# Supplementary Material for "End-to-End Low Cost Compressive Spectral Imaging with Spatial-Spectral Self-Attention"

Ziyi Meng[1,2][0000−0001−8294−8847], Jiawei Ma[3][0000−0002−8625−5391], and Xin Yuan[4][0000−0002−8311−7524]

[1] Beijing University of Posts and Telecommunications, Beijing, 100876, China, mengziyi@bupt.edu.cn
[2] New Jersey Institute of Technology, Newark, NJ 07102, USA
[3] Columbia University, New York NY 10027, USA, jiawei.m@columbia.edu
[4] Nokia Bell Labs, Murray Hill NJ 07974, USA, xyuan@bell-labs.com

## Contents

**Imbalanced Response of Disperser** ..... 2
**Network Implementation** ..... 3
**Hardware Setting** ..... 5
**Attention Visualization** ..... 6
**Noise Analysis** ..... 7
**Results** ..... 8

### *Attached Videos*

**File name "Video_lego.mp4"**
The video contains 35 frames. (Left) the measurement captured at 35 fps by our real SD-CASSI system; (Right) the reconstructed frame with 28 spectral channels by the proposed TSA-Net. The object is moving from left to right. We play this video at 10 fps for better visualization.

**File name "Video_plant.mp4"**
The video contains 105 frames. (Left) the measurement captured at 35 fps by our real SD-CASSI system; (Right) the reconstructed frame with 28 spectral channels by the proposed TSA-Net. The object is rotating counterclockwise. We play this video at 10 fps for better visualization.

# 1 Imbalanced Response of Disperser

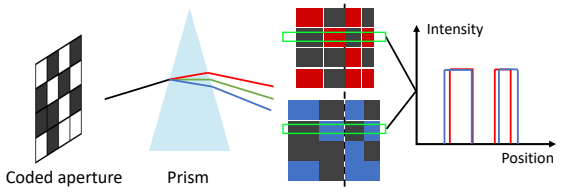


**Fig. M1.** Imbalanced response of disperser (prism). As a prism causes light deflection, length difference exists between every two pixels' optical paths: 1) Distortion, *i.e.*, horizontal stretch and compression, exists in the modulated frames. 2) The distortion degree varies due to the deflection variation across all wavelengths.

The SD-CASSI system has been developed before but suffers from the imbalanced response (Fig. M1) and low quality reconstruction.

As shown in Fig. M1, the distortion behaves as the stretch and compression of the left and right field of view on dispersion direction, which is caused by the light deflection generated by the prism. Furthermore, the distortion varies for different wavelength due to the existence of dispersion, thus causing a serious calibration error of mask when only using a single wavelength to do calibration. In addition, the distortion is amplified as the increasing of the measurement scale. This problem causes performance drop in real data. This is the motivation that we invoke deep learning to mitigate the problem.

## 2 Network Implementation

### 2.1 Dataset

**CAVE** [7] has 30 hyperspectral images and the spatial size of each sample is $512 \times 512$. For model training, the data augmentation methods include scaling, rotation and concatenation among the samples. For the *concatenation*, we randomly align the images in a $2 \times 2$ or larger array to have images with large scale. Then, we select the spatial region of a fixed size and crop the corresponding hyperspectral cube to generate one sample for model training/validation. The scaling and rotation is applied on the hyperspectral images before concatenation.

**KAIST** [2] has 30 hyperspectral images and the spatial size of each sample is $2704 \times 3376$. We adopt rotation and scaling for data augmentation. Then, we select the spatial region of a fixed size and crop the corresponding hyperspectral cube to generate one sample.

***Simulation***. 30 scenes from **CAVE** were used for model training. In total, we have created 4000 samples for model training and 50 samples for model validation. 10 scenes, never been seen in model training, from **KAIST** are used for model testing. The spatial size of all samples are $256 \times 256$.

***Real Data Experiment***. All scenes from **CAVE** and from **KAIST** are used for model training. In total, we have created 5000 samples for model training and 100 samples for model validation. The spatial size of all samples is $660 \times 660$.

### 2.2 Hyper-parameters

As shown in Fig.4 in the main paper, there are 5 blocks in encoder and decoder respectively. We use $(k, o, n, p)$ to denote the kernel size, number of kernels, number of layers in each block and the pooling/upsampling size. We set two convolution layers in the bottleneck and the structure of each layer is denoted by $B(k, o, p)$.

***Simulation***. The batch size is 10 and the model training stops at the 250-th epoch. The network structure is: [(3,64,3,2), (3,128,3,2), (3,256,3,2), (3,512,3,2), (3,1024,3,2)], B(3,1280,3).

***Experiment***. The batch size is 5 and the model training stops at the 140-th epoch. The network structure is: [(3,64,3,3), (3,128,3,2), (3,256,3,2), (3,512,3,2), (3,1024,3,2)], B(3,1280,3).

***Spatial-Spectral Self-Attention Module***. As shown in the Fig. 6 in the main paper, there are 3 TSA-Modules added in the decoder. The TSA-Module added at the top of decoder is used to generate the final output. There is one $1 \times 1$ convolution layer applied on module input to change the size along the spectral dimension as 28. The number of heads is set to 4 while the dimension-specific vector are of size $(f_x, f_y, f_\lambda) = (48, 48, 30)$. For the other TSA-Module, There are 8 heads and the dimension-specific vector are of size $(f_x, f_y, f_\lambda) = (64, 64, 128)$.

### 2.3 Metric

Suppose $\mathcal{X} \in \mathcal{R}^{W\times H\times C}$ represents the ground truth and $\hat{\mathcal{X}} \in \mathcal{R}^{W\times H\times C}$ represents the predicted value.
Root Mean Square Error (RMSE)

$$\mathrm{RMSE}(\mathcal{X},\hat{\mathcal{X}}) = \sqrt{\frac{1}{WHC}\sum_{w=1}^{W}\sum_{h=1}^{H}\sum_{c=1}^{C}(\mathcal{X}(w,h,c)-\hat{\mathcal{X}}(w,h,c))^2}.$$

Mean Squared Error (MSE)

$$\mathrm{MSE}(\mathcal{X},\hat{\mathcal{X}}) = \frac{1}{WHC}\sum_{w=1}^{W}\sum_{h=1}^{H}\sum_{c=1}^{C}(\mathcal{X}(w,h,c)-\hat{\mathcal{X}}(w,h,c))^2.$$

Spectrum Constancy Loss

$$\mathrm{SpecCon}(\mathcal{X},\hat{\mathcal{X}}) = \frac{1}{WH(C-1)}\sum_{c=2}^{C}\sum_{w=1}^{W}\sum_{h=1}^{H}\left(\nabla_c\mathcal{X}(w,h,c)-\nabla_c\hat{\mathcal{X}}(w,h,c)\right)^2.$$

where $\nabla_c\mathcal{X}(w,h,c) = \mathcal{X}(w,h,c)-\mathcal{X}(w,h,c-1)$.
Peak Signal-to-Noise Ratio (PSNR)

$$\mathrm{PSNR}(\mathcal{X},\hat{\mathcal{X}}) = 10\log(\mathrm{Max}(\mathcal{X})/\mathrm{MSE}(\mathcal{X},\hat{\mathcal{X}})).$$

Structural Similarity Index Metrics (SSIM)

$$\mathrm{SSIM}(\mathcal{X},\hat{\mathcal{X}}) = \left[\frac{2\mu_{\mathcal{X}}\mu_{\hat{\mathcal{X}}}+C_1}{\mu_{\mathcal{X}}^2+\mu_{\hat{\mathcal{X}}}^2+C_1}\right]^{\alpha}\left[\frac{2\sigma_{\mathcal{X}}\sigma_{\hat{\mathcal{X}}}+C_2}{\sigma_{\mathcal{X}}^2+\sigma_{\hat{\mathcal{X}}}^2+C_2}\right]^{\beta}\left[\frac{\sigma_{\mathcal{X}\hat{\mathcal{X}}}+C_3}{\sigma_{\mathcal{X}}\sigma_{\hat{\mathcal{X}}}+C_3}\right]^{\gamma}.$$

where $\mu$ denotes the mean and $\sigma$ denotes the standard deviation / cross-covariance. The value of $\alpha,\beta,\gamma$ is default in MATLAB R2018a.

### 2.4 Optimizer & Learning Rate

We use the Adam optimizer [3] and the learning rate in each epoch, indexed by $e$, is set as

$$lr(e) = r_0 \times \alpha^{ceil(\max(0,e-d)/i)}.$$

In simulation, $(r_o,\alpha,d,i) = (0.01,0.8,30,30)$. In experiment, $(r_o,\alpha,d,i) = (0.01,0.8,20,20)$.

