POLITECNICO DI TORINO
Repository ISTITUZIONALE

NTIRE 2022 Spectral Recovery Challenge and Data Set

Original
NTIRE 2022 Spectral Recovery Challenge and Data Set / Boaz, A., Radu, T., Rony, Y., Nimrod, M., Amir, B., Yuanhao, C., Jing, L., Zudi, L., Haoqian, W., Yulun, Z., Agarla, M., Others,. - (2022), pp. 862-880. (2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW New Orleans, LA (USA) 19-20 June 2022) [10.1109/CVPRW56347.2022.00102].

Availability:
This version is available at: 11583/2971543 since: 2022-09-21T10:58:43Z

Publisher:
IEEE/CVF

Published
DOI:10.1109/CVPRW56347.2022.00102

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

27 November 2026

This CVPR workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

NTIRE 2022 Spectral Recovery Challenge and Data Set

Boaz Arad Radu Timofte Rony Yahel Nimrod Morag Amir Bernat
Yuanhao Cai Jing Lin Zudi Lin Haoqian Wang Yulun Zhang Hanspeter Pfister
Luc Van Gool Shuai Liu Yongqiang Li Chaoyu Feng Lei Lei Jiaojiao Li
Songcheng Du Chaoxiong Wu Yihong Leng Rui Song Mingwei Zhang
Chongxing Song Shuyi Zhao Zhiqiang Lang Wei Wei Lei Zhang Renwei Dian
Tianci Shan Anjing Guo Chengguo Feng Jinyang Liu Mirko Agarla
Simone Bianco Marco Buzzelli Luigi Celona Raimondo Schettini Jiang He
Yi Xiao Jiajun Xiao Qiangqiang Yuan Jie Li Liangpei Zhang Taesung Kwon
Dohoon Ryu Hyokyoung Bae Hao-Hsiang Yang Hua-En Chang Zhi-Kai Huang
Wei-Ting Chen Sy-Yen Kuo Junyu Chen Haiwei Li Song Liu Sabarinathan
K Uma B Sathya Bama S. Mohamed Mansoor Roomi

Abstract

This paper reviews the third biennial challenge on spectral reconstruction from RGB images, i.e.*, the recovery of whole-scene hyperspectral (HS) information from a 3-channel RGB image. This challenge presents the "ARAD_1K" data set: a new, larger-than-ever natural hyperspectral image data set containing 1,000 images. Challenge participants were required to recover hyperspectral information from synthetically generated JPEG-compressed RGB images simulating capture by a known calibrated camera, operating under partially known parameters, in a setting which includes acquisition noise. The challenge was attended by 241 teams, with 60 teams competing in the final testing phase, 12 of which provided detailed descriptions of their methodology which are included in this report. The performance of these submissions is reviewed and provided here as a gauge for the current state-of-the-art in spectral reconstruction from natural RGB images.*



Figure 1. Sample images from the ARAD_1K hyperspectral image data set. Note the variety of settings and viewpoints (images modified for optimal display).

1. Introduction

Hyperspectral imaging systems (HIS) are able to record the distribution of light in a scene across a large number of narrow spectral bands [12]. HISs can therefore provide more detailed visual information than conventional RGB cameras which are limited to three wide spectral bands (red, green, blue). While HISs can provide many benefits to a wide range of computer vision applications their size, cost, limited resolution, and often long image acquisition times have thus far limited their use to specialized industrial and scientific applications [4, 5].

To facilitate more widespread use of spectral information in computer vision application, researchers have continued to develop improved physical HISs [27] as well as software systems to recover spectral information from more readily available data sources such as RGB images. Early attempts at this task relied on sparse-coding/regression based methods [1, 3, 36, 39, 43]. In recent years neural

net based methodologies have become significantly more prominent [10, 23, 32] though not entirely displacing approaches such as sparse-coding [33]. The goal of this challenge is to gauge the state-of-the-art in spectral recovery from natural RGB images and provide a larger-than-every natural hyperspectral image data set to facilitate future development.

This challenge is one of the NTIRE 2022 associated challenges: spectral recovery [7], spectral demosaicing [6], perceptual image quality assessment [19], inpainting [40], night photography rendering [17], efficient super-resolution [31], learning the super-resolution space [34], super-resolution and quality enhancement of compressed video [49], high dynamic range [38], stereo super-resolution [45], burst super-resolution [8].

2. Data Set

To facilitate development and evaluation of state-of-the-art methodologies for recovering spectral information from natural RGB images, a larger-than-ever natural hyperspectral image data set is presented. This data set expands on the previously published ARAD HS data set [5] nearly doubling its size to 1,000 images. This data set is termed the "ARAD_1K Natural Hyperspectral Image Data Set" Figure 1 depicts a set of sample images from the ARAD_1K data set. The 1,000 images included in the data set were divided as follows: 900 training images, 50 validation images, and 50 test images. Training and validation images were fully released to participants during the challenge, while ground truth hyperspectral information for the 50 test images remains confidential to facilitate equal grounds evaluation of future works.

The ARAD_1K data set was collected with a Specim IQ mobile hyperspectral camera. The Specim IQ camera is a stand-alone, battery-powered, compact, push-broom spectral imaging system which can operate independently without the need for an external power source or computer controller. As in the previously released ARAD data set, the use of a highly mobile spectral imaging system facilitated collection of an extremely diverse data set with a large variety of scenes and subjects.

The Specim IQ camera provides RAW 512×512px images with 204 spectral bands in the 400-1000nm range. For the purpose of this challenge, manufacturer-supplied radiometric calibration was applied to the RAW images, and the images were resampled to 31 spectral bands in the visual range (400-700nm). Radiometric calibration corrects for measurement biases introduced by the camera system's CMOS sensor, converting the recorded RAW per channel intensity data into accurate spectral measurements. "Lines" (image columns) with excessive interference are also removed by this process, resulting in a 482×512px image, resampled to 31 bands from 400nm to 700nm with a 10nm step. Images previously included in the ARAD data set have been updated to the most recent radiometric calibration standard.

This data set is further expanded for the NTIRE 2022 Spectral Demosaic Challenge [6], where 16 channel spectral images are provided over a 400-1000nm range - covering a wider range of wavelengths at a reduced spectral resolution.

Additional information regarding the data set, its relation to the previously published ARAD data set, instructions for data access, and relevant code is available at the following GitHub repository: https://github.com/boazarad/ARAD_1K

2.1. Camera Simulation

The NTIRE 2020 [5] and NTIRE 2018 [4] spectral recovery challenges included two tracks: a "Real-World" track which attempted to simulate recovering spectral information from physical cameras, and a "clean" track which required recovery of spectral information from a noiseless projection to RGB. Due to increasingly low error rates in the latter task, as well as its low practical feasibility in real-world applications, this challenge includes a single track which aims to predict the performance of proposed methods in a feasible real-world setting.

The challenge aims to simulate a setting where the source camera is *known* but not fully controllable hence the following assumptions are made:

1. The camera's spectral response function is known.
2. The camera determines its exposure settings automatically - the exposure algorithm is known, but parameters used to compute it for each scene are not (*e.g.* average scene brightness).
3. The camera implements a realistic noise model.
4. Rudimentary image signal processing (ISP) is applied in-camera (highlight clipping).
5. Images are saved in compressed JPEG format.

Participants were provided with training images produced by the challenge camera simulation pipeline, camera simulation pipeline code, and the camera response function used in the simulation. Figure 2 depicts the camera response function used for camera simulation in this challenge. While pipeline code and camera response function were provided to participants, the exact noise parameters and JPEG compression level used to generate RGB images for the challenge was kept confidential.

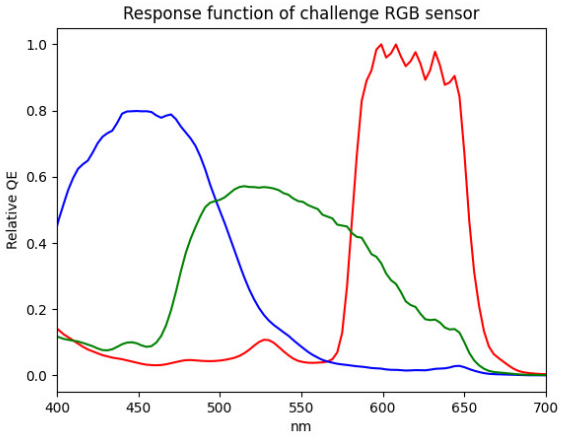


Figure 2. Response function of a challenge RGB camera sensor based on physical measurements of a Basler ace 2 camera (model A2a5320-23ucBAS).

3. Challenge

The NTIRE 2022 Spectral Recovery Challenge was presented as a competition on the CodaLab [1] platform which consisted of two phases:

1. **Development** participants were provided with 900 training and 50 validation RGB images generated by the camera simulation pipeline (c.f. Sec. 2.1). Corresponding ground truth hyperspectral images were provided for the 900 training images. A test server was made available where participants could upload recovered spectral information for the 50 validation images and receive immediate feedback on their performance in terms of MRAE and RMSE per-image (c.f. Sec. 3.1). During the development phase, there were no limits on the amount of submissions per team.

2. **Testing** Ground truth hyperspectral images for the 50 validation images were released, alongside 50 test RGB images. Similarly to the development phase, a test server was made available where participants could upload their results and receive feedback on their performance, but each team was limited to a total of three submissions. This feedback allowed participants to select their best model, while limiting the possibility of overfitting to the test set.

Code and other data provided to participants is curated in the following GitHub repository:https://github.com/boazarad/NTIRE2022_spectral

[1] https://codalab.lisn.upsaclay.fr/competitions/721

3.1. Evaluation Metrics

As in previous competitions [4, 5], Mean Relative Absolute Error (MRAE) computed between the submitted reconstruction results and the ground truth images was selected as the quantitative measure for the competition. Root Mean Square Error (RMSE) was reported as well, but not used to rank results. MRAE and RMSE are defined as follows:

$$MRAE = \frac{\sum_{i,c} \frac{|P_{gt_{i_c}} - P_{rec_{i_c}}|}{P_{gt_{i_c}}}}{|P_{gt}|} \tag{1}$$

$$RMSE = \sqrt{\frac{\sum_{i,c} \left(P_{gt_{i_c}} - P_{rec_{i_c}}\right)^2}{|P_{gt}|}} \tag{2}$$

where $P_{gt_{i_c}}$ and $P_{rec_{i_c}}$ denote the value of the c spectral channel of the i-th pixel in the ground truth and the reconstructed image, respectively, and $|P_{gt}|$ is the size of the ground truth image (pixel count $\times$ number of spectral channels).

