color correction and gamma correction. As we are already given the expected end-of-pipe image with each raw image in the INTEL-TAU database, we can alternatively build a 3D Look-up-table (LUT) which approximates the actual image processing pipeline: for each image, the LUT is built to relate each color in the ground-truth raw RGB image to the colors in the supplied (expected) end-of-pipe image.

This LUT can be optimized - in a least-squares sense - by *lattice regression* [6, 4]. To speed up the optimization process, we train the LUT on *thumbnail images*, where we simply downsample the images from the original 2448×3264 to 108×144 , and bin the colors by $24 \times 24 \times 24$ in the three color channels. Then, the full resolution ground-truth raw RGB and the raw RGB reintegrated from the reconstructed hyperspectral image are mapped to their respective end-of-pipe renditions by applying the same 3D LUT.

In Figure 3-8, an example image is shown in the bottom-left of each figure, in which the 4 regions of interest are marked with white squares. The ‘Ground Truth’ image (top-left of each figure) is actually the end-of-pipe image rendered by the trained 3D LUT mapping. On the other hand, from the ground-truth raw RGB we carry out spectral reconstruction (*i.e.* the two trained SR models) and reintegrate the recovered hyperspectral images with the camera sensitivities to get an *approximate* raw image. By applying the same LUT to this derived raw image we generate the end-of-pipe images predicted by the two SR models, as shown in the top-middle and top-right images in each figure.

We can already see that HSCNN-R, as an physically non-plausible spectral reconstruction model, introduces color shifts that are quite visible after color rendering, while our physically plausible HSCNN-R^{pd} successfully preserves the original colors in the ground-truth images. To further quantify the color shifts, we are bound to calculate the *color difference* between the ground-truth and the reintegrated RGB images.

A.3.2 Quantifying color differences

We wish to use the CIE 1976 color difference (ΔE) [7] to quantify the colorimetric errors. Since the ΔE is defined in CIELAB color coordinates (as shown in Equation (19) in the main paper), we must consider how we transform the camera raw RGB to their CIELAB counterparts.

The procedure is summarized in Figure 2. Unlike in the main paper where the CIELAB coordinates can be transformed directly from the CIEXYZ colors (with the white point color this mapping is one-to-one [10]), the mapping from the real camera’s raw RGB to CIELAB is *unknown* if the raw data is the only given information. Fortunately, INTEL-TAU also provides with each raw image the *color correction matrix* (CCM) that transforms the image into sRGB colors and the information of ground-truth white point (WP) that ensures one-to-one mapping between sRGB and CIELAB [10]. Finally, the desired ΔE color difference between ground-truth and reintegrated RGB images can be calculated from the transformed CIELAB images.

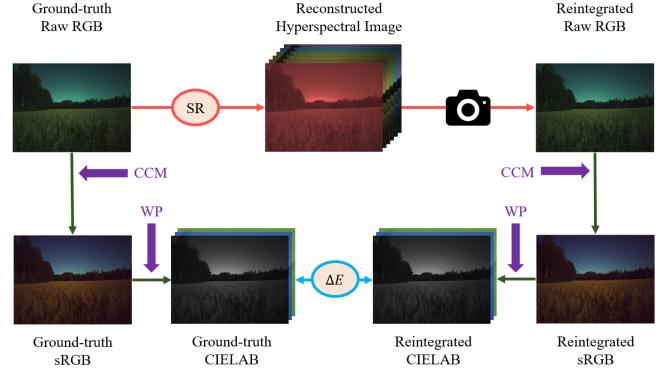


Figure 2: The process of calculating CIE 1976 color difference ΔE between ground-truth and reintegrated color images.

We show the ΔE error maps in the bottom-middle and bottom-right of Figure 3-8, which detail the pixel-wise colorimetric errors introduced by the two trained SR models in the 4 selected regions of interest. It is evident that HSCNN-R recovers spectra that reintegrate into *wrong* colors with significant errors (we remark once again that referring to [8] human observers can sense noticeable color difference above $\Delta E \approx 2.3$). Remarkably, our proposed HSCNN-R^{pd} model - which possesses both physical plausibility and exposure invariance - preserves complete color fidelity.