# 3 Hardware Setting

## 3.1 Hardware

The hardware system consists of a 18mm objective lens (Olympus RMS10X), a coded aperture (mask), two relay lens (Olympus RMS4X 45mm and Thorlabs AC254-050-A-ML 50mm), a dispersive prism (Edmund 43649) and a detector (Basler acA2000-340km). The detector has $2048 \times 1088$ pixels with the pixel pitch of 5.5 $\mu$m and supports 12-bit depth.

## 3.2 Wavelength

The system is calibrated by a 660nm laser. Two edgepass filters are used to limit the spectral range of the system to 450-650 nm. In this range, the prism with 30° apex produces 54-pixel dispersion. By calibrate the prism, we determine the 28 spectral channels with wavelengths: {453.3, 457.6, 462.1, 466.8, 471.6, 476.5, 481.6, 486.9, 492.4, 498.0, 503.9, 509.9, 516.2, 522.7, 529.5, 536.5, 543.8, 551.4, 558.6, 567.5, 575.3, 584.3, 594.4, 604.2, 614.4, 625.1, 636.3, 648.1}nm.

## 3.3 Mask capture

The mask is captured by a uniform illumination with a 450nm source. The physical size of the smallest mask code feature is twice larger than the detector pixel (5.5$\mu$m ). However, due to the different focal length of the two relay lens (45mm and 50mm), the smallest mask code feature on the plane of the detector is slightly larger than $2 \times 2$ pixels of detector.

## 4 Attention Visualization

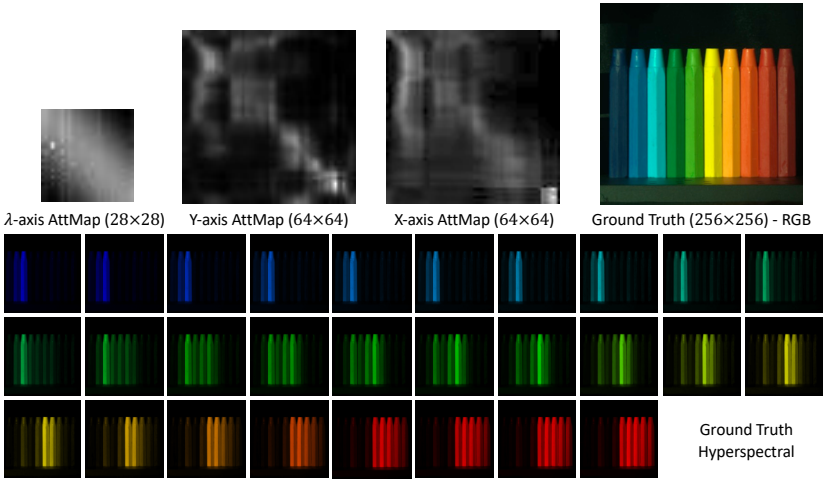


**Fig. M2.** Attention maps visualization when reconstructing one sample

We visualize TSA by extracting the attention maps in the last TSA module.

Spectral: We reconstruct images with 28 channels. For each channel, the attention from its neighbor channel (including the channel it self) is usually stronger than the attention from the distant channels. As shown, each pixel $(m, n)$ indicates the attention from channel $n$ to channel $m$ and the pixels close to the diagonal are usually have higher intensity. Comparing with the hyperspectral image ground truth, the channel-wise attention value can be used to indicate the correlation between two channels, i.e., high attention value indicates the global/spatial image patterns of the corresponding two channels are similar to each other, and low attention value indicates the image patterns of the corresponding two channels are different from each other.

Spatial: The spatial attention is represented by the attention map for $X$-axis (row-region-wise) and the attention map for $Y$-axis (column-region-wise). Due to the limitation of GPU memory, we use attention map to represent the correlation between the 4-by-4 non-overlapping regions to have more attention heads and learn to model the correlation more comprehensively. Each region is cropped by a 4-unit non-overlapping window. The pixel-wise attention map can be estimated by the tensor product of these two attention maps in the same head. As shown in Fig. M2, for the pixels of the same color (or similar color, e.g., red and orange) in RGB image, the attention among the corresponding columns is higher than the columns of different color (e.g., red and blue) in RGB image. In contract, since the image patterns in different row regions are similar with each other, the intensities in $X$-axis map are not diagonal-centered.

## 5 Noise Analysis

We have performed a noise analysis to campare the robustness of TSA-Net traind with shot noise and Gaussian noise. We trained a TSA-Net on data stained by shot noise (SN, 11 bit) and keep the same signal-to-noise ratio (SNR) with data trained by Gaussian noise (GN, N(0, 0.011)) for fair comparison. We tested the models on data with different level shot noise and Gaussian noise, and the results are listed in Table. M1 and Table. M2, respectively. We observe that, for all testing on data with shot noise, the performance of TSA-Net w/SN is better than that of TSA-Net w/GN. When the SNR of training data is larger than the SNR in testing, TSA-Net w/SN is better and robust to GN. Otherwise, there are significant performance drop for both model while TSA-Net w/GN is a bit better.

We further compare the results of TSA-Nets w/SN and w/GN on real data, which have the same SNR, as shown in Fig. M3. It can be seen that the TSA-Net w/SN reconstructions have less artifacts, which shows injecting shot-noise in training is more effective than adding Gaussian noise for our system.

**Table M1.** Results of testing on data with different level shot noise

| Testing noise | w/o noise | 12-bit SN | 11-bit SN | 10-bit SN |
|---|---|---|---|---|
| TSA-Net w/SN | 28.69, 0.859 | 28.55, 0.856 | 28.35, 0.849 | 28.08, 0.841 |
| TSA-Net w/GN | 28.03, 0.835 | 27.91, 0.828 | 27.79, 0.824 | 27.53, 0.818 |

**Table M2.** Results of testing on data with different level Gaussian noise

| Testing noise | w/o noise | GN($\sigma = 0.01$) | GN($\sigma = 0.02$) | GN($\sigma = 0.05$) | GN($\sigma = 0.1$) |
|---|---|---|---|---|---|
| TSA-Net w/SN | 28.69, 0.859 | 27.89, 0.833 | 26.85, 0.781 | 23.50, 0.660 | 19.67, 0.528 |
| TSA-Net w/GN | 28.03, 0.835 | 27.71, 0.829 | 27.24, 0.803 | 24.58, 0.709 | 20.12, 0.542 |

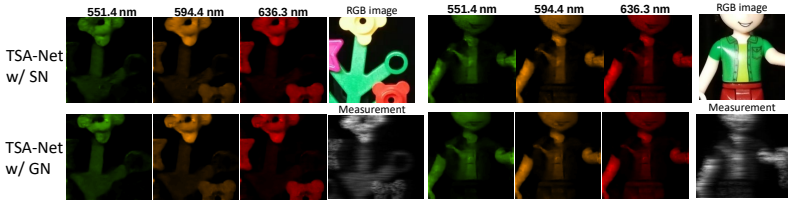


**Fig. M3.** Real data: The reconstructed images ($256 \times 256$) using TSA-Net trained with shot noise and Gaussian noise.

# 6 Results

## 6.1 Simulation Results

Fig. M4-M13 show the simulation results with 28 spectral channels for 10 scenes from KAIST. Truth, measurements, and RGB images are shown for reference. We compare the proposed TSA-Net with the $\lambda$-net [5] and HSSP [6] algorithms and list the corresponding PSNR and SSIM.

## 6.2 Real Data

Fig. M14-M17 show the RGB images, measurements and the reconstructed 28 spectral channels for four real scenes with a size of $660 \times 660$ pixels captured by our system. We compare the proposed TSA-Net with TwIST [1], GAP-TV [8], and DeSCI [4] algorithms.

Fig. M18-M19 show the RGB images, measurements and the reconstructed 28 spectral channels for two scenes with a size of $256 \times 256$ pixels croped from the captured large scale scenes. We compare the results of deep learning methods, including TSA-Net trained with and without shot noise, the $\lambda$-net and HSSP algorithms.