3.2. Evaluation Protocol

A significant challenge in evaluating the performance of spectral recovery methodologies is curating a test set which is representative of the desired target domains. While the ARAD_1K data set provides a larger-than-ever 50 image test set, including a large variety of images from multiple settings (c.f. Sec. 2), limitations of the CodaLab platform prevented large scale evaluation over the full test set due to space and bandwidth constraints. To overcome this limitation without significantly reducing the representation power of the test set, test images were cropped from their original 482×512 spatial resolution to a central 226×256 region. Participants were provided with code to prepare images for evaluation and their results were scored for MRAE and RMSE over the selected central region of the test images.

4. Challenge Results

Table 1 details the final rankings of all participants over the primary evaluation metrics. The lowest MRAE achieved was 0.1131 and the lowest RMSE achieved was 0.02308. The top-7 ranked results would remain consistent had RMSE been the primary metric.

Despite the use of significantly advanced methodologies and vastly improved hardware, the top performing method in this challenge achieved a MRAE score nearly twice as high as the top performing solution in the comparable "Real World" track of the NTIRE 2020 challenge [5] (0.1131 vs. 0.06200). This is likely both due to challenges posed by a more realistic camera model (c.f. Sec. 2.1) as well as the increased size of the test set (50 images vs. 10 images).

Rank	Team	Username	MRAE	RMSE
1	MST++ [10]	THU-SIGS-MEAI	0.1131	0.02308
2	MIALGO	mialgo_ls	0.1247	0.02569
3	CVIA_SSR [28]	deeppf	0.1766	0.03217
4	IFL	Ptdoge	0.2035	0.03237
5	SSR	songyonger	0.2586	0.03876
6	Yuelushan	anjing_guo	0.2802	0.04161
7	IVL [2]	IVLLuigiCelona	0.2915	0.05412
8	SGG_RS_Whu	hj_whu	0.3060	0.05071
9	star-spectral	star.kwon	0.4127	0.04898
10	NTU607QCO-Spectral	Alex_Huang	0.4361	0.07689
11	OPT KLSIT	xiongmao	0.5304	0.11310
12	Image Lab	SabariNathan	0.7795	0.10161

Table 1. NTIRE 2022 Spectral Reconstruction Challenge results and final rankings on the ARAD_1K HS test data.

Team Name	CPU	GPU	Platform	Train Time	Inference Time
MST++	Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz	NVIDIA RTX 3090	PyTorch	40 hours	0.10s
MIALGO	Intel(R) Xeon(R) Gold 6240 CPU @ 2.60GHz * 2	8 x NVIDIA Tesla V100 32GB	PyTorch	5 Days	0.43s
CVIA_SSR	Intel i9-10900K CPU @ 3.70GHz	NVIDIA RTX 3090 GPU	PyTorch	24 Hours	0.15s
IFL	Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz	NVIDIA RTX3090 GPU	PyTorch	3 Days	0.12s
SSR	Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz	NVIDIA 1080Ti	PyTorch	60 Hours	1.7s
Yuelushan	Intel i9-11900K	2 x NVIDIA RTX 1080Ti + NVIDIA RTX 3090 GPU	PyTorch, TensorFlow	26 Hours	1.03s
IVL	Intel i7-4770 CPU @3.40GHz	NVIDIA GeForce GTX 1080	PyTorch, MATLAB	30 Minutes	**0.01s**
SGG RS Whu		NVIDIA RTX A5000	PyTorch	6 Hours	0.28s
star-spectral	Intel i7-8700 CPU @3.20GHz	NVIDIA RTX 2080Ti	PyTorch	5 Hours	0.09s
NTU607QCO-Spectral		NVIDIA V100	PyTorch	9 Days	2.8s
OPT-KLSIT	Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz	NVIDIA GeForce RTX 3090	PyTorch	7 Days	0.67s
Image Lab	IntelCore i7 processor	NVIDIA RTX 2060	Keras,Tensorflow	20 Days	1.0s

Table 2. Self-reported training and inference runtimes for proposed methods.

Section 6 describes the methodologies used by top-performing teams in this challenge, as described by their authors.

4.1. Performance on "Out-of-Scope" Image

As in the previous competition [5], finalists were presented with an "out-of-scope" image to recover. Figure 3 depicts the out-of-scope image selected for this challenge: it features a prominent human subjects and calibration target - objects which are very rare in the training data set. Furthermore, the image was taken under photographic studio lights, while the majority of training images were captured under natural illumination or conventional indoor lighting. While performance on the out-of-scope image may be indicative of a methods' extrapolation power, these measurements did not affect participants final ranking in the challenge.

Table 3 details the performance of most submitted methods over the out-of-scope image. Figure 4 depicts recovered spectra sampled from the red, green, and blue tiles of the calibration target. While some methodologies exhibited improved average performance over the out-of-scope image, none were able to accurately recover spectra from the calibration target. Out-of-scope performance was not correlated strongly with primary test set performance. This variability in performance indicates that, despite being the largest of its kind, the ARAD_1K data set is not yet comprehensive. Further collection of natural hyperspectral images would be conductive to the development of improved spectral recovery systems and their evaluation.

Rank	Team	MRAE	RMSE
1	Yuelushan$_{(6)}$	0.1646	0.05480
2	CVIA_SSR [28]$_{(3)}$	0.1786	0.05501
3	SSR$_{(5)}$	0.1915	0.04387
4	SGG_RS_WHU$_{(8)}$	0.2197	0.05531
5	MST++ [10]$_{(1)}$	0.2206	0.05196
6	IFL$_{(4)}$	0.2365	0.03702
7	MIALGO$_{(2)}$	0.2503	0.03437
8	star-spectral$_{(9)}$	0.3439	0.04812
9	NTU607QCO-Spectral$_{(10)}$	0.3610	0.05319
10	OPT KLSIT$_{(11)}$	0.4299	0.07634
11	IVL [2]$_{(7)}$	0.5350	0.05879

Table 3. Performance of proposed methodologies for "out-of-scope" image, ranking on the primary test set is denoted in subscript beside the team name.



Figure 3. "Out-of-scope" image used to gauge the extrapolation ability of methods presented in this challenge. This image contains a prominent human subject, a calibration target, and was taken under studio lighting (image modified for optimal display).

5. Conclusion

The NTIRE 2022 Spectral Recovery Challenge continues the biennial tradition of providing the most extensive evaluation of methods for spectral recovery from RGB images. The challenge provided a larger-than-ever natural hyperspectral data set for both training and evaluation. Participation in this year's challenge was the highest to date, with participation numbers increased by over 130% relative to the 2020 challenge.

While neural networks remain the primary tool used by top-performing methodologies, this year's top-performing methodologies were able to present inference times below the 0.5 second processing time required by top-performers in the 2020 challenge [5].The hybrid methodology presented by Hu *et al*. [22] is of particular note, providing sub-30ms inference times, albeit at the cost of higher error rates.

Between the higher overall MRAE/RMSE scores of this years top performers relative to previous challenges and variability observed in the "out-of-scope" test - it is clear that there is both significant potential for improved performance over the currently available data set as well as room to expand the coverage of future natural hyperspectral image data sets to additional scene types and domains. It is our hope that the data and methodologies described here will facilitate both these goals.

6. Methods and Teams

6.1. MST++: Multi-stage Spectral-wise Transformer for Efficient Spectral Reconstruction [10]

Figure 5, describes the MST++ pipeline: (a) depicts the proposed Multi-stage Spectral-wise Transformer (MST++), which is cascaded by N_s Single-stage Spectral-wise Transformers (SSTs). MST++ takes a RGB image as input and reconstructs its HSI counterpart. A long identity mapping is exploited to ease the training procedure. Fig. 5 (b) shows the U-shaped SST consisting of an encoder, a bottleneck, and a decoder. The embedding and mapping block are single $conv3\times3$ layers. The feature maps in the encoder sequentially undergo a downsampling operation (a strided $conv4\times4$ layer), N_1 Spectral-wise Attention Blocks (SABs), a downsampling operation, and N_2 SABs. The bottleneck is composed of N_3 SABs. The decoder employs a symmetrical architecture. The upsampling operation is a strided $deconv2\times2$ layer. To avoid the information loss in the downsampling, skip connections are used between the encoder and decoder. Fig. 5 (c) illustrates the components of SAB, *i.e.*, a Feed Forward Network (FFN as shown in Fig. 5 (d)), a Spectral-wise Multi-head Self-Attention (S-MSA), and two layer normalization. Details of S-MSA are given in Fig. 5 (e).

