Deep Specialized Network for Illuminant Estimation Supplemental Material

Wu Shi[1], Chen Change Loy[1,2], Xiaoou Tang[1,2]

[1] Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China
[2] Shenzhen Institutes of Advanced Technology, CAS, Shenzhen, China
{sw015, ccloy, xtang}@ie.cuhk.edu.hk

1 Data Preprocessing for HypNet

We apply an additional data preprocessing step for HypNet, which is subtracting per-channel means of a patch from each channel. In this section, we will show that this operation can make the performance of our method stable to a variety of illuminants.

Recall that in our simplified diagonal model, the value of an image pixel I is the product of the illuminant E and the surface reflectance R:

$$I = E \times R. \tag{1}$$

where both E and R are in the RGB space. Then we convert the RGB values to the UV values in the log-homogeneous chrominance and obtain

$$\begin{aligned} I_u &= \log(I_r/I_g) = \log(E_r/E_g) + \log(R_r/R_g) = E_u + R_u, \\ I_v &= E_v + R_v. \end{aligned} \tag{2}$$

Note that the illuminant E is assumed globally or locally (patch-wise) uniform. Therefore, for any patch X, we have $E(x) = E, \forall x \in X$. Subtracting the per-channel means $(\bar{I}_u, \bar{I}_v)$ from that patch will cancel the term of illumination:

$$\begin{aligned} I_u(x) - \bar{I}_u &= (E_u(x) - \bar{E}_u) + (R_u(x) - \bar{R}_u) = R_u(x) - \bar{R}_u, \\ I_v(x) - \bar{I}_v &= R_v(x) - \bar{R}_v, \qquad \forall x \in X. \end{aligned} \tag{3}$$

The actual input data are $\bar{\mathbf{I}} = (I_u - \bar{I}_u, I_v - \bar{I}_v) = (R_u - \bar{R}_u, R_v - \bar{R}_v)$ and the ground truth output becomes $F^*(\bar{\mathbf{I}}) = (E_u - \bar{I}_u, E_v - \bar{I}_v) = (-\bar{R}_u, -\bar{R}_v)$. That is, the input and output for HypNet are independent of the change of illumination. This agrees with the color constancy mechanism in the human vision system, where human can perceive a white-balanced scene without explicitly knowing the color of illuminant. Therefore, the performance of our HypNet is stable to a variety of illuminants. To partially verify this effect, we train our model on the Color Checker dataset and test that model on the NUS 8-camera dataset. That model produces competitive results whose geometric mean of the mean and median angular errors are respectively 2.99 and 2.37.

2 Additional Examples

In this section, we present more examples of our results from the reprocessed [1] Color Checker Dataset [2]. As comparison, the results of three state-of-art methods: Regression Tree [3], CCC [4], and CNN [5] are also shown here. To fairly pick examples as suggested by [4], we sort the 568 images by the average angular errors of the four algorithms, and sample 10 images with evenly spaced indices. All the results can be found in Figures 1-5.

3 Alternative Learning Scheme for HypNet

Besides the winner-take-all learning scheme described in the main paper, we also try an alternative one, which uses a weighted sum of errors as the loss function. Specifically, given an input patch $\mathbf{I}$ and its ground truth illuminant $\mathbf{E}^*$, we obtain two hypotheses of illumination $\mathbf{E}_A, \mathbf{E}_B$ from HypNet, and two scores (s_A, s_B) from SelNet. The loss function for that patch is defined by

$$L(\mathbf{I}; \Theta) = \sum_{k \in \{A,B\}} s_k * (\|\tilde{\mathbf{E}}_k - \mathbf{E}^*\|_2^2), \tag{4}$$

which is the weighted sum of errors for the two branches. The original winner-take-all learning scheme whose loss function is defined by

$$L(\mathbf{I}; \Theta) = \min_{k \in \{A,B\}} (\|\tilde{\mathbf{E}}_k - \mathbf{E}^*\|_2^2), \tag{5}$$

only optimizes the branch with lowest error, while the weighted-sum-of-error scheme (4) takes into account both branches with different penalties. For the model learned using (5), if SelNet makes wrong prediction, the error of the wrong branch is not guaranteed and could be very large. In contrast, for the model learned using (4), the error of the wrong branch is mostly reasonable. In this sense, (4) can be considered as a smoothed version of (5). Though the overall performance using the weighted-sum-of-error learning scheme is not superior to that using the winner-take-all method on the whole dataset, we observe some improvements in terms of pixel-wise angular error on many images. The results of per-pixels angular error map using these two learning schemes respectively can be seen in Figure 6.