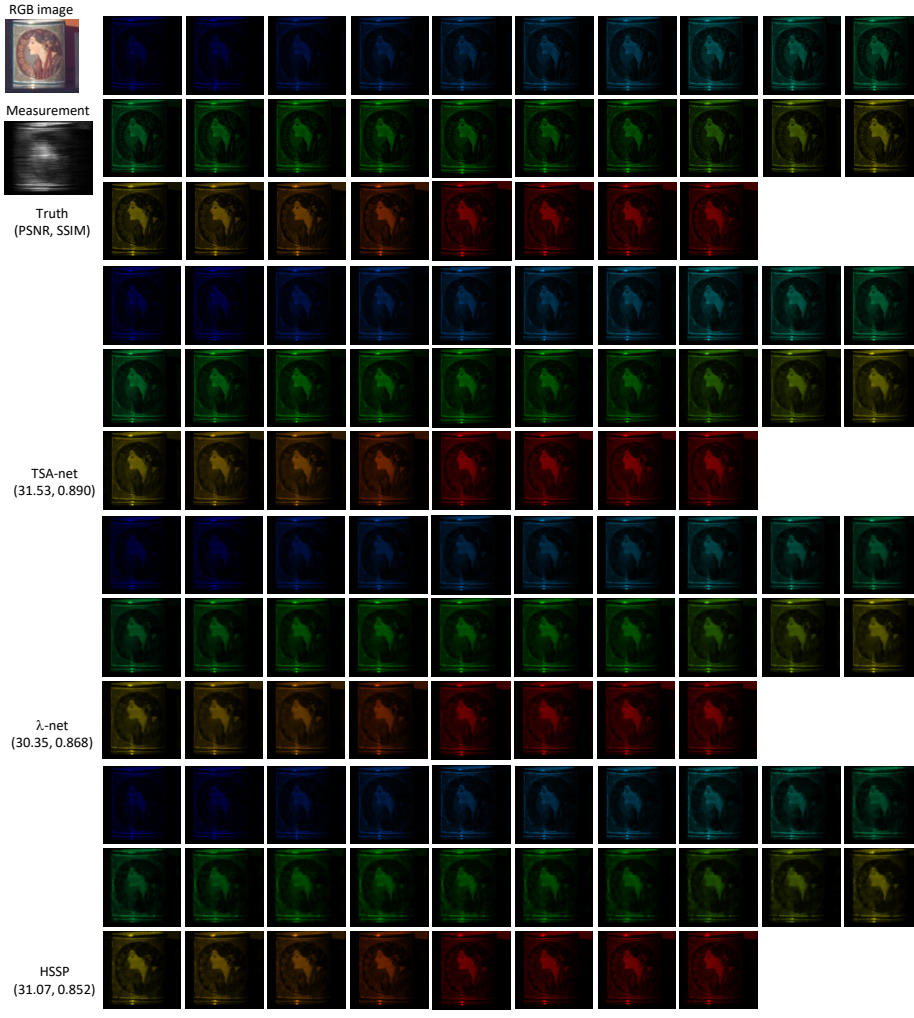


**Fig. M4.** Simulation: RGB image, measurement, ground truth and reconstructed results by TSA-Net, $\lambda$-net, and HSSP for Scene 1. The PSNR in dB and SSIM for the result images are shown in the parenthesis.

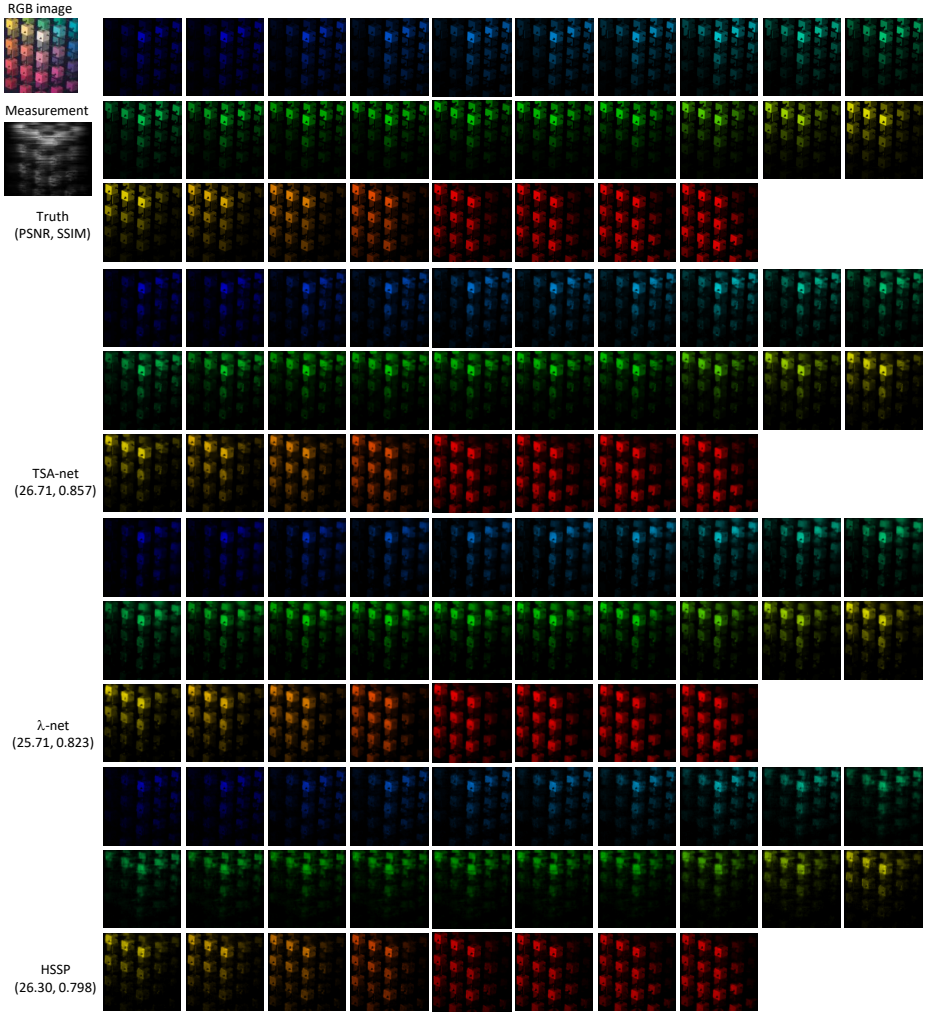


**Fig. M5.** Simulation: RGB image, measurement, ground truth and reconstructed results by TSA-Net, $\lambda$-net, and HSSP for Scene 2. The PSNR in dB and SSIM for the result images are shown in the parenthesis.

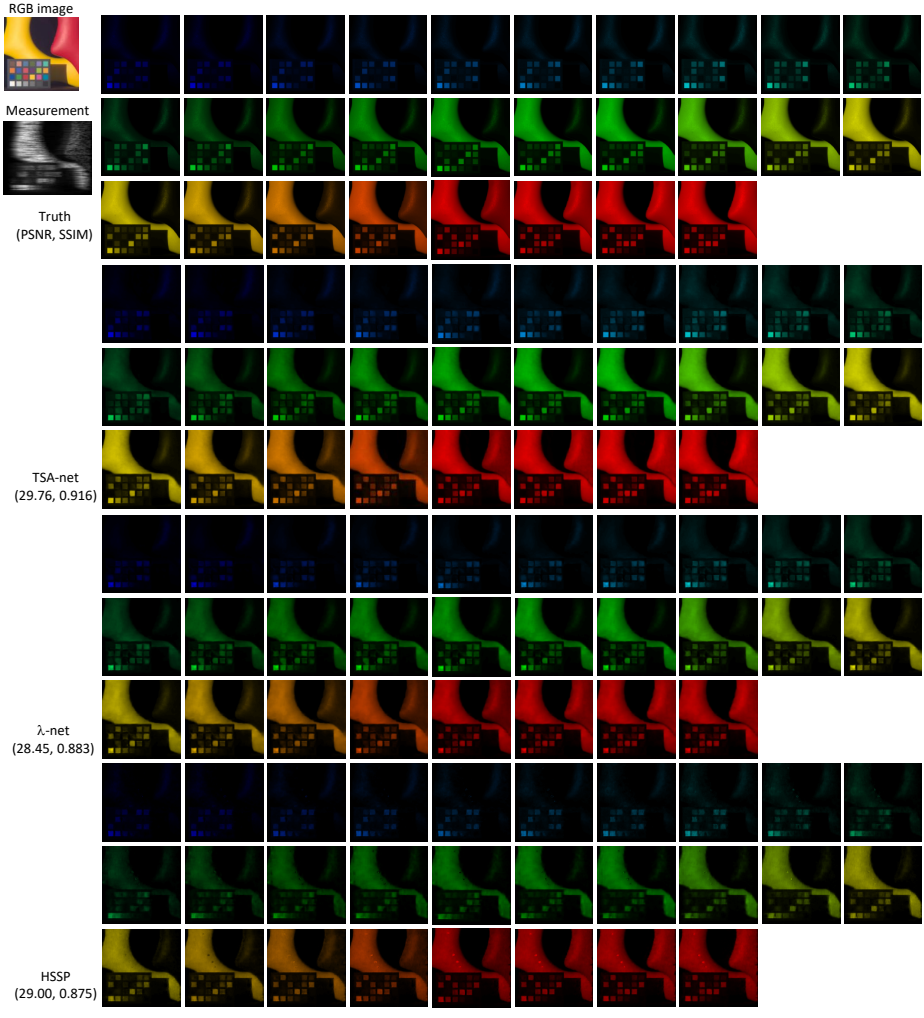


**Fig. M6.** Simulation: RGB image, measurement, ground truth and reconstructed results by TSA-Net, $\lambda$-net, and HSSP for Scene 3. The PSNR in dB and SSIM for the result images are shown in the parenthesis.