S-MSA. Suppose $\mathbf{X}_{in} \in \mathbb{R}^{H\times W\times C}$ as the input of S-MSA, which is reshaped into tokens $\mathbf{X} \in \mathbb{R}^{HW\times C}$. Then $\mathbf{X}$ is linearly projected into *query* $\mathbf{Q} \in \mathbb{R}^{HW\times C}$, *key* $\mathbf{K} \in \mathbb{R}^{HW\times C}$, and *value* $\mathbf{V} \in \mathbb{R}^{HW\times C}$:

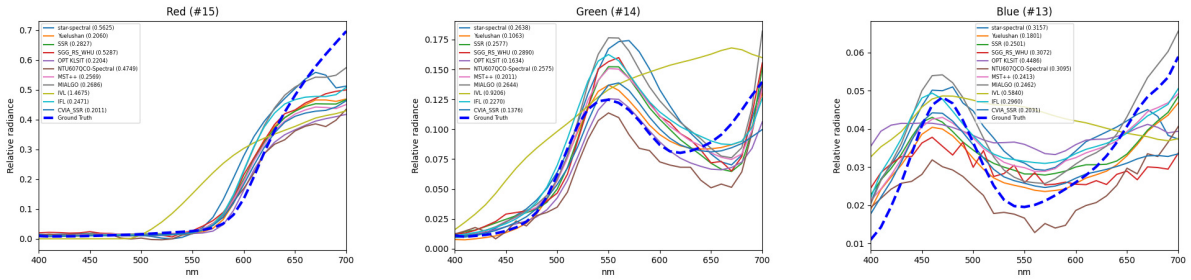
$$\mathbf{Q} = \mathbf{X}\mathbf{W}^{\mathbf{Q}}, \mathbf{K} = \mathbf{X}\mathbf{W}^{\mathbf{K}}, \mathbf{V} = \mathbf{X}\mathbf{W}^{\mathbf{V}}, \tag{3}$$

where $\mathbf{W}^{\mathbf{Q}}$, $\mathbf{W}^{\mathbf{K}}$, and $\mathbf{W}^{\mathbf{V}} \in \mathbb{R}^{C\times C}$ are learnable parameters; $biases$ are omitted for simplification. Subsequently, we respectively split $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ into N *heads* along the spectral channel dimension: $\mathbf{Q} = [\mathbf{Q}_1, \ldots, \mathbf{Q}_N]$, $\mathbf{K} = [\mathbf{K}_1, \ldots, \mathbf{K}_N]$, and $\mathbf{V} = [\mathbf{V}_1, \ldots, \mathbf{V}_N]$. The dimension of each head is $d_h = \frac{C}{N}$. Please note that Fig. 5 (e) depicts the situation with N = 1 and some details are omitted for simplification. Different from original MSAs, our S-MSA treats each spectral representation as a token and calculates self-attention for $head_j$:

$$\mathbf{A}_j = \text{softmax}(\sigma_j \mathbf{K}_j^{\mathrm{T}} \mathbf{Q}_j), \quad head_j = \mathbf{V}_j \mathbf{A}_j, \tag{4}$$

where $\mathbf{K}_j^{\mathrm{T}}$ denotes the transposed matrix of $\mathbf{K}_j$. Because the spectral density varies significantly with respect to the wavelengths, we use a learnable parameter $\sigma_j \in \mathbb{R}^1$ to adapt the self-attention $\mathbf{A}_j$ by re-weighting the matrix multiplication $\mathbf{K}_j^{\mathrm{T}}\mathbf{Q}_j$ inside $head_j$. Subsequently, the outputs of N *heads* are concatenated to undergo a linear projection and then is added with a position embedding:

$$\text{S-MSA}(\mathbf{X}) = \big(\underset{j=1}{\overset{N}{\text{Concat}}}(head_j)\big)\mathbf{W} + f_p(\mathbf{V}), \tag{5}$$



(a) (b) (c)

Figure 4. Recovered radiance spectra from the Red(a), Green(b), and Blue(c) tiles of the color calibration target in the out-of-scope image (Figure 3). Per-spectra MRAE values for each method are included in parenthesis on the plot legends.

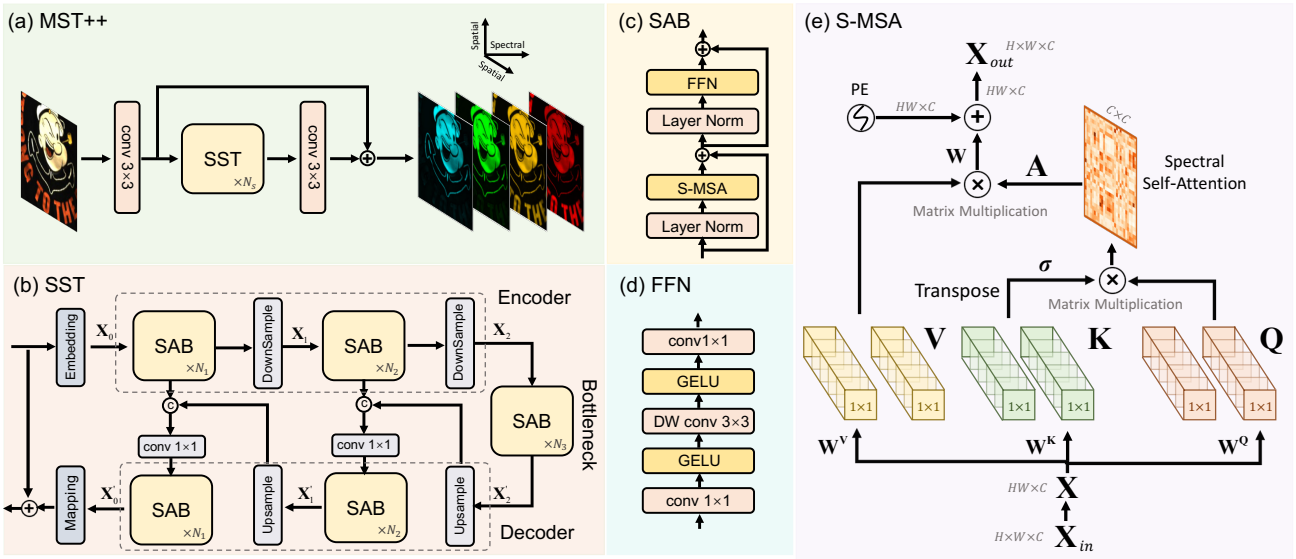


Figure 5. The overall pipeline of MST++. (a) Multi-stage Spectral-wise Transformer. (b) Single-stage Spectral-wise Transformer. (c) Spectral-wise Attention Block. (d) Feed Forward Network. (e) Spectral-wise Multi-head Self-Attention.

where $\mathbf{W} \in \mathbb{R}^{C\times C}$ are learnable parameters, $f_p(\cdot)$ is the function to generate position embedding. It consists of two depth-wise *conv*3×3 layers, a GELU activation, and reshape operations. The HSIs are sorted by the wavelength along the spectral dimension. Therefore, we exploit this embedding to encode the position information of different spectral channels. Finally, we reshape the result of Eq. (5) to obtain the output feature maps $\mathbf{X}_{out} \in \mathbb{R}^{H\times W\times C}$.

- **Total method complexity**

  Our MST++ requires 1.62 M Params and 23.05 G FLOPS.
  The test size is 256×256×3.

- **Training**

  During the training procedure, RGB images are linearly rescaled to [0, 1], after which 128×128 RGB and HSI sample pairs are cropped from the data set. The batch size is set to 20 and the parameter optimization algorithm chooses Adam modification with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is initialized as 0.0004 and the Cosine Annealing scheme is adopted for 300 epochs. The trianing data is augmented with random rotation and flipping. The proposed MST++ has been implemented on the Pytorch framework and approximately 40 hours are required for training a network on a single RTX 3090 GPU. MRAE loss function between the predicted and ground-truth HSI is adopted as the objective. In the implementation of our MST++, we set $N_s = 3$, $N_1 = N_2 = N_3 = 1$, $C = 31$.

- **Testing**

  During the testing phase, the entire RGB image is also linearly rescaled to [0, 1] and fed into the network to

fulfill the spectral recovery. Our MST++ takes 102.48 ms for per image (size 482×512×3) reconstruction on a single RTX 3090 GPU.

- **Ensembles and fusion strategies**

  We adopt three ensemble strategies, including:

  (a) self-ensemble [44], the RGB input is flipped up/down/left/right or rotated 90°/180°/270° to be fed into the network, the output HSIs are then averaged.

  (b) multi-model ensemble, we also train MIRNet [52], MPRNet [53], Restormer [51], HiNet [13], MST [9] families. The reconstructed HSIs of these models and our MST++ are linearly fused together.

  (c) multi-scale ensemble, we respectively train our models with patches at size of 256×256, 128×128, and 64×64. Then fuse the output HSIs.

  On the validation set, self-ensemble, multi-model ensemble, and multi-scale ensemble can obtain about 0.015, 0.045, 0.033 in terms of MRAE, respectively.

- **Code and Pre-trained Models**

  We contribute a baseline and toolbox containing 11 SOTA image restoration methods and their pre-trained models to beneift the comunity of spectral reconstruction. The repository is at https://github.com/caiyuanhao1998/MST-plus-plus

6.2. MIALGO: Enhanced Holistic Attention Network for Spectral Reconstruction

Spectral reconstruction, as a typical reconstruction task is highly similar to the image super-resolution. We utilize Holistic Attention Network(HAN [35]), a SOTA method in the super-resolution tasks as the backbone to solve it. To be specific, we notice that the brightness(mean) of the input images is set to a fixed value(typical scene reflectivity, 0.18), it is particularly important to estimate the brightness of the target, and thus we divide the RGB/Mosaic image into two cases based on the maximum value. Followings are detailed explanations for the two cases:
1. The maximum value is **less than** the upper limit (255 for rgb or 4095 for mosaic). In this case, the maximum value of the input corresponds to the maximum value of GT (ignoring the effects of mosaic processing and quantization errors), so we add a simple normalization layer before the backbone, after that, the brightness of the image is basically same with GT. This case is relatively simple, and the network can handle it well.
2. The maximum value of the input is **equal** the upper limit. In this case, the clip operation during the generation of input causes a lot of energy loss, so the brightness cannot be estimated by referring to the maximum value like case 1. To deal with this ill-conditioned and difficult problem, we use a lot of augmented data for training.
We also remove the upsampling layer of HAN to keep the size of the input, and add a normalization layer after the backbone to avoid the loss caused by the clip operation.



Figure 6. Architecture of the Enhanced Holistic Attention Network for Spectral Reconstruction.

Figure 6 describes the high-level architecture of the solution.

- **Total Method Complexity**

  the total number of GMACS is 1822, and the total number of parameters is 7457168.

- **Additional Training Data**

  We found that the bottleneck of the task is the brightness estimation, that is, the richness of the data, so we tried to use a lot of additional data, including ICLV [3], CAVE [50] and Harvard [11].