References

- [1] Boaz Arad and Ohad Ben-Shahar. Sparse recovery of hyperspectral signal from natural rgb images. In *European Conference on Computer Vision*, pages 19–34. Springer, 2016.
- [2] Boaz Arad, Ohad Ben-Shahar, and Radu Timofte. Ntire 2018 challenge on spectral reconstruction from rgb images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 929–938, 2018.
- [3] Commission Internationale de l’Eclairage. Cie proceedings (1964) vienna session, committee report e-1.4. 1. 1964.
- [4] Eric Garcia and Maya Gupta. Lattice regression. In *Advances in Neural Information Processing Systems*, pages 594–602, 2009.
- [5] Firas Laakom, Jenni Raitoharju, Alexandros Iosifidis, Jarno Nikkanen, and Moncef Gabbouj. Intel-tau: A color constancy dataset. *arXiv preprint arXiv:1910.10404*, 2019.
- [6] Hai Ting Lin, Zheng Lu, Seon Joo Kim, and Michael S Brown. Nonuniform lattice regression for modeling the camera imaging pipeline. In *European Conference on Computer Vision*, pages 556–568. Springer, 2012.
- [7] Alan R Robertson. The cie 1976 color-difference formulae. *Color Research & Application*, 2(1):7–11, 1977.
- [8] Gaurav Sharma and Raja Bala. *Digital color imaging handbook*. CRC press, 2002.
- [9] Zhan Shi, Chang Chen, Zhiwei Xiong, Dong Liu, and Feng Wu. Hscnn+: Advanced cnn-based hyperspectral recovery from rgb images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 939–947, 2018.
- [10] Sabine Süsstrunk, Robert Buckley, and Steve Swen. Standard rgb color spaces. In *Color and Imaging Conference*, volume 1999, pages 127–134. Society for Imaging Science and Technology, 1999.