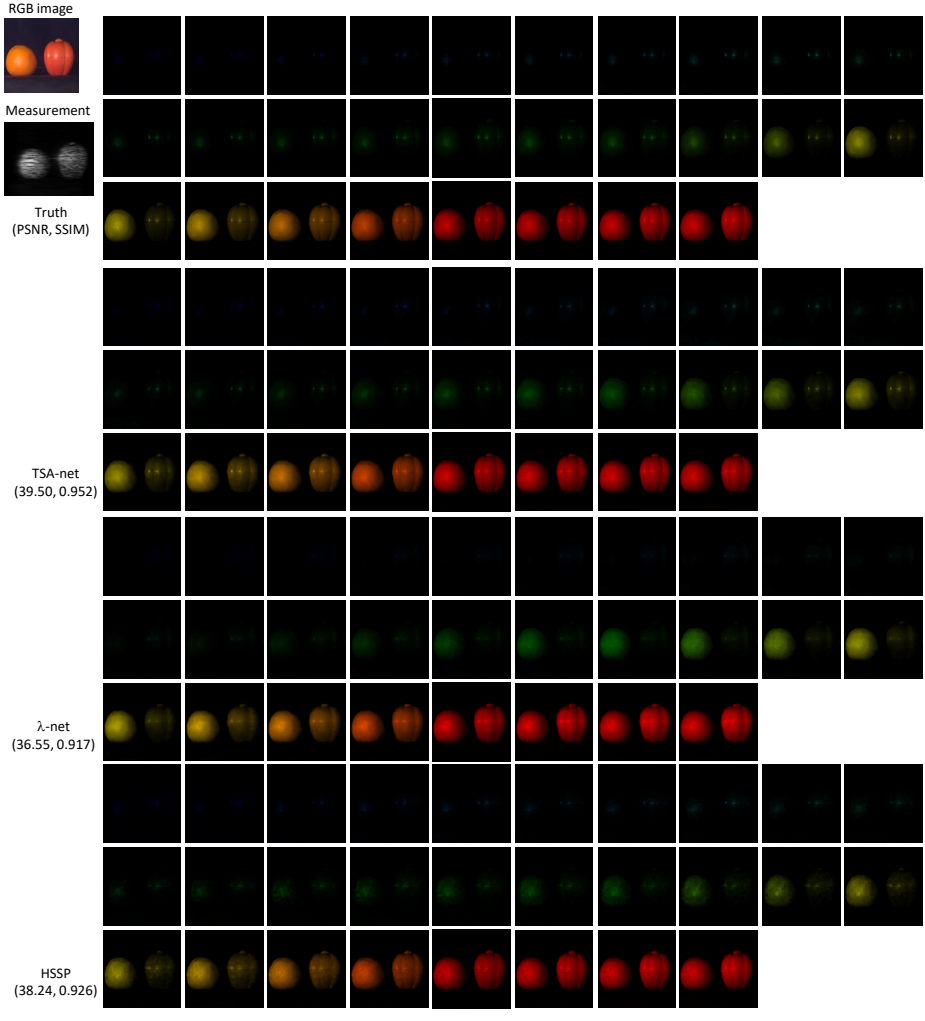


**Fig. M7.** Simulation: RGB image, measurement, ground truth and reconstructed results by TSA-Net, $\lambda$-net, and HSSP for Scene 4. The PSNR in dB and SSIM for the result images are shown in the parenthesis.

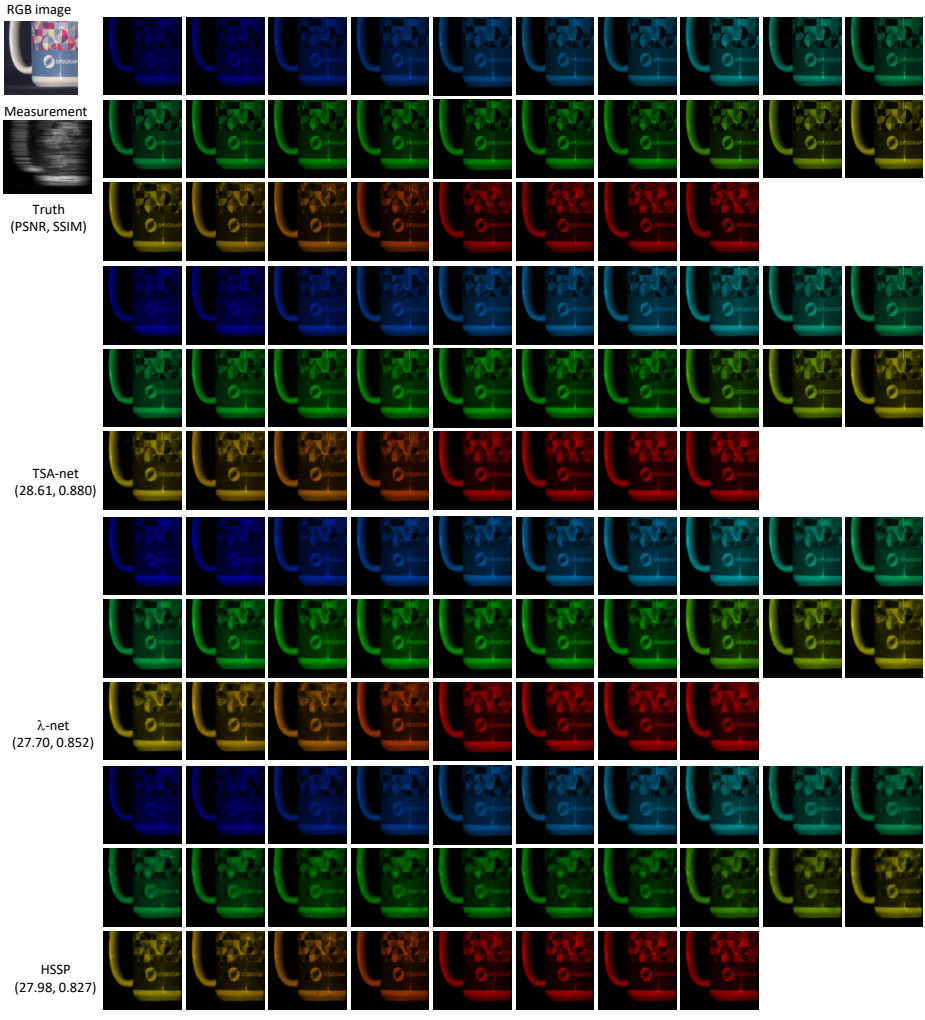


**Fig. M8.** Simulation: RGB image, measurement, ground truth and reconstructed results by TSA-Net, $\lambda$-net, and HSSP for Scene 5. The PSNR in dB and SSIM for the result images are shown in the parenthesis.

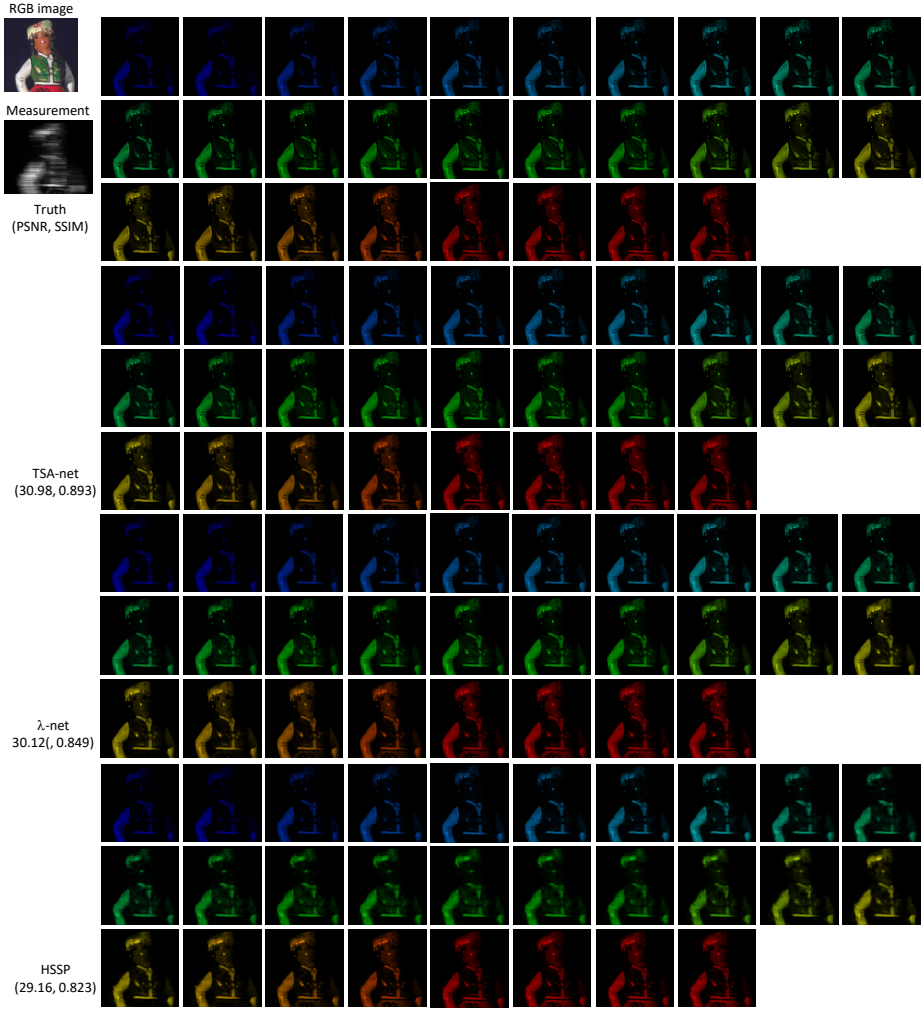


**Fig. M9.** Simulation: RGB image, measurement, ground truth and reconstructed results by TSA-Net, $\lambda$-net, and HSSP for Scene 6. The PSNR in dB and SSIM for the result images are shown in the parenthesis.