- **Training**

  According to the code provided by the organizer, we generate and augment the input data ourselves, including random brightness, random noise, random padding, flip, rotation, etc. We first train on all the data for 100k iterations, and then train separately on each case's data for 100k iterations. In the later stages of training, we increase the proportion of hard samples. MRAE and SSIM were used as training loss and in late training, we keep only the luminance component of SSIM.

- **Testing**

  The model is switched according to the maximum value of the input, which corresponds to the two cases in the training phase.

6.3. CVIA_SSR: DRCR Net: Dense Residual Channel Re-calibration Network with Non-local Purification for Spectral Super Resolution [28]

In this section, we describe our proposed dense residual channel re-calibration network (DRCR Net) in detail. Given $\mathbf{I}_{RGB}$ as the input of DRCR Net. As illustrated in Fig. 8, we first employ two convolutional layers to extract the shallow feature $\mathbf{F}_0$ as well as boost the number of bands from input RGB images.

$$\mathbf{F}_{SF} = H_{SFE_convs}(\mathbf{I}_{RGB}), \tag{6}$$

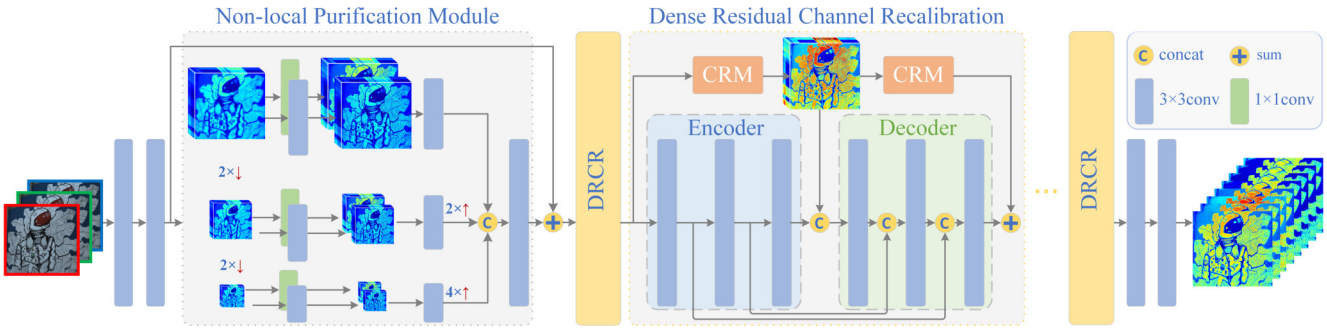


Figure 7. Architecture of dense residual channel re-calibration network for Spectral Super Resolution from RGB Images. ↓ and ↑ indicate downsampling and upsampling, respectively.

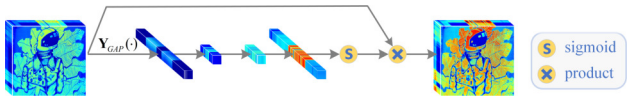


Figure 8. Architecture of channel re-calibration module. In the figure, $\mathbf{Y}_{GAP}(\cdot)$ represents the global average pooling operation.

where $H_{SFE_convs}(\cdot)$ stands for front convolution operations. Then we use the extracted shallow feature $\mathbf{F}_{SF}$ as the input of the non-local purification module (NPM). Thus we can further have

$$\mathbf{F}^0 = H_{DF}(\mathbf{F}_{SF}), \tag{7}$$

where $H_{DF}(\cdot)$ represents our designed very simple but efficient NPM whose output $\mathbf{F}^0$ is then taken as the input of our multiple dense residual channel re-calibration (DRCR) blocks.

$$\begin{aligned}\mathbf{F}^m &= H^m_{DRCR}(\mathbf{F}^{m-1}) \\ &= H^m_{DRCR}(H^{m-1}_{DRCR}(\cdots H^1_{DRCR}(\mathbf{F}^0)\cdots)),\end{aligned} \tag{8}$$

where $\mathbf{F}^m$ and $\mathbf{F}^{m-1}$ denote the output and the input of the mth DRCR block, separately. $H^m_{DRCR}(\cdot)$ represents the mth DRCR block. To be specific, the DRCR blocks have a U-shaped structure in which the encoding part and decoding part consist of three 3×3 plain convolution layers, separately. Additionally, three concatenation operations between the encoding part and decoding part are utilized to explore the information interaction among the intermediate layers and such skip cross-layer connections help to alleviate the vanishing gradient problem. Besides, we employ the dual channel re-calibration module (CRM) to re-calibrate the features associated with the DRCR block along the channel dimension, where the first CRM $H^{(m,1)}_{DCRM}(\cdot)$ draws the calibration feature $\mathbf{F}^{m,1}_{RF}$ from the input of the mth DRCR block. The above process can be expressed as

$$\mathbf{F}^{(m,1)}_{RF} = H^{(m,1)}_{DCRM}(\mathbf{F}^{m-1}) \tag{9}$$

We then fuse the $\mathbf{F}^{(m,1)}_{RF}$ with the aggregated features in the middle layer of mth DRCR block, additionally, we also input the $\mathbf{F}^{(m,1)}_{RF}$ into the second DCRM $H^{(m,2)}_{DCRM}(\cdot)$ for further calibration of the channel-dimensional features and the above process can be formulated as

$$\mathbf{F}^{(m,2)}_{RF} = H^{(m,2)}_{DCRM}(\mathbf{F}^{(m,1)}_{RF}), \tag{10}$$

Therefor, the output of the ith convolution layer in mth DRCR block can be expressed as:

$$\mathbf{F}^{(m,i)} = \begin{cases} H^{(m,i)}_{DRCR_conv}(\mathbf{F}^{m-1}) & i = 1 \\ H^{(m,i)}_{DRCR_conv}(\mathbf{F}^{(m,i-1)}) & i = 2,3 \\ H^{(m,i)}_{DRCR_conv}([\mathbf{F}^{(m,1)}_{RF}, \mathbf{F}^{(m,i-1)}]) & i = 4 \\ H^{(m,i)}_{DRCR_conv}([\mathbf{F}^{m,i-1}, \mathbf{F}_{(m,7-i)}]) & i = 5,6, \end{cases} \tag{11}$$

where $H^{(m,i)}_{DRCR_conv}(\cdot)$ denotes the ith convolution operation of mth DRCR block, and $[\cdot,\cdot]$ represents the concatenation operation of two features. Moreover, we add $\mathbf{F}^{(m,2)}_{RF}$ to the output of the last convolution layer $\mathbf{F}^(m,6)$, thus we can further have

$$\mathbf{F}^m = \mathbf{F}^{m,i-1} + \mathbf{F}^{(m,2)}_{RF}. \tag{12}$$

Finally, similar to the front structure of the network, we use two plain convolutions to aggregate features and map the number of bands to 31 to obtain the spectral reconstructed HSI $\mathbf{I}_{SR}$.

$$\mathbf{I}_{SR} = H_{AF_convs}(\mathbf{F}^m), \tag{13}$$

where $H_{AF_convs}(\cdot)$ stands for tail convolution operations.

- **Training**

  For training details, we set the number of DRCR blocks to 10, and the channels of intermediate layer features to 100. The image pairs are cropped to 128×128 region before normalized to $[0, 1]$. The reduction ratio r value of the channel re-calibration module (CRM) is 8. For optimization, we choose Adam

with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 10^{-8}$. The learning rate is set as 0.0001 initially and a decay policy with a power of 1.5. We stop network training at 100 epoch. Our DRCR Net has been implemented on the Pytorch framework and approximately 24 hours are required for training the NTIRE2022 data set on 1 NVIDIA 3090Ti GPU.

- **Testing**

  We choose to input the complete RGB images to the network to fulfill the spectral recovery on an NVIDIA 3090Ti GPU with 24G memory. Our network takes 0.158s per image (GPU time) for test data.

6.4. IFL: Residual Dual Attention Network (RDAN)

In this challenge, we propose a residual dual attention network (RDAN) for spectral reconstruction from RGB images. Residual dual attention block (RDAB) is the basic unit of RDAN, which includes two key components, the dual attention module (DAM) and group progressive convolution module (GPCM). Note that the structure of RDAB benefits from [29]. DAM is developed to capture spatial long-range similarity and channel short-range dependency within intermediate features, while GPCM is designed to explore local contextual consistency. Fig. 9 shows the detail architecture of RDAN. Concretely, DAM includes two parts, non-local spatial attention module (NLSAM) and local spectral attention module (LSAM). As shown in Fig. 10a, NLSAM and LSAM are jointed parallel. NLSAM adopts classic non-local operations to explore spatial long-range similarity. To reduce computational burden, different from [47], NLSAM regards one patch as one pixel, which making it possible to embed non-local operations in each basic unit of network in the case of limited memory resources. Inspired by recent work [46] and considering high correlation of adjacent spectral bands in HSIs, LSAM is introduced to model channel short-range dependency. Derived from [18], GPCM is proposed. From Fig. 10b, the number of feature channels and size of receptive field are progressively increased and enlarged in GPCM, which is helpful to explore local contextual consistency effectively. Besides, owing to group and concatenation operation, GPCM improves diversity and richness of representation apparently through implicit reuse of features and indirect multi-scale fusion.

In addition to the above well-designed network, we explore the data processing strategy seriously. Maybe Norm-factor provided is always ignored. Instead, the data used to supervised the output of our network is not normalized. Experiments prove that the strategy is effective in this challenge. Besides, mean relative absolute error (MARE) loss function [42] is used to train our model.

- **Training** During the development phase, we split the provided NTIRE training data into two parts for training and validation. We retain fours models trained with different parameter settings or training data, which have the best performance on corresponding validation data respectively. We evaluate the models retained again when validation data is public. Table 4 shows the performance, parameters setting, and training strategy of all models retained. In addition, during the development phase, the batch size of our model is 32, and the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ is adopted. 64×64 RGB-HSI pairs are cropped with a stride of 32 from the original data set for training [29]. The learning rate is initialized to 0.0001 and the linear function is set as the decay strategy. Then PyTorch framework is used to realize proposed models. The optimization of models is implemented on NVIDIA GPU.