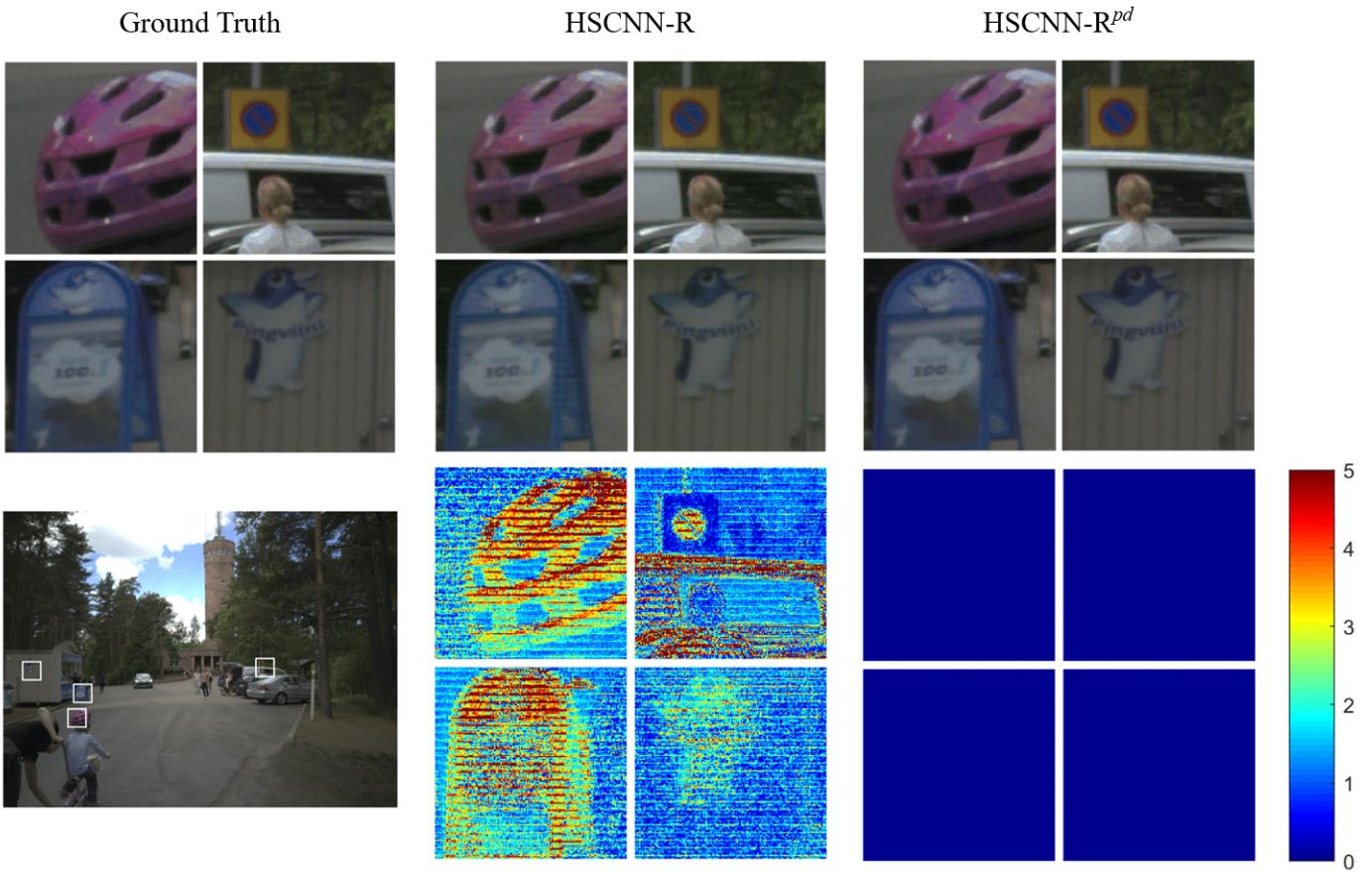


Figure 3: Visual comparison 1. Top row: the rendered images. Bottom row: the corresponding ΔE error maps.