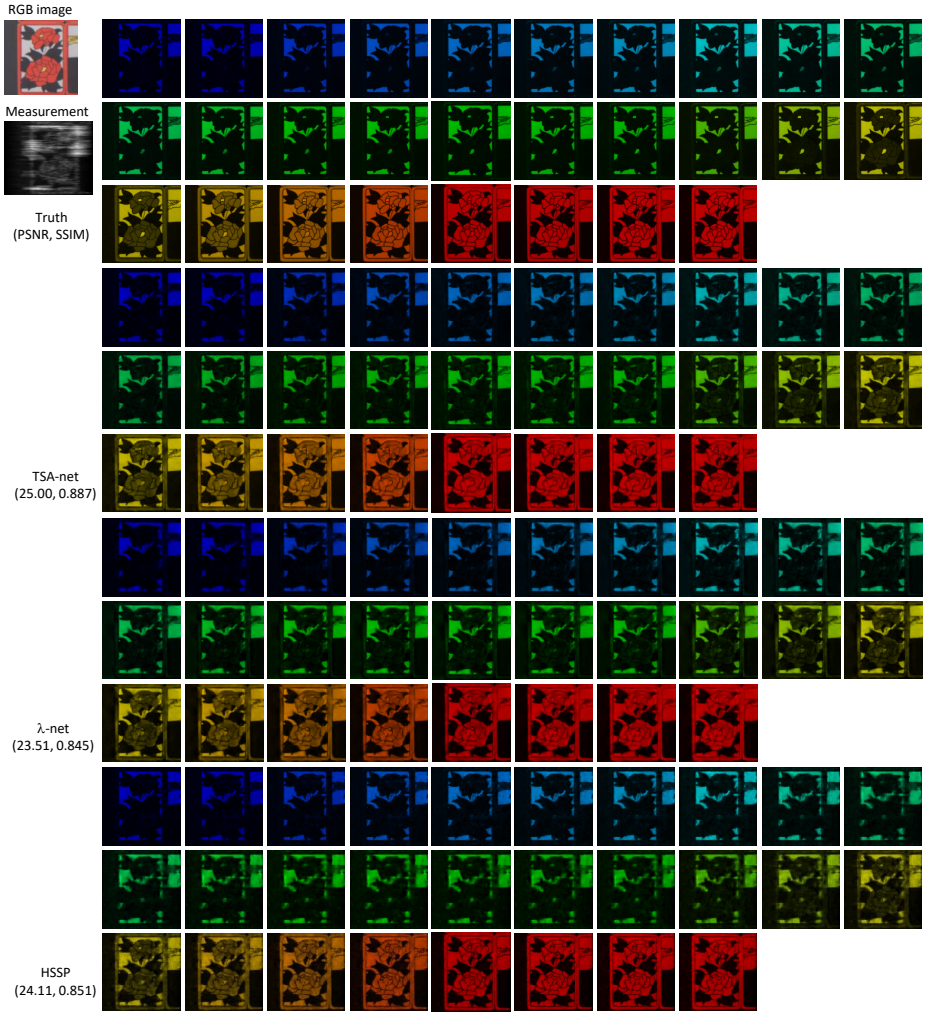


**Fig. M10.** Simulation: RGB image, measurement, ground truth and reconstructed results by TSA-Net, $\lambda$-net, and HSSP for Scene 7. The PSNR in dB and SSIM for the result images are shown in the parenthesis.

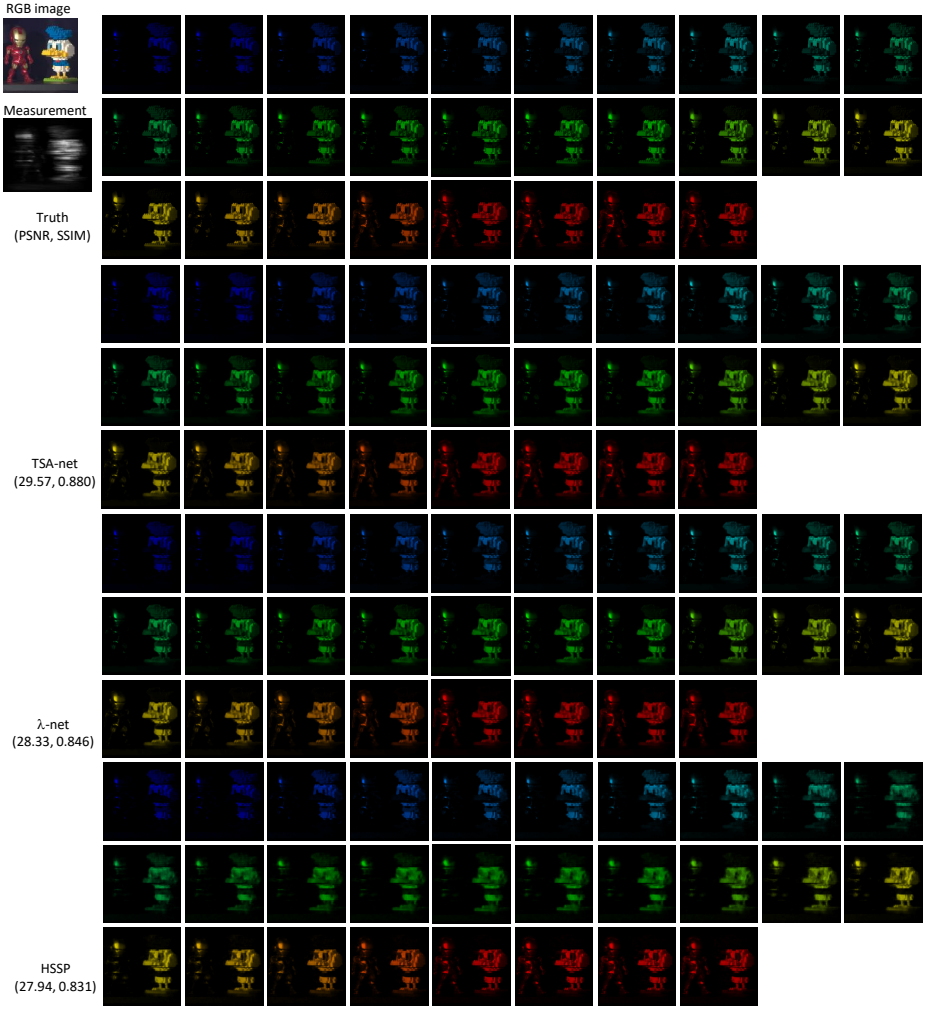


**Fig. M11.** Simulation: RGB image, measurement, ground truth and reconstructed results by TSA-Net, $\lambda$-net, and HSSP for Scene 8. The PSNR in dB and SSIM for the result images are shown in the parenthesis.