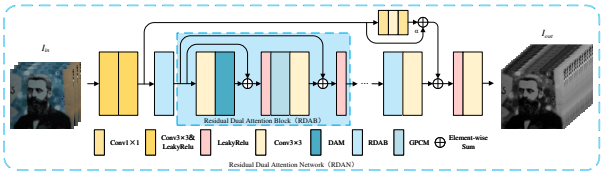
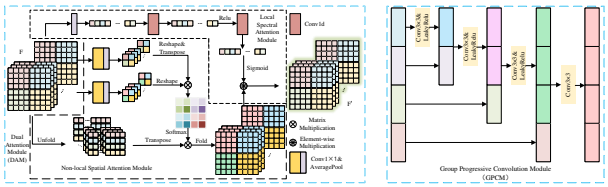


Figure 9. Overall Architecture of RDAN.



(a) Diagram of DAM (b) Diagram of GPCM

Figure 10. Proposed Attention and Convolution Module.

- **Testing**

  During the testing phase, the whole RGB image is input into our trained model. Output images were normalized (divided by maximum value) before submission.

- **Ensembles and fusion strategies**

  Four best models trained with different parameters setting are used to reconstruct spectral from given testing RGB images. Then all results are averaged as the final result. Compared with the single model, as shown in Tab. 4, model-ensemble strategy can further improve the accuracy of spectral recovery.

6.5. SSR: Improved Attention Network for Spectral Reconstruction

We improve the backbone of a single image spatial super-resolution model named HAN [35]. We just replace

Model Name	Patch Size in NLSAM	Total Iterations	MRAE on Cropped Image
Best_v1k4	4×4	278150	0.181807
Best_v1k8	8×8	278150	0.188734
Best_v2k4	4×4	277850	0.208914
Best_v2k8	8×8	277850	0.210183
Ensemble	-	-	0.177332

Table 4. Performance of individual models and model ensemble over the challenge's validation data.

the SE [22]-like channel attention in RCAB [54] unit with ECA [46]-like channel attention and added spatial attention after it. In this way, our model can focus on adjacent bands without the influence of distant irrelevant bands, and at the same time spatial attention emphasizes more important parts spatially. We first transform the RGB image with 3 channels into 31 channels through Linear interpolation, and then extract shallow features through a convolutional layer. After the improved backbone network, the network tail is a convolutional layer with the number of output channels of 31. During this process, the spatial size of feature map remains unchanged. In addition, we add global shortcuts to enable the backbone network to learn residuals. We set the number of groups in the HAN to 6, stack 12 RCAB blocks in each group, and set the number of channels in the middle layer to 64.

Besides, considering that the self-attention in Transformer can capture long-range dependency, we also try use Restormer [51] to complete the Spectral Reconstruction. We halved the size of the original Restormer and transform into a spectral super-resolution model as described above.

- **Training** The RGB-HSI pair is split into 64*64'size patch without overlapping. We adopt the min-max normalization for each input image independently. The initial learning rate is set as 1e-4 and decays by half every 30,000 iterations for 3 times. The batch size is 32 when training improved HAN and the optimizer is Adam with default hyper-parameters. When training Restormer, the batch size is 8.

- **Testing** In test phase, the whole image serves as network's input.

- Ensembles and Fusion Strategies We ensemble 4 models of improved HAN with different learning rates and batch sizes and one model of Restormer using the weighted average method.

6.6. Yuelushan: Stepwise spectral super-resolution

We think that the competition consists of three sub-tasks, and they are the image denoising (DN) task, the image de-white-balancing (DWB) task, and the final image spectral super-resolution (ISR) task. We firstly train an image denoising network for removing the noises and compressing artifacts. Then an image de-white-balancing network is proposed for restoring the original illumination intensity in spectral imaging. Finally, the de-white-balanced images are used for spectral reconstruction. all the three parts are implemented with the RCAN-like [54] deep models. The flow of the proposed framework is shown in Figure 11.



Figure 11. The flow of the proposed spectral reconstruction method.

- **Total Method Complexity**

  In the image denoising part, we utilize the original HSI images and official code for generating the noise-free RGB images, this step is called DN. In the second part, we use the image denoising model to generate the denoised images, which are then utilized for de-white-balancing, and the reference ground truth is generated by the official code. Finally, we restore the hyperspectral information from the DWBed images, we call it the ISR stage.

- **Training** In the DN and DWB parts, we utilize 5 RCABs for constructing the deep models, We train the networks for 100 epochs with batch size 4 and an initial learning rate of 0.0005, which reduces by a factor of 0.5 every 20 epochs. We choose the L_1 loss and Adam optimizer. All the experiments in these two stages are implemented by the TensorFlow framework in the Ubuntu16.04 environment with 128G RAM and 2 NVIDIA RTX 1080Ti GPUs.

  In the ISR part, we utilize 10 RCABs for constructing the deep model, We train the networks for 100 epochs with batch size 4 and an initial learning rate of 0.0005, which reduces by a factor of 0.5 every 20 epochs. We choose the MRAE loss and Adam optimizer. All the experiments in these two stages are implemented by the TensorFlow and PyTorch frameworks in the Ubuntu16.04 environment with 128G RAM and 2 NVIDIA RTX 1080Ti GPUs.

- **Testing** In the test phase, the images are sent to the trained DN, DWB, and ISR models, respectively, and then we can obtain the restored HSIs. We conduct the whole framework without model-ensemble.

6.7. IVL: Fast-n-Squeeze [2]

- **General method description:** Our method applies a 3×31 linear transformation matrix to convert RGB data into 31-band spectral data. For each image in the training set, such matrix is individually optimized using the Moore-Penrose pseudoinverse [37]. The training matrices are then used to define a new matrix to be used at inference time for each image, based on different rationales. We describe four variants based on this idea:
  - Fast: the images from the training set that are most similar to the test image are identified (based on low-level RGB statistics). The corresponding matrices are extracted, and a median matrix is computed for application to the test image.
    Test MRAE = 0.4629
  - Squeeze: a SqueezeNet [24] CNN model is trained and applied to estimate, given the input RGB image, a single global scaling factor to be applied to the reconstructed spectral image from Fast. In particular a weighted average of the spectral reconstructions obtained from two training iterations is used.
    Test MRAE = 0.4160
  - Fast-n-Squeeze: a weighted average of the spectral reconstructions obtained from Fast and Squeeze is used.
    Test MRAE = 0.3647
  - Fast-n-Squeeze (lower): an ideal lower-bound of Fast-n-Squeeze is computed, assuming the existence of an oracle that determines for each input image whether to use Fast or Squeeze.
    Test MRAE = 0.2915
- **Representative image / diagram of the method(s):** see Figure 12.

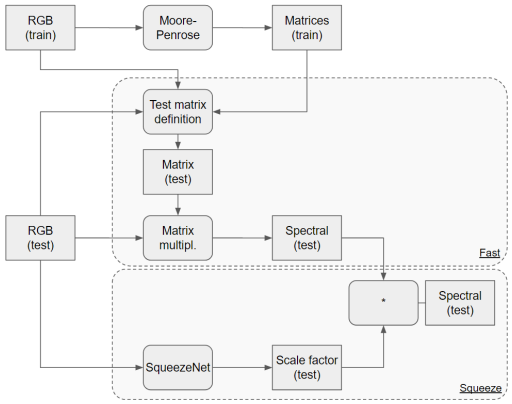


Figure 12. Schematic representation of Fast-n-Squeeze: our solution for spectral reconstruction from RGB data.

- **Training**

  The training of the proposed method is divided into two phases:

  In the first phase, pseudo-inverse RGB-to-spectral matrices [37] are optimized for each training image. This is at the core of all presented solutions.

  In the second phase, specific for the Squeeze and subsequent solutions, the parameters of the SqueezeNet-v1.1 model are finetuned using the Mean Relative Absolute Error (MRAE) as loss function. We train the model for a total of 30 epochs by using Adam optimizer with starting learning rate of 1×10^{-4} which decays by a factor of 0.5 every 10 epochs, a batch-size equal to 16, and exponential decay rates β_1 and β_2 equal to 0.9 and 0.999. Each RGB image feeded to the model is normalized by its global maximum value and then resized to $256px$ resolution using bilinear interpolation. We randomly apply horizontal and vertical flip and rotate of an angle between 0 and 360 degrees. At the end of each epoch, the MRAE is estimated on the validation set. The model that achieves the lowest MRAE is chosen as best model.

- **Testing** In the testing phase we first select the respective pseudo-inverse matrix for each image using the Fast method. Consequently the Squeeze algorithm estimates the global scale factor to be applied to this pseudo-inverse matrix to better match the spectral representation of the RGB image.

- **Ensembles and fusion strategies** Our first solution, named Fast, achieves MRAE 0.4629. It does not exploit any form of ensamble or fusion.

  Our second solution, named Squeeze, achieves MRAE 0.4160. It is computed as a weighted average between two training iterations of the SqueezeNet model:

$$\text{Squeeze} = \frac{1}{3}\text{Squeeze}_{(1)} + \frac{2}{3}\text{Squeeze}_{(2)} \qquad (14)$$

  The weights were empirically set.

  Our third solution, named Fast-n-Squeeze, exploits the uncorrelated nature of Fast and Squeeze to improve upon the spectral reconstruction of both, by resorting to a weighted average:

$$\text{Fast-n-Squeeze} = \frac{1}{2}\text{Fast} + \frac{1}{2}\text{Squeeze} \qquad (15)$$

The weights were also empirically set. The Fast-n-Squeeze solution achieves MRAE = 0.3647, which is a 21% improvement over Fast, and a 12% improvement over Squeeze.

Our fourth solution is intended as an hypothetical lower bound: for each test image, the best solution between Fast and Squeeze is selected assuming the availability of an oracle. This configuration achieves MRAE = 0.2915, highlighting the potential of the proposed solution in case of a classifier trained to identify two different classes of images, and suggesting a direction for future developments.

- **Performance Analysis**

  We implement the proposed method in Python3.8 using the PyTorch package with CUDA-v11.6 as backend. The proposed model is trained on a workstation equipped with an Intel i7-4770 CPU @3.40GHz, 16GB DDR4 RAM 2400MHz, NVIDIA GeForce GTX 1080 GPU with 2560 CUDA cores.