References

1. Shi, L., Funt, B.: Re-processed version of the gehler color constancy dataset of 568 images. accessed from http://www.cs.sfu.ca/ colour/data/
2. Gehler, P.V., Rother, C., Blake, A., Minka, T., Sharp, T.: Bayesian color constancy revisited. In: IEEE Conference on Computer Vision and Pattern Recognition. (2008) 1–8

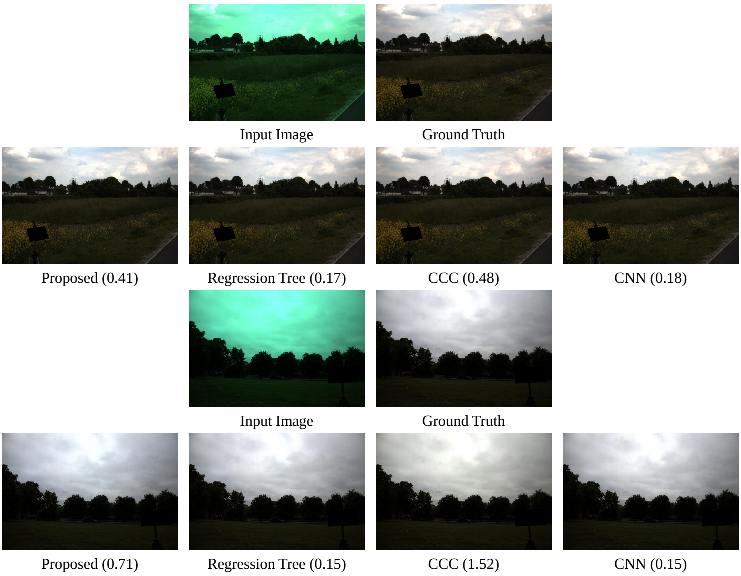

Fig. 1. *Additional examples*: Restored images from the Color Checker dataset using the illuminants estimated from four different methods including the proposed CC-Net, Regression Tree [3], CCC [4] and CNN [5]. The name of method and the corresponding angular error is provided at the bottom of each image. For better visualization we follow [3] to apply gamma function on RAW images and the images are normalize to the 90th percentile.

3. Cheng, D., Price, B., Cohen, S., Brown, M.S.: Effective learning-based illuminant estimation using simple features. In: IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1000–1008
4. Barron, J.T.: Convolutional color constancy. In: IEEE International Conference on Computer Vision. (2015) 379–387
5. Bianco, S., Cusano, C., Schettini, R.: Single and multiple illuminant estimation using convolutional neural networks. arXiv preprint arXiv:1508.00998 (2015)