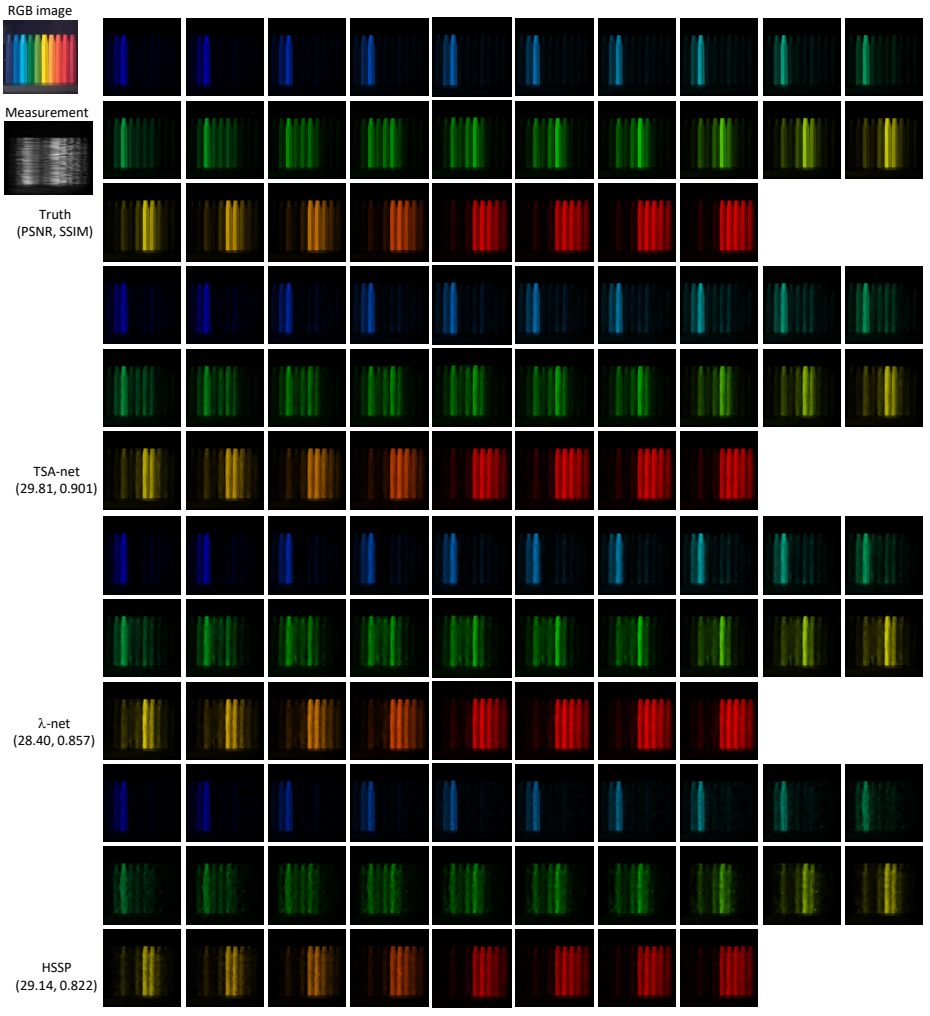


**Fig. M12.** Simulation: RGB image, measurement, ground truth and reconstructed results by TSA-Net, $\lambda$-net, and HSSP for Scene 9. The PSNR in dB and SSIM for the result images are shown in the parenthesis.

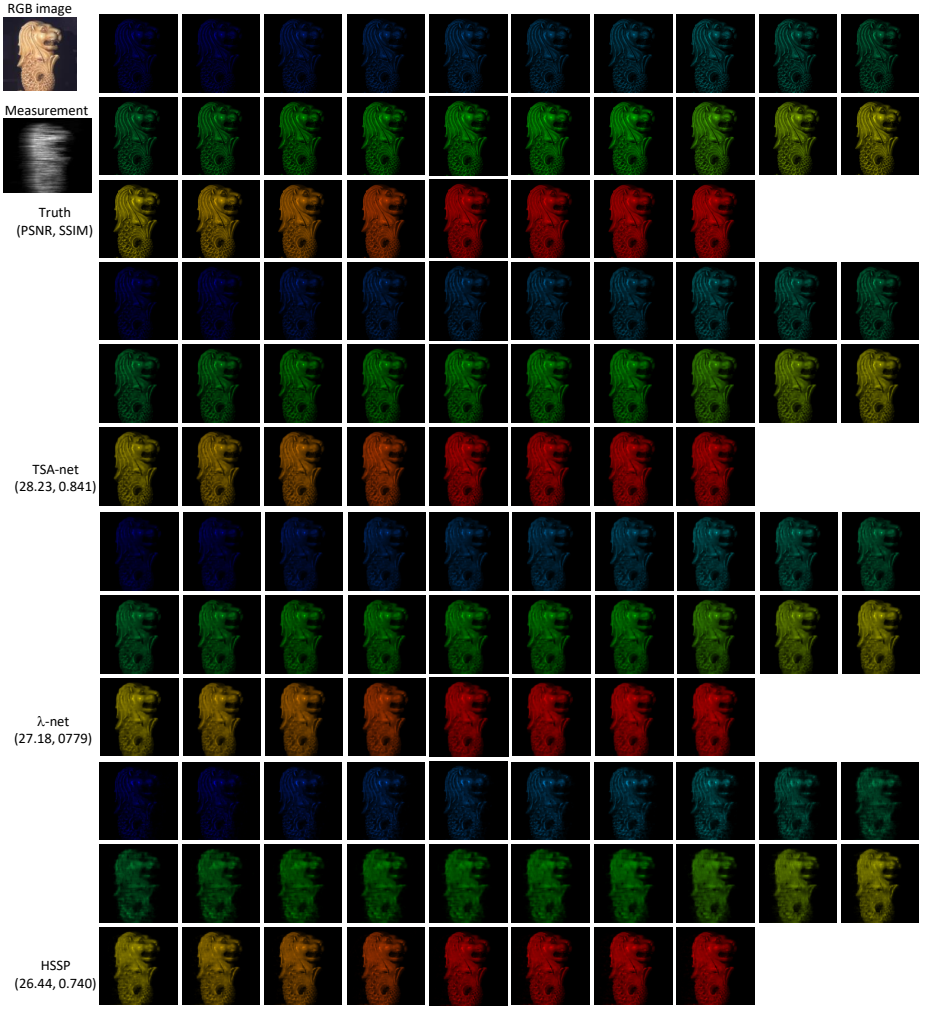


**Fig. M13.** Simulation: RGB image, measurement, ground truth and reconstructed results by TSA-Net, $\lambda$-net, and HSSP for Scene 10. The PSNR in dB and SSIM for the result images are shown in the parenthesis.

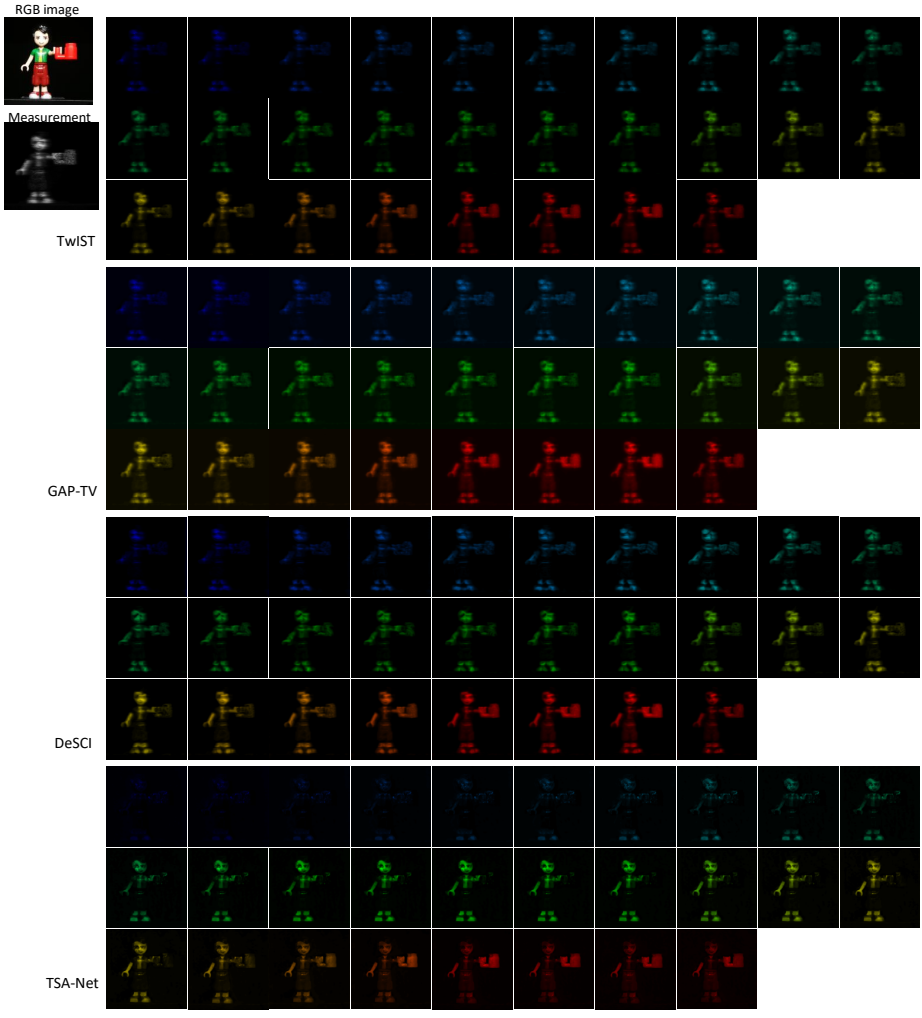


**Fig. M14.** Real data: RGB image, measurement and reconstructed results by TwIST, GAP-TV, DeSCI and TSA-Net for scene 1.