  The training of the SqueezeNet takes about 30 minutes.

  When processing 512×482 input images, the proposed method provides 104.71 FPS real-time processing performance on an NVIDIA GeForce GTX 1080.

  The proposed method is lightweight and efficient and can be run on devices with limited computational resources. It consists of the matrix multiplication of the RGB triplets by the pseudo-inverse and by the forward pass of SqueezeNet. The first step depends on the resolution of the input image. The forward pass of the SqueezeNet, which is the most computationally expensive operation of our method, is resolution-independent. In fact, the image is fed to the CNN at the fixed size of 256×256 pixels.

  Therefore, our method can scale to high resolution images without problems of insufficient GPU memory and at a low impact on the inference time.

6.8. SGG_RS_Whu: PoNet+: A Physical Optimization-based Network with Spectral Grouping for Spectral Recovery

A physical optimization-based spectral recovery methods is unrolled into an end-to-end CNN as our previous work PoNet [21]. Besides, we employed the spectral grouping similar to HSRnet [20].

6.8.1 Physical Optimization Unrolling

Let $X \in \mathbb{R}^{W \times H \times C}$ represent the observed HSI, where C is the number of the spectral channels, and W and H are the width and height, respectively. $Y \in \mathbb{R}^{W \times H \times c}$ represents the observed multispectral image, where $c < C$ is the number of multispectral bands, specifically for RGB image, with $c = 3$. Varying in SRF, the sensors obtain different MS or HS data with different bands. A transformation matrix $\Phi \in \mathbb{R}^{c \times C}$ can be used to describe the spectral degradation between MS and HS imaging as follows.

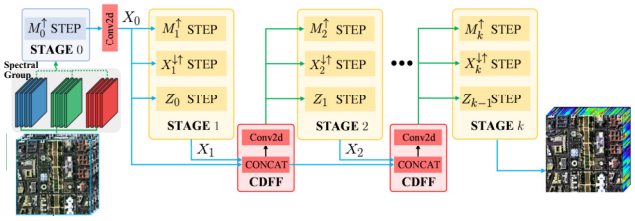


Figure 13. The framework of the proposed PoNet+.

$$Y = \Phi X \tag{16}$$

The high-dimension HSIs can be approximately predicted by adopting some priors to a minimization problem to constrain the solution space as follows:

$$\hat{X} = \arg\min_X \|Y - \Phi X\|_2^2 + \lambda \mathcal{R}(X) \tag{17}$$

where λ is a trade-off parameter, and $\mathcal{R}(\cdot)$ is a regularization function. Employ the half-quadratic splitting method with a penalty parameter as μ and solve it by the gradient descent algorithm:

$$\hat{X}_{k+1} = (1 - \epsilon\mu) X_k - \epsilon X_k \Phi\Phi^T + \epsilon M_H \Phi^T + \epsilon\mu Z_k \tag{18}$$

$$\hat{Z}_k = Prox(X_k) = \arg\min_Z \|Z - X_k\|_2^2 + \frac{\lambda}{\mu}\mathcal{R}(Z) \tag{19}$$

where ϵ is the optimization stride. As for the Z-subproblem, proximal operators that impose prior knowledge can deal with it.

Unrolling the physical optimization method into CNN, the proposed PoNet+ is shown in Fig. 13.

6.8.2 Cross-Dimensional Channel Attention

In traditional physical optimization-based algorithms, hyperparameters need to be defined manually and adjust to the optimal through a large number of experiments. Furthermore, in spectral super-resolution, differential treatment should be performed for the hyperparameters of different channels due to the different radiation characteristics.

Pooling is a common operation used in traditional channel attention, which is popular for fast computation and no parameter requirement at the cost of high information loss.

Furthermore, traditional channel attention weights the different channels of features separately ignoring the interaction between channels. There have been many works stated that building relationships between every two channels is much of importance. However, when the number of channels is large and attention mechanisms are frequently employed, the problem of computational burden should also be focused on.

Inspired by the above-mentioned points, we proposed a strategy named *Cross-Dimensional Channel Attention* (CDCA) employing 1D and 2D convolutional layers to manage the hyperparameter learning in this paper. 2D convolutional layers are used to extract pixel-by-pixel attention maps. On the other hand, 1D convolutional layers are employed to integrate attention maps for fast computational speed. Details of the proposed module are shown in Fig. 14.

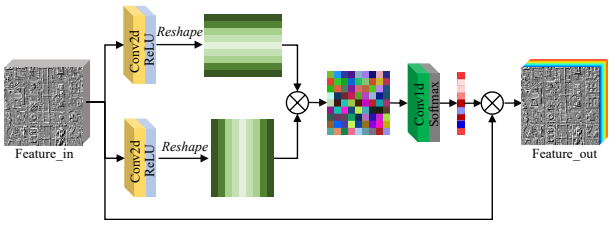


Figure 14. Cross-dimensional channel attention

We adopt two 2D convolutional layers with the kernel size of 1×1 to extract different spectral features $R, S \in \mathbb{R}^{W \times H \times C}$. Attention map $A \in \mathbb{R}^{C \times C}$ between any two channels can be calculated as follows:

$$a_{ij} = R_i S_j^T \tag{20}$$

where a_{ij} measures the attention between the i^{th} and j^{th} bands. R_i and S_j is the reshaped channel. To boost the computational speed, we employ a 1D convolutional layer with the kernel size of k to integrate channel-to-channel attention map A. Then, the final cross-dimensional channel attention-based hyperparameter $P \in \mathbb{R}^{1 \times C}$ will be obtained after a softmax layer:

$$p_j = \frac{\exp\left(W^{1\mathrm{d}} * A_j + b^{1\mathrm{d}}\right)}{\sum_{j=1}^{C} \exp\left(W^{1\mathrm{d}} * A_j + b^{1\mathrm{d}}\right)} \tag{21}$$

where W^{1d} and b^{1d} mean the kernel weights and biases for the 1D convolutional layer, and $p_j \in P$ is the parameter for the j^{th} band. In this way, we build a learnable end-to-end CNN by unrolling the physical optimization algorithm, which keeps the advantages of deep learning and physical model-based algorithm.

6.8.3 Cross-Depth Feature Fusion

In deep learning-based algorithms, the depth makes much sense for the network effect, in other words, the deeper networks get the better results. However, the shallow features are also very important. In the proposed method, we get multiple updated results at different depths. To improve the model memory of shallow features, a strategy named *Cross-Depth Feature Fusion* (CDFF) is proposed as shown in Fig. 13.

Given a set of intermediate results $\{X_0, X_1, X_2, \cdots X_{k-1}\}$, PoNet+ firstly concatenates results at different depths:

$$F_{k-1}^{C} = Concat(X_0, X_1, X_2, \cdots X_{k-1}) \tag{22}$$

Then, a convolutional layer with ReLU is also employed to fuse cross-depth features to obtain the input for the next stage.

$$X_k^{In} = ReLU(W_{k-1}^{F} * F_{k-1}^{C} + b_{k-1}^{F}) \tag{23}$$

where X_k^{In} means the input feature for the k^{th} optimization stage. Acquiring various information from cross-depth features, X_k^{In} can represent more comprehensive spectral information from shallow and deep features, which is beneficial to the subsequent optimization.

6.8.4 Spectral Grouping

Spectral grouping is achieved by grouping bands with spectral relevance according to spectral response functions (SRFs). The spectral grouping is used to avoid reconstruction distortion caused by the excessive spectral difference between different channels. Nevertheless, it seems inevitable that there still will be some differences between bands in the same group. The proposed strategy ensures that intra-group bands reconstruction is determined by the same combination of multispectral channels. By roughly representing spectral relevance from the similarity of imaging according to spectral response functions, SRF-guided convolutional layers don't have to be adjusted for the same sensor, which improves the generalization of this module. Note that, PoNet is a universal network presented to address generalized spectral super-resolution, including classical spectral super-resolution (sSR), FusSR, and PansSR. So, we only employed the PoNet in sSR. A representative diagram of the method is shown in Fig. 13.

- **Training**

  We directly read 900 images into ".h5" file with the original size (512×482). And in each iteration, we just input one image pairs. Training loss is ℓ_1 loss function.

- **Testing** The proposed method is an end-to-end network, we just generate the recovered hyperspectral images by feeding the test image into the trained model.

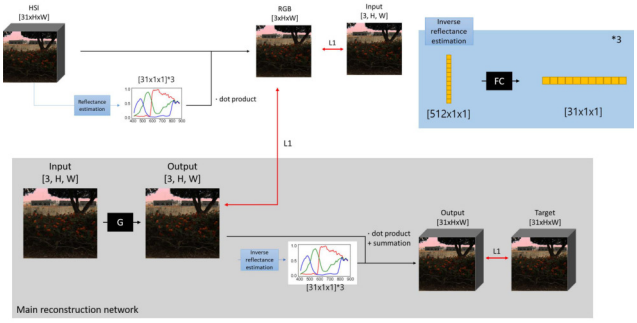


Figure 15. Representative training scheme for proposed method.

6.9. star-spectral: U-net with Learnable Inverse RGB Filter

We use 3 fully-connected layer to construct inverse R, G, B filter for 31 channels. The output of each fully-connected layer has size of (batch-size, 31) and it represents inverse R or G or B filter to reconstruct spectral information from R or G or B channel. After we get the output of each fully-connected layers, we apply dot-product between inverse filter and corresponding color channel. After that, we summate dot-producted output.

The proposed inverse RGB filter preserves input spatial resolution. Therefore, if the spatial resolution of the input image is same as target image. Then we look forward to neural network learns to reconstruct spectral information much easily.

- **Training**

  Our training is composed of two parts to train each goal. First, we reconstruct RGB ground truth data by applying learnable RGB filter to spectral ground truth. Then we trained DenseNet to translate input image to RGB ground truth. After that, we fixed pre-trained DenseNet, we trained inverse RGB filter to reconstruct channel-wise information by giving L1 loss between spectral ground truth. Fig. 15 describes the architecture of the proposed method. The complexity increases due to additional usage on inverse RGB filter compared to using a simple CNN model.