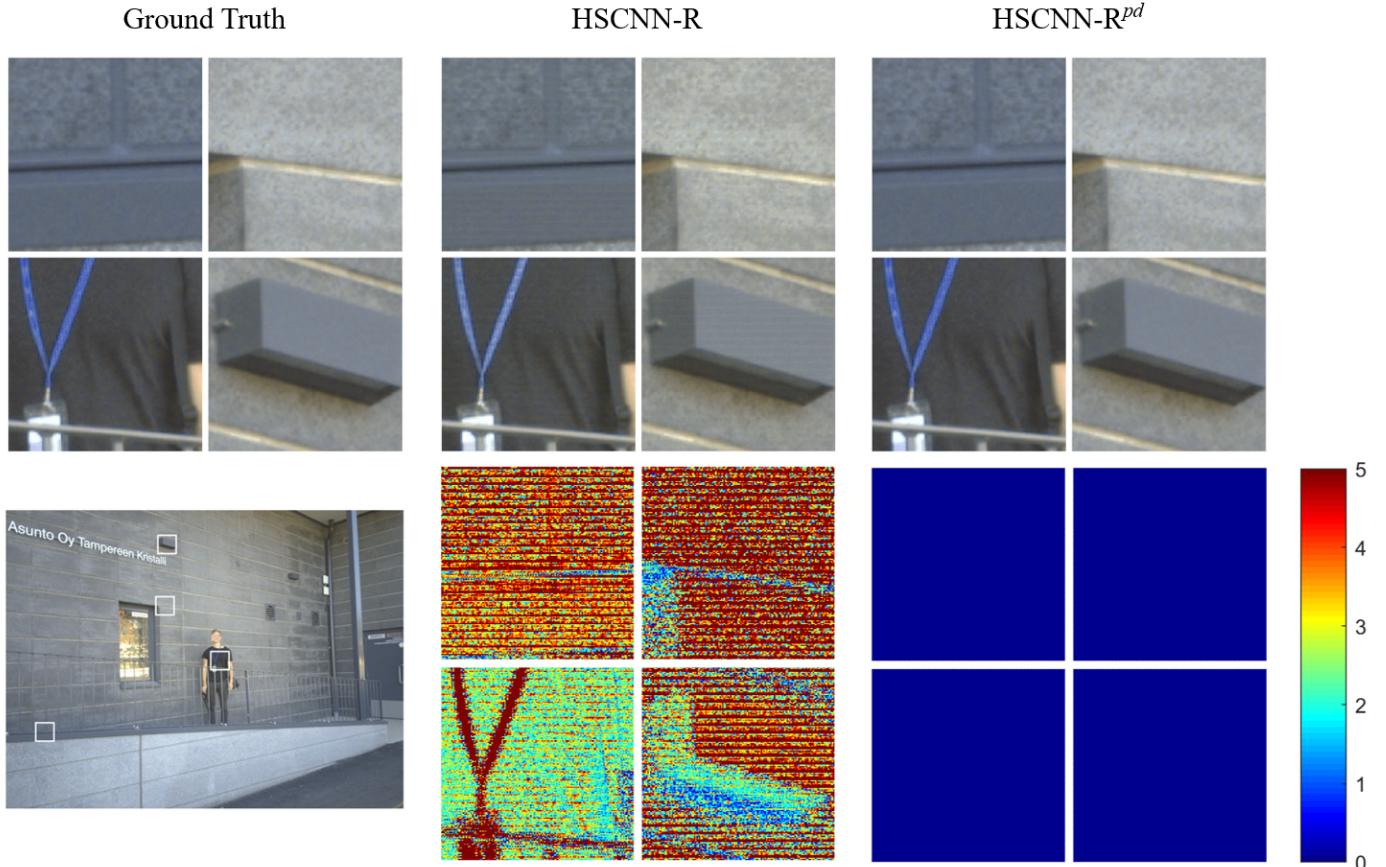


Figure 4: Visual comparison 2. Top row: the rendered images. Bottom row: the corresponding ΔE error maps.