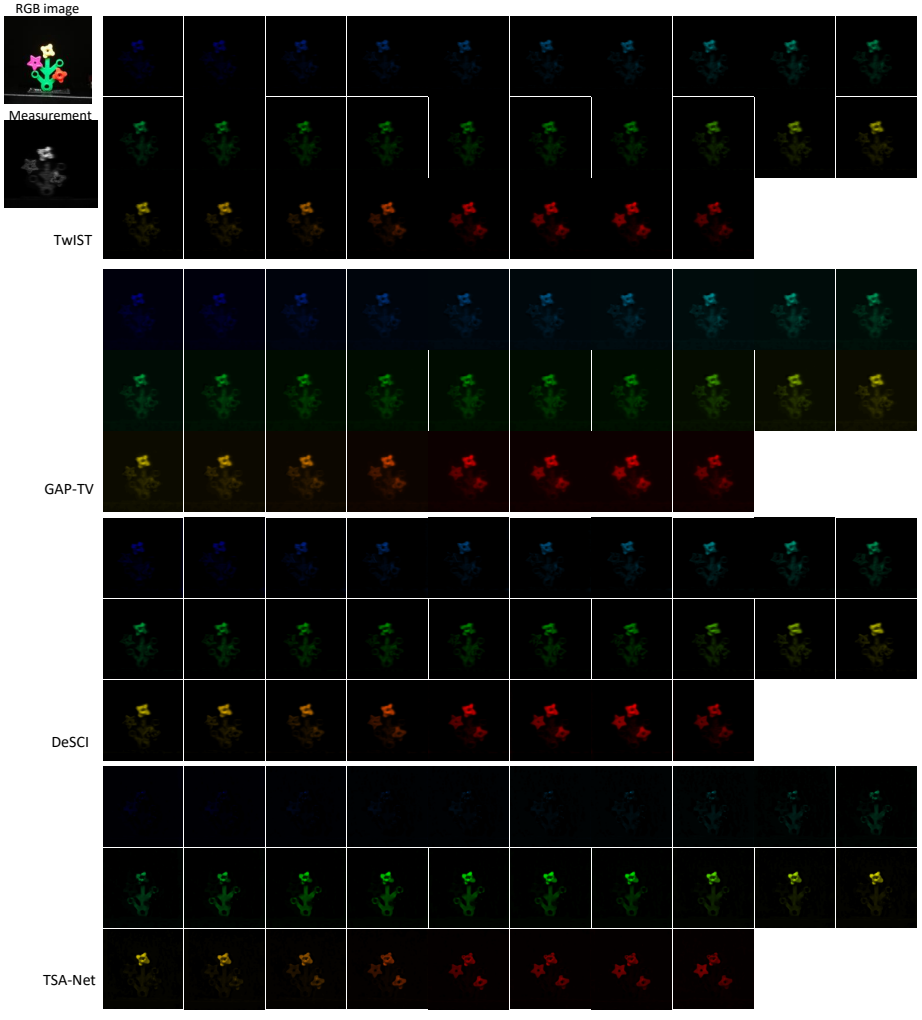


**Fig. M15.** Real data: RGB image, measurement and reconstructed results by TwIST, GAP-TV, DeSCI and TSA-Net for scene 2. Please refer to the video files "Video_lego.mp4".

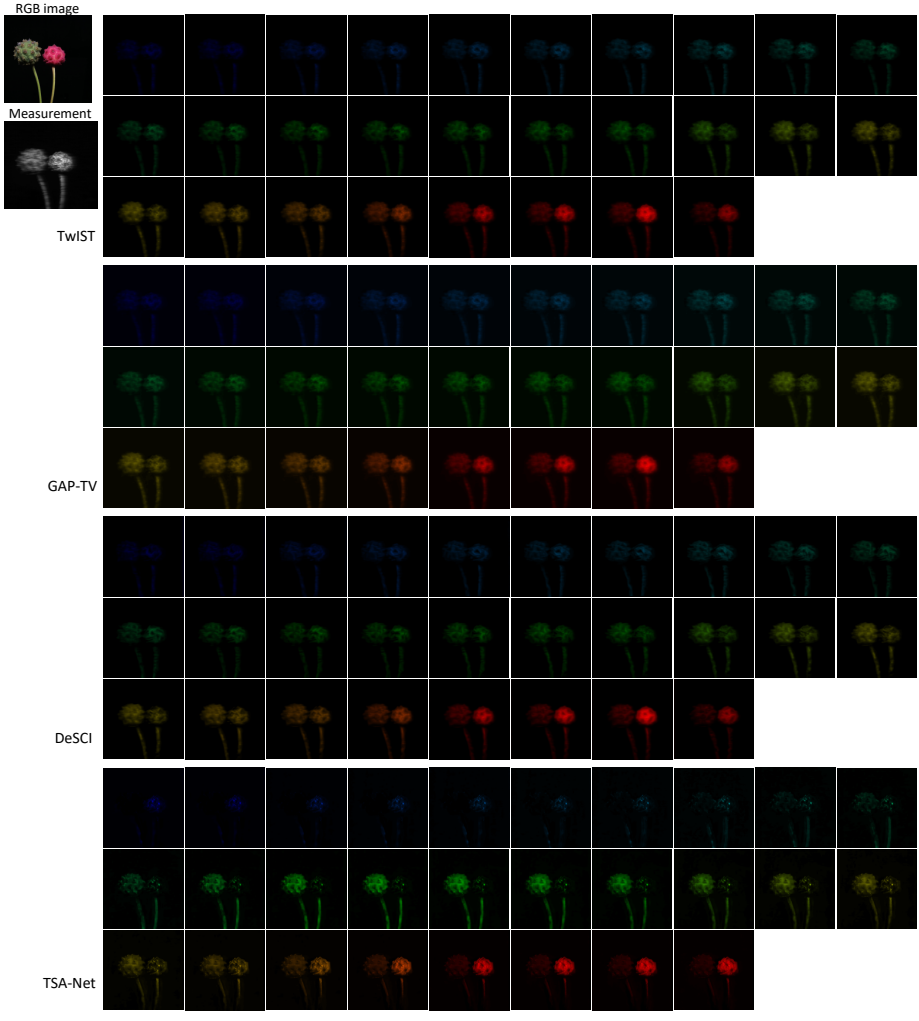


**Fig. M16.** Real data: RGB image, measurement and reconstructed results by TwIST, GAP-TV, DeSCI and TSA-Net for scene 3. Please refer to the video files "Video_plant.mp4".

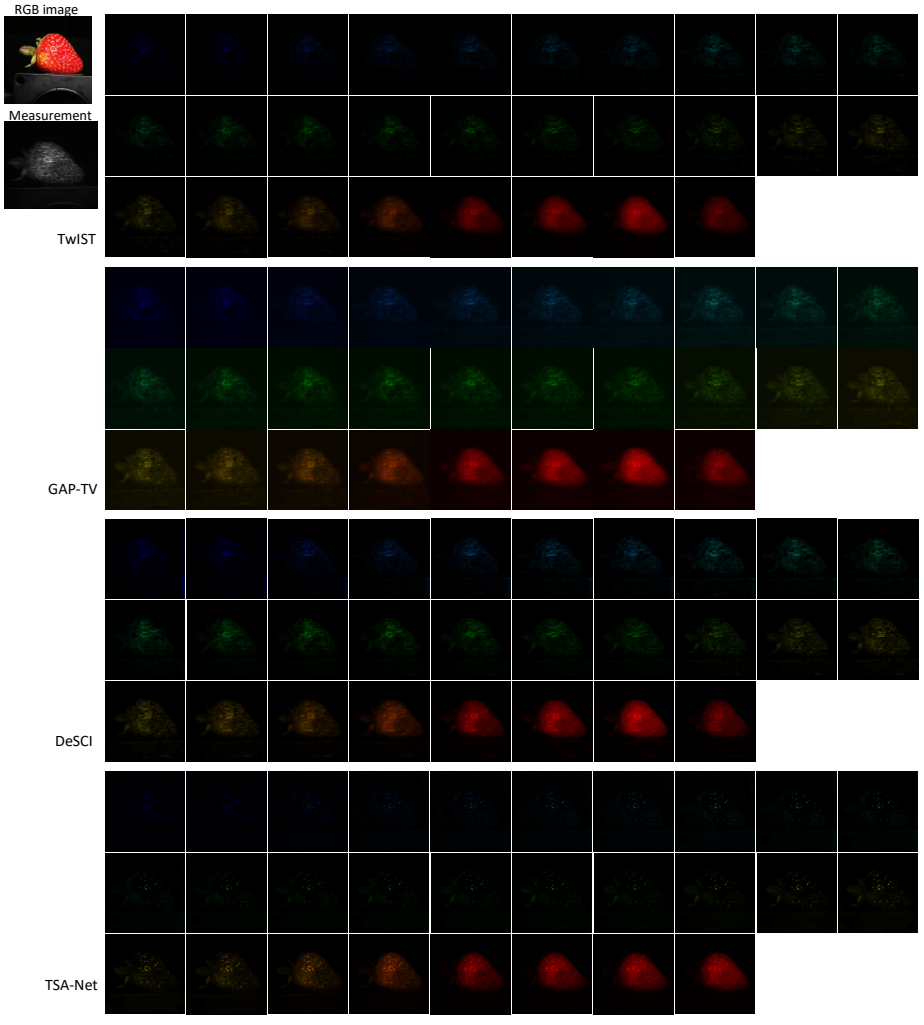


**Fig. M17.** Real data: RGB image, measurement and reconstructed results by TwIST, GAP-TV, DeSCI and TSA-Net for scene 4.