- **Testing**

  After training, we directly apply trained model to validation RGB data and evaluate PSNR, SSIM, and MRAE to see how the trained network reconstruct spectral information well.

- **Ensembles and fusion strategies**

  We ensemble CNN(DenseNet) to reconstruct spatial information and inverse RGB filter to reconstruct spectral information. By assembling these architectures, both networks are easy to reconstruct hyper-spectral image much easily.

6.10. NTU607QCO-Spectral: Knowledge transferring and edge preserving loss functions for spectral reconstruction

We use the MSBDN [16] as backbone. We adjust the final output layer as 31 to fit this track. Furthermore, during training, we apply the knowledge transfer [14, 55] technique to improve the model performance. That is, we first train several model with different initialization and optimizer. And then we fine-tune all model and add the loss to regularize the model. We not only desire the predicted images are identical to ground truth but also the average predictions. The loss functions $L_{transfer}$ for certain model j is written as:

$$L_{transfer}(y_j, \hat{y}) = L(y_j, \hat{y}) + L(y_j, \frac{\sum_0^n \hat{y}_i}{n}) \tag{24}$$

where L means the edge-preserving loss function, y_i means the predictions from model i, and the $\hat{y}$ is the ground truth image. With learn multiple characteristics from different model, the performance of single model can be increased. We use loss function in [15, 48] as edge preserving loss function.

Training. During the training phase, we randomly crop the image as 512x512 to optimize two models. We use Adamw and SGD optimizer with the learning rate of 0.0001 and the learning rate decrease of 0.1 every 1000 epochs. The total epoch is 10000 and takes 7 days. We set the batch size as 10. And then we use the knowledge transfer [55] to finetune all model. We set learning rate of 0.00001 for 1000 epochs and takes 2 days.

Testing. We use two models for evaluation. Similarly to training phase, the images are divided into two 512x512 images with overlaying pixels. We predict two images and then merge them.

6.11. OPT-KLSIT: An interpretable hyperspectral reconstruction network from RGB images

For high-quality spectral reconstruction from RGB images, we propose an interpretable mixed regression network (IMRnet), as shown in Figure 16. Specifically, based on the fact that natural scene hyperspectral is essentially on a low-dimensional manifold [25], we constructed a 3d-encoder network that could encode hyperspectral into a low-dimensional embedding space. The key of this 3d-encoder is to use three 3*3*200 3d-convolution kernels to simulate spectral response function with physical characteristics to reduce the dimension. Then, we use a decoder network with a self-attentional mechanism [30] to recover the original hyperspectral data from 3D embedding space. See the 3d-encoder module and SA-decoder module in Figure 16. The

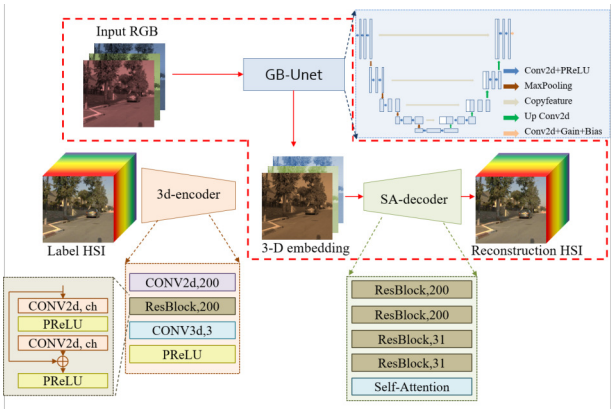


Figure 16. An Interpretable mixed Regression Network.

whole autoencoder network uses the residual network with PReLU activation function as the basic architecture to stabilize the network. Next, we use an modified Unet network (GB-Unet) with hyperparameter bias and gain to complete the nonlinear transformation from RGB to the corresponding 3D embedding space. Finally, the SA-decoder module is combined to complete the reconstruction from RGB to hyperspectral.

It is worth noting that only hyperspectral data as input in our autoencoder, which is useful for robust reconstruction. In other words, the optimized 3d-encoder and SA-decoder network are equivalent to a feature extractor and can be applied to any hyperspectral data. In the reconstruction stage, only the mapping of RGB to 3D embedding space needs to be optimized. This greatly reduces the workload for spectral reconstruction from RGB Images. More importantly, it avoids solving complex three-to-many mapping problems directly and transforms complex problems into simple convex problems of 3 to 3.

- **Training**

  In the training process, we use MRAE Loss to optimize the network. What's more, we use the joint training method of freezing and unfreezing to further improve the reconstruction accuracy. Specifically, the 3d-encoder and SA-decoder are first trained jointly. Then, the GE-Unet network is combined with the SA-decoder. Next, the optimized SA-decoder parameters are imported and frozen. Begin to optimize the GE-Unet to realize the nonlinear conversion from RGB to 3D-embedded space. Finally, when the loss is reduced to a certain precision, unfreeze the SA-decoder network to further optimize the whole network. This training process can effectively guide the development of the reconstructed network towards the low-dimensional representation based on mathematical model. The accuracy is higher than that of direct blind training under the same parameters.

  The input RGB image and output spectral images were randomly cropped to a 64×64 region, then rescaled to [0, 1]. The parameters of network are Xavier initialized. The learning rate is initialized as 0.0001 the polynomial function is set as the decay policy with power = 1.5. For optimization, the Adam [26] optimizer was used with $\beta 1 = 0.5$, $\beta 2 = 0.999$ and a batch size of 36. Reflection padding was used in the system to avoid border artifacts. All the experiments were performed on 1 NVIDIA RTX 3090 GPU.

- **Testing**

  Testing Instead of cropping image into small blocks, the complete RGB image is used to get a complete spectral image on an NVIDIA RTX 3090. The IMRnet takes 0.67s per image

- **Ensembles and fusion strategies**

  Our three-step training method can effectively guide the development of the reconstructed network to the low-dimensional representation based on mathematical model. Under the same parameters, the training accuracy of this method is higher than that of direct blind training.

  We used AWAN [30], the 2020 champion, and the general Unet network [41] as a baseline.

6.12. Image Lab: Dual Residual Channel Attention Net for Spectral Reconstruction

We proposed a novel deep dual residual channel attention(DRCA) network is presented for spectral reconstruction from RGB images shown in Figure 17. The Network has two-level feature extraction mechanisms. The input image is passed to the Coordinate convolution layer to improve the spatial information. The output of the coordinate convolution layer is connected with a densely connected block and a sequence of six DRCA blocks. The densely connected block contains the four convolution blocks, and the output channel of the convolution layer is set as 128. The block diagram of the DRCA block is shown in the Figure 18. In the second level, the input of the $DRCA_i$ block and $DRCA_{i+1}$ are added together. The same steps applied to the remaining DRCA blocks. The $DRCA_n$ connected to the convolution layer with 128 channels. The output of level one features and level two features output fused. The spectral bands are generated from the fused features.

- **Training**

  The shared data set consists of 899 images. We split into 675 images for training and 224 images for validation. Training images were cropped into sub-image

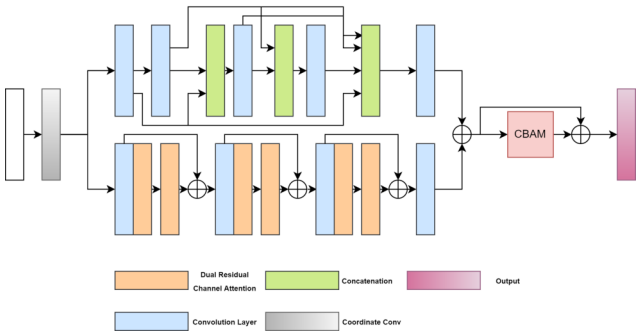


Figure 17. Architecture for Spectral Reconstruction

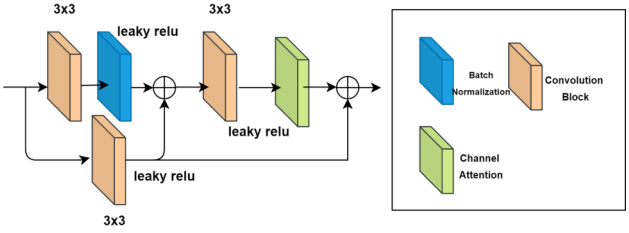


Figure 18. Dual Residual Attention Block

patches with a resolution of 64×64 and a batch size of 8 was selected empirically for stochastic gradient decent. The model is trained with 10784 training patches and 3584 validation patches. We used Adam optimizer with a learning rate of 0.001 to 0.00001 and 500 epochs for training the model. The proposed network was trained with the IntelCore i7 processor, RTX 2060 GPU, 8GB RAM, Platform keras

- **Testing**

  We extracted the patches from the test image in a non-overlapping mode and predicted them with a model.

Acknowledgments

We thank the NTIRE 2022 sponsors: Huawei, Reality Labs, Bending Spoons, MediaTek, OPPO, Oddity, Voyage81, ETH Zurich (Computer Vision Lab) and University of Wurzburg (CAIDAS).