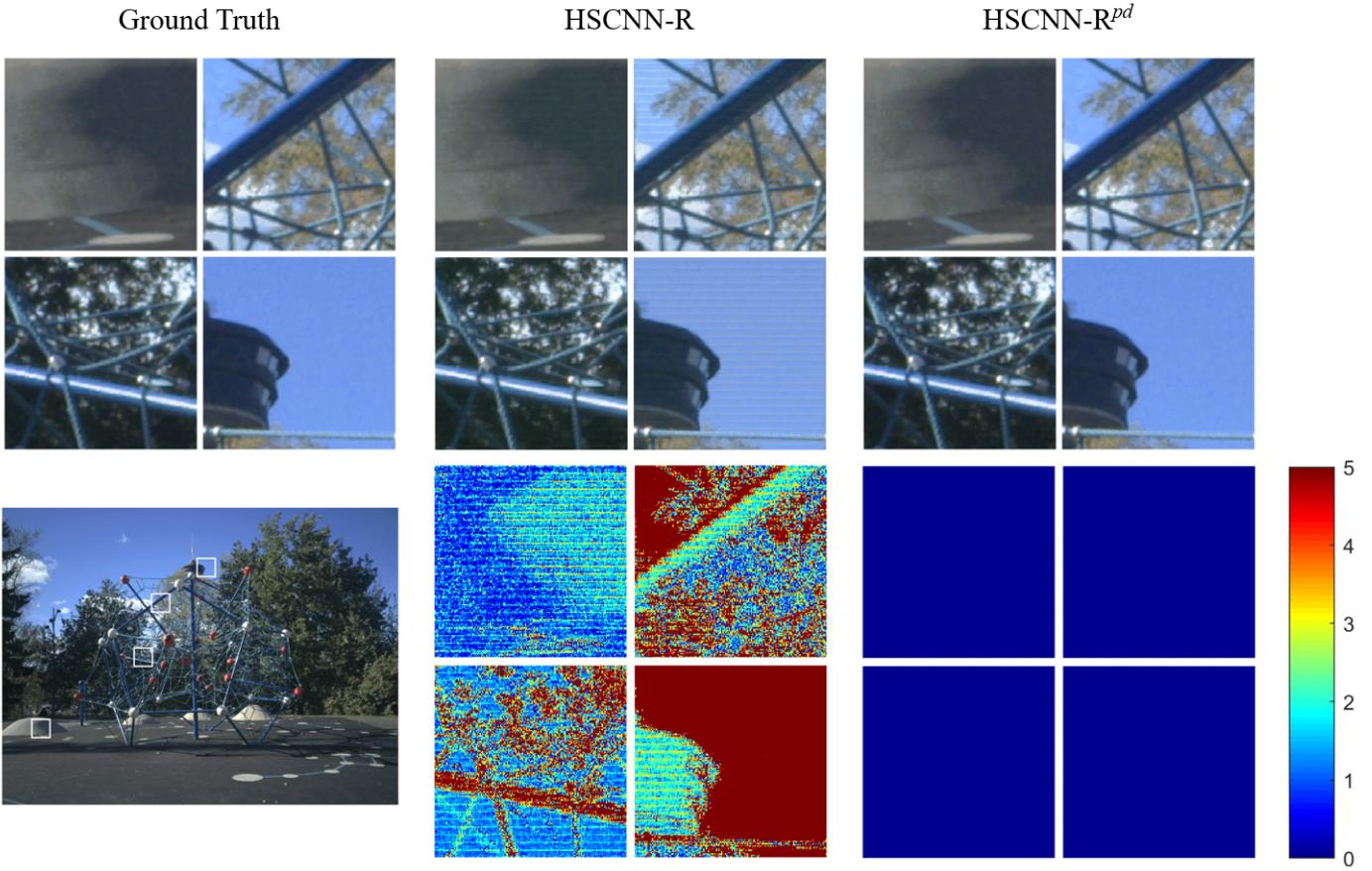


Figure 5: Visual comparison 3. Top row: the rendered images. Bottom row: the corresponding ΔE error maps.