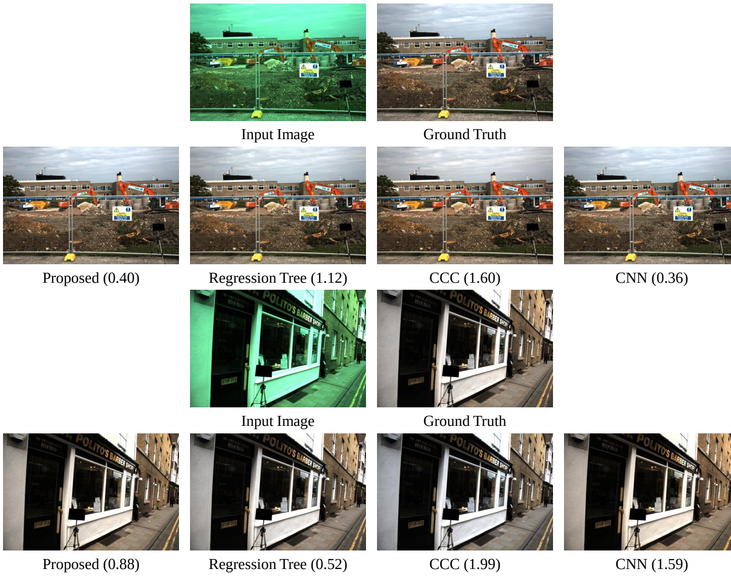

Fig. 2. *Additional examples*: In the same format as Fig. 1

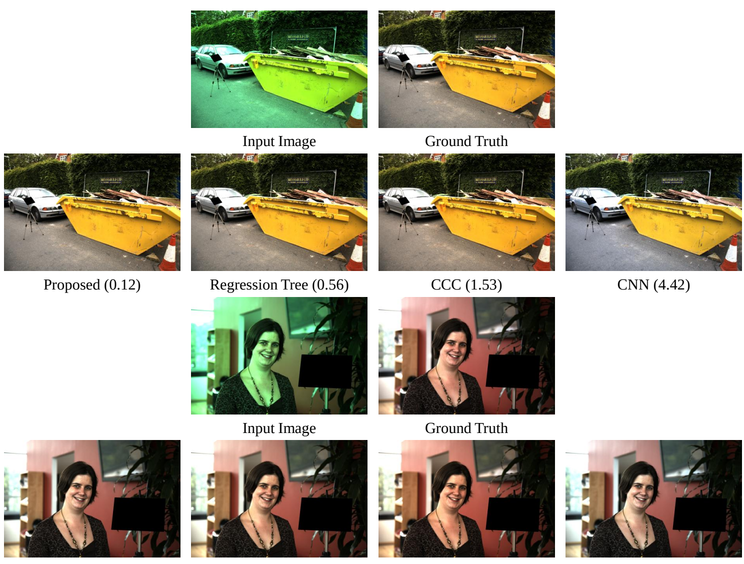

Fig. 3. *Additional examples*: In the same format as Fig. 1

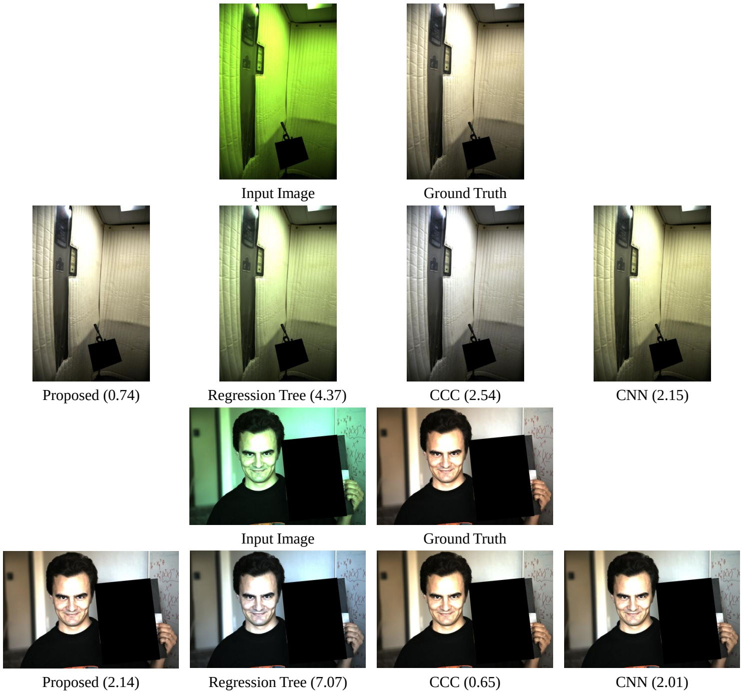

Fig. 4. *Additional examples*: In the same format as Fig. 1

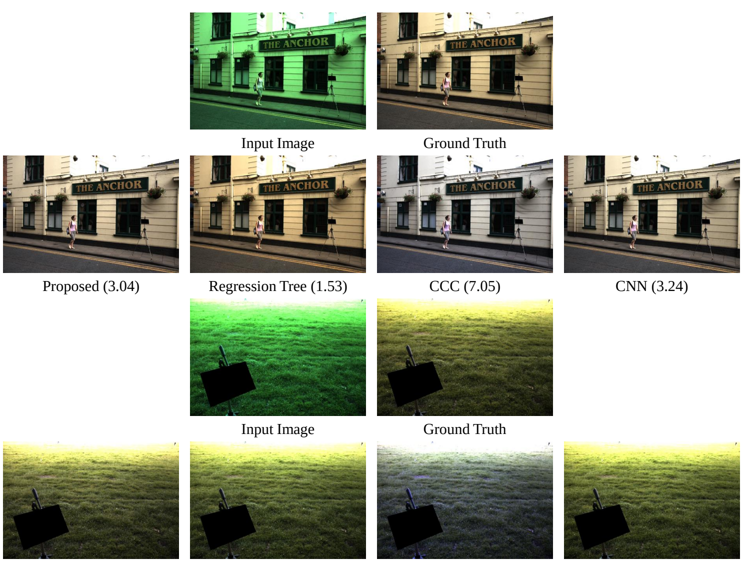

Fig. 5. *Additional examples*: In the same format as Fig. 1

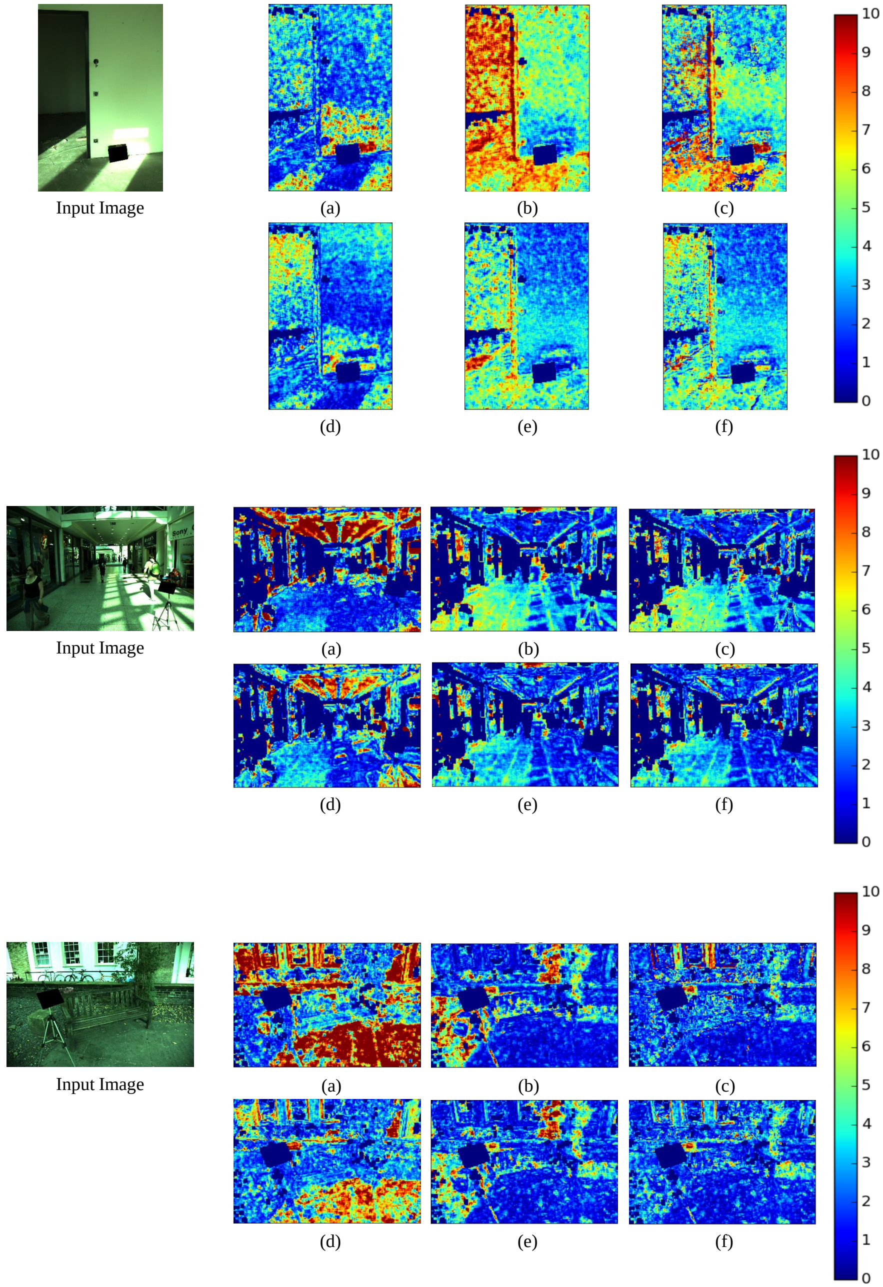

Fig. 6. *Comparison of the results using different learning schemes*: (a, b) The respective per-pixel angular error maps of A-branch and B-branch of HypNet using the winner-take-all learning scheme. (c) CC-Net (HypNet+SelNet) using the winner-take-all learning scheme (d, e) A-branch and B-branch of HypNet using the weighted-sum-of-error learning scheme. (f) CC-Net (HypNet+SelNet) using the weighted-sum-of-error learning scheme. The error bar represents the angular error.