A. Teams and affiliations

NTIRE2022 team

Members: *Boaz Arad[1,2](boazar@post.bgu.ac.il), Radu Timofte (radu.timofte@uni-wuerzburg.de),* Rony Yahel[1,2,5], Nimrod Morag[1,2,6], Amir Bernat[1,2]
Affiliations:
[1] Oddity tech Ltd.
[2] Voyage81 Ltd.
[3] Computation Vision Lab, ETH Zürich
[4] Center for Artificial Intelligence and Data Science, University of Würzburg
[5] The Academic College of Tel Aviv–Yaffo
[6] Tel Aviv University

MST++

Members: *Yuanhao Cai[1](cyh20@mails.tsinghua.edu.cn), Jing Lin[1], Zudi Lin[2], Haoqian Wang[1], Yulun Zhang[3], Hanspeter Pfister[2], Radu Timofte[3,4], Luc Van Gool[3]*
Affiliations:
[1] Shenzhen International Graduate School, Tsinghua University.
[2] Visual Computing Group, Harvard University.
[3] Computation Vision Lab, ETH Zürich.
[4] Center for Artificial Intelligence and Data Science, JMU Würzburg

MIALGO

Members: *Shuai Liu[1](liushuai21@xiaomi.com), Yongqiang Li, Chaoyu Feng, Lei Lei*
Affiliations:
[1] Xiaomi Inc., China

CVIA_SSR

Members: *Jiaojiao Li [1](jjli@xidian.edu.cn), Songcheng Du[1], Chaoxiong Wu[1], Yihong Leng[1], Rui Song[1]*
Affiliations:
[1] Xidian University, Xian, China

IFL

Members: *Mingwei Zhang[1](dlaizmw@gmail.com)*
Affiliations:
[1] Northwestern Polytechnical University, Xi'an, China

SSR

Members: *Chongxing Song[1](927879551@qq.com), Shuyi Zhao[1], Zhiqiang Lang[1], Wei Wei[1], Lei Zhang[1]*
Affiliations:
[1] Chang'an campus of Northwestern Polytechnical University

Yuelushan

Members: *Renwei Dian[1](drw@hnu.edu.cn), Tianci Shan[1], Anjing Guo[1], Chenguo Feng[1]*
Affiliations:
[1] Hunan University

IVL

Members: *Simone Bianco[1](simone.bianco@unimib.it), Mirko Agarla[1], Marco Buzzelli[1], Luigi Celona[1], Raimondo Schettini[1]*
Affiliations:
[1] Department of Informatics Systems and Communication, University of Milano - Bicocca

SGG_RS_Whu

Members: *Jiang He[1](jiang_he@whu.edu.cn), Yi Xiao[1], Jiajun Xiao[1], Qiangqiang Yuan[1], Jie Li[1], Liangpei Zhang[2]*
Affiliations:
[1] School of Geodesy and Geomatics, Wuhan University, China
[2] State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, China

star-spectral

Members: *Taesung Kwon[1](star.kwon@kaist.ac.kr), Dohoon Ryu[1], Hyokyoung Bae[1]*
Affiliations:
[1] Department of Bio and Brain Engineering, KAIST.

NTU607QCO-Spectral

Members: *Hao-Hsiang Yang[1](islike8399@gmail.com), Hua-En Chang[1], Zhi-Kai Huang[1], Wei-Ting Chen[2], Sy-Yen Kuo[1]*
Affiliations:
[1]Department of Electrical Engineering, National Taiwan University, Taiwan
[2]Graduate Institute of Electronics Engineering, National Taiwan University, Taiwan

OPT-KLSIT

Members: *Junyu Chen[1](chenjunyu2016@opt.cn), Haiwei Li[1], Song Liu[1]*
Affiliations:
[1] Xi'an Institute of Optics and Precision Mechanics of CAS, Xi'an, China

Image Lab

Members: *Sabarinathan[1](sabarinathantce@gmail.com), K Uma[2], B Sathya Bama[2],S. Mohamed Mansoor Roomi[2]*
Affiliations:
[1] Couger Inc,Japan [2] Thiagarajar College of Engineering , India

References

[1] Jonas Aeschbacher, Jiqing Wu, and Radu Timofte. In defense of shallow learned spectral reconstruction from rgb images. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 471–479, 2017. 1

[2] Mirko Agarla, Simone Bianco, Marco Buzzelli, Luigi Celona, and Raimondo Schettini. Fast-n-squeeze: towards real-time spectral reconstruction from rgb images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 4, 11

[3] 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. 1, 7

[4] Boaz Arad, Ohad Ben-Shahar, Radu Timofte, et al. NTIRE 2018 challenge on spectral reconstruction from rgb images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 1, 2, 3

[5] Boaz Arad, Radu Timofte, Ohad Ben-Shahar, Yi-Tun Lin, Graham D Finlayson, et al. NTIRE 2020 challenge on spectral reconstruction from an RGB image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020. 1, 2, 3, 4, 5

[6] Boaz Arad, Radu Timofte, Rony Yahel, Nimrod Morag, Amir Bernat, et al. NTIRE 2022 spectral demosaicing challenge and dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2

[7] Boaz Arad, Radu Timofte, Rony Yahel, Nimrod Morag, Amir Bernat, et al. NTIRE 2022 spectral recovery challenge and dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2

[8] Goutam Bhat, Martin Danelljan, Radu Timofte, et al. NTIRE 2022 burst super-resolution challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2

[9] Yuanhao Cai, Jing Lin, Xiaowan Hu, Haoqian Wang, Xin Yuan, Yulun Zhang, Radu Timofte, and Luc Van Gool. Mask-guided spectral-wise transformer for efficient hyperspectral image reconstruction. In *CVPR*, 2022. 7

[10] Yuanhao Cai, Jing Lin, Zudi Lin, Haoqian Wang, Yulun Zhang, Hanspeter Pfister, Radu Timofte, and Luc Van Gool. Mst++: Multi-stage spectral-wise transformer for efficient spectral reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2, 4, 5

[11] Ayan Chakrabarti and Todd Zickler. Statistics of real-world hyperspectral images. In *CVPR 2011*, pages 193–200. IEEE, 2011. 7

[12] Chein-I Chang. *Hyperspectral data exploitation: theory and applications*. John Wiley & Sons, 2007. 1

[13] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *CVPRW*, 2021. 7

[14] Wei-Ting Chen, Zhi-Kai Huang, Cheng-Che Tsai, Hao-Hsiang Yang, Jian-Jiun Ding, and Sy-Yen Kuo. Learning multiple adverse weather removal via two-stage knowledge learning and multi-contrastive regularization: Toward a unified model. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 14

[15] Wei-Ting Chen, Cheng-Che Tsai, Hao-Yu Fang, I Chen, Jian-Jiun Ding, Sy-Yen Kuo, et al. Contourletnet: A generalized rain removal architecture using multi-direction hierarchical representation. *arXiv preprint arXiv:2111.12925*, 2021. 14

[16] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang. Multi-scale boosted dehazing network with dense feature fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2157–2167, 2020. 14

[17] Egor Ershov, Alex Savchik, Denis Shepelev, Nikola Banic, Michael S Brown, Radu Timofte, et al. NTIRE 2022 challenge on night photography rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2

[18] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE transactions on pattern analysis and machine intelligence*, 43(2):652–662, 2019. 9

[19] Jinjin Gu, Haoming Cai, Chao Dong, Jimmy Ren, Radu Timofte, et al. NTIRE 2022 challenge on perceptual image quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2

[20] Jiang He, Jie Li, Qiangqiang Yuan, Huanfeng Shen, and Liangpei Zhang. Spectral response function-guided deep optimization-driven network for spectral super-resolution. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 12

[21] Jiang He, Qiangqiang Yuan, Jie Li, and Liangpei Zhang. Ponet: A universal physical optimization-based spectral super-resolution network for arbitrary multispectral images. *Information Fusion*, 80:205–225, 2022. 12

[22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 5, 10

[23] Xiaowan Hu, Yuanhao Cai, Jing Lin, Haoqian Wang, Xin Yuan, Yulun Zhang, Radu Timofte, and Luc Van Gool. Hdnet: High-resolution dual-domain learning for spectral compressive imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[24] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 11

[25] Yan Jia, Yinqiang Zheng, Lin Gu, Art Subpa-Asa, Antony Lam, Yoichi Sato, and Imari Sato. From rgb to spectrum for natural scenes via manifold-based mapping. In *Proceedings of the IEEE international conference on computer vision*, pages 4705–4713, 2017. 14

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 15

[27] Alexander Kokka, Hans Toivanen, Rami Mannila, and Antti Näsilä. High-resolution hyperspectral imager based on tunable fabry-pérot interferometer filter technology. In *Photonic Instrumentation Engineering IX*, volume 12008, pages 39–46. SPIE, 2022. 1

[28] Jiaojiao Li, Songcheng Du, Chaoxiong Wu, Yihong Leng, Rui Song, and Yunsong Li. Drcr net: Dense residual channel re-calibration network with non-local purification for spectral super resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 4, 7

[29] Jiaojiao Li, Chaoxiong Wu, Rui Song, Yunsong Li, and Fei Liu. Adaptive weighted attention network with camera spectral sensitivity prior for spectral reconstruction from RGB images. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, pages 462–463, 2020. 9

[30] Jiaojiao Li, Chaoxiong Wu, Rui Song, Yunsong Li, and Fei Liu. Adaptive weighted attention network with camera spectral sensitivity prior for spectral reconstruction from rgb images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 462–463, 2020. 14, 15

[31] Yawei Li, Kai Zhang, Radu Timofte, Luc Van Gool, et al. NTIRE 2022 challenge on efficient super-resolution: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2

[32] Jing Lin, Yuanhao Cai, Xiaowan Hu, Haoqian Wang, Xin Yuan, Yulun Zhang, Radu Timofte, and Luc Van Gool. Coarse-to-fine sparse transformer for hyperspectral image reconstruction. *arXiv preprint arXiv:2203.04845*, 2022. 2

[33] Yi-Tun Lin and Graham D Finlayson. Investigating the upper-bound performance of sparse-coding-based spectral reconstruction from rgb images. In *Color and Imaging Conference*, volume 2021, pages 19–24. Society for Imaging Science and Technology, 2021. 2

[34] Andreas Lugmayr, Martin Danelljan, Radu Timofte, et al. NTIRE 2022 challenge on learning the super-resolution space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2

[35] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. Single image super-resolution via a holistic attention network. In *European conference on computer vision*, pages 191–207. Springer, 2020. 7, 9

[36] Manu Parmar, Steven Lansel, and Brian A Wandell. Spatio-spectral reconstruction of the multispectral datacube using sparse recovery. In *2008 15th IEEE International Conference on Image Processing*, pages 473–476. IEEE, 2008. 1

[37] Roger Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge University Press, 1955. 11

[38] Eduardo Pérez-Pellitero, Sibi Catley-Chandar, Richard Shaw, Ales Leonardis, Radu Timofte, et al. NTIRE 2022 challenge on high dynamic range imaging: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2

[39] Antonio Robles-Kelly. Single image spectral reconstruction for multimedia applications. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 251–260. ACM