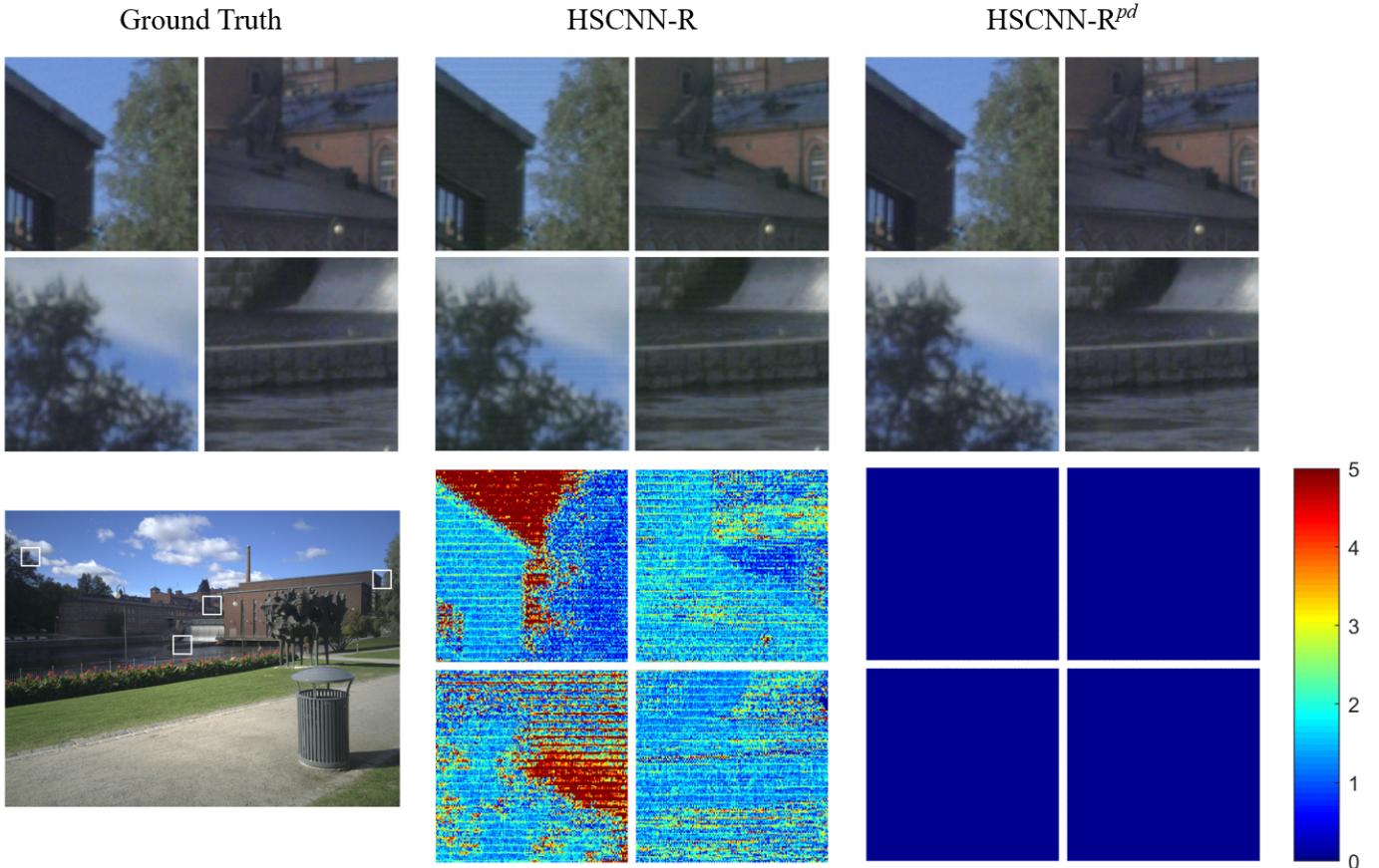


Figure 6: Visual comparison 4. Top row: the rendered images. Bottom row: the corresponding ΔE error maps.

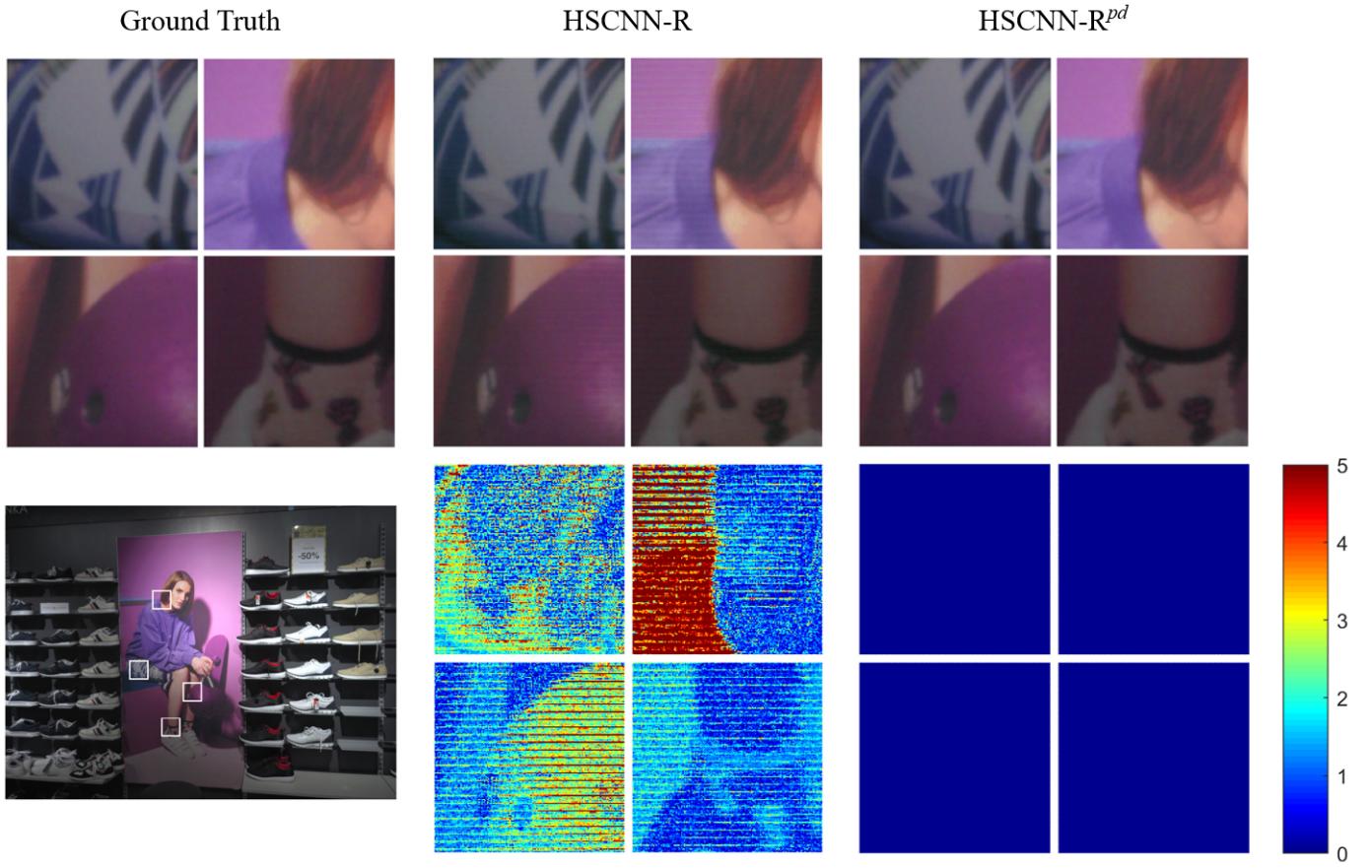


Figure 7: Visual comparison 5. Top row: the rendered images. Bottom row: the corresponding ΔE error maps.