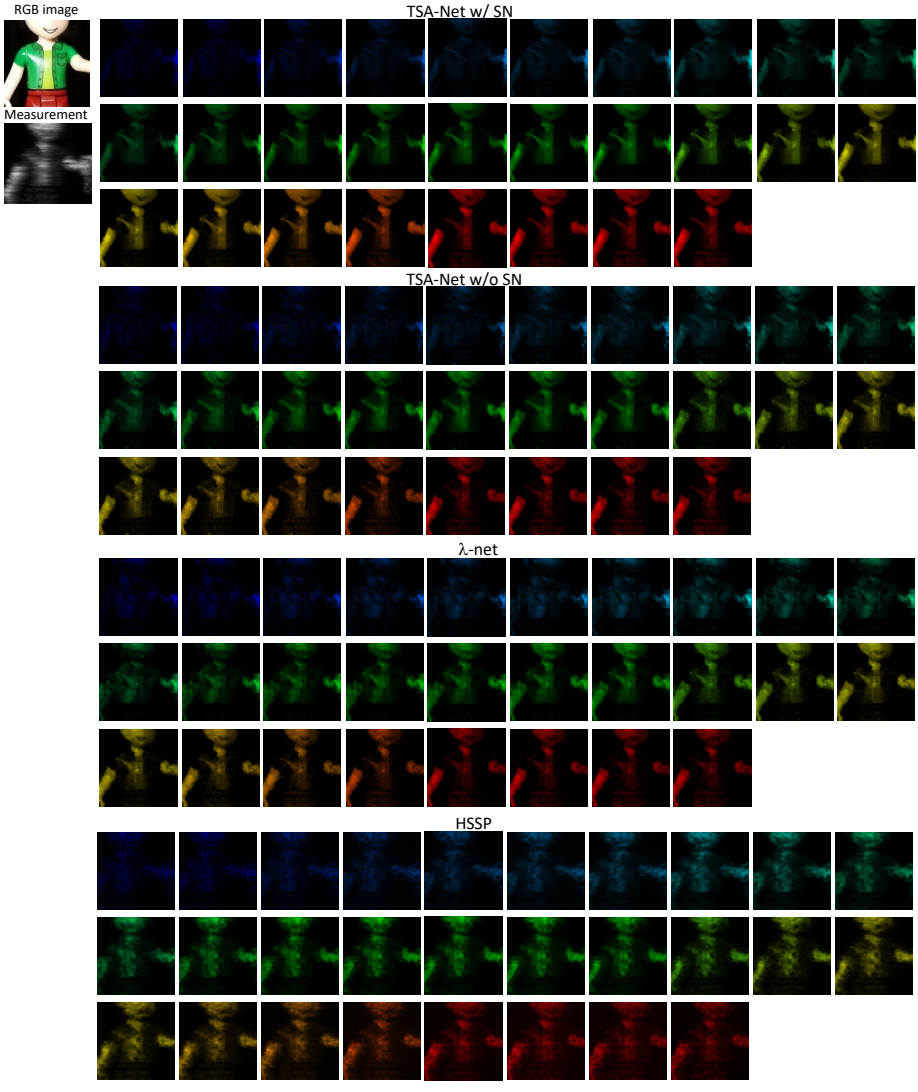


**Fig. M18.** Real data: RGB image, measurement and reconstructed results by TSA-Net trained with and without shot noise, as well as $\lambda$-net and HSSP.

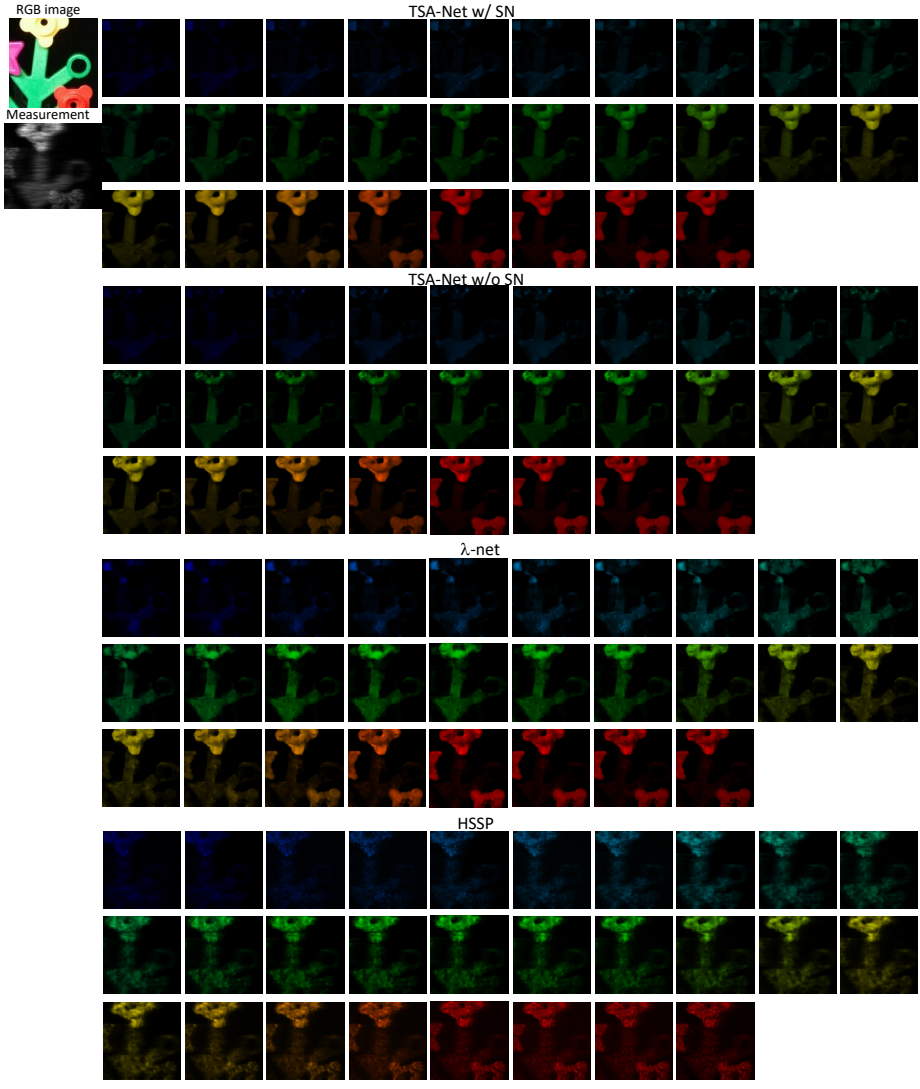


**Fig. M19.** Real data: RGB image, measurement and reconstructed results by TSA-Net trained with and without shot noise, as well as $\lambda$-net and HSSP.

## References

1. Bioucas-Dias, J., Figueiredo, M.: A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration. IEEE Transactions on Image Processing **16**(12), 2992–3004 (December 2007)
2. Choi, I., Jeon, D.S., Nam, G., Gutierrez, D., Kim, M.H.: High-quality hyperspectral reconstruction using a spectral prior. vol. 36, p. 218. ACM (2017)
3. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
4. Liu, Y., Yuan, X., Suo, J., Brady, D.J., Dai, Q.: Rank minimization for snapshot compressive imaging. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(12), 2990–3006 (Dec 2019)
5. Miao, X., Yuan, X., Pu, Y., Athitsos, V.: $\lambda$-net: Reconstruct hyperspectral images from a snapshot measurement. In: IEEE/CVF Conference on Computer Vision (ICCV) (2019)
6. Wang, L., Sun, C., Fu, Y., Kim, M.H., Huang, H.: Hyperspectral image reconstruction using a deep spatial-spectral prior. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
7. Yasuma, F., Mitsunaga, T., Iso, D., Nayar, S.K.: Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. vol. 19, pp. 2241–2253. IEEE (2010)
8. Yuan, X.: Generalized alternating projection based total variation minimization for compressive sensing. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 2539–2543 (Sept 2016)