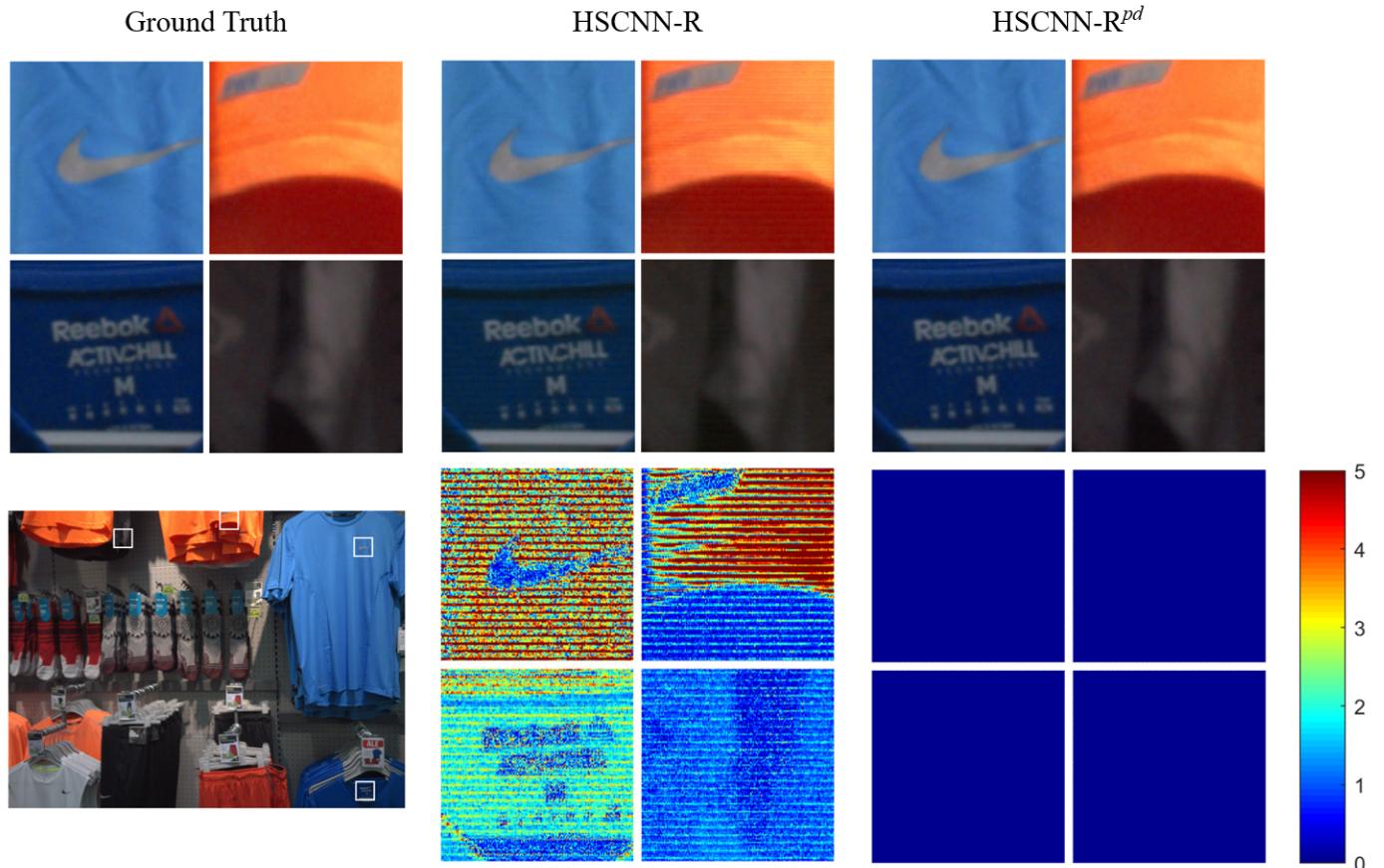


Figure 8: Visual comparison 6. Top row: the rendered images. Bottom row: the corresponding ΔE error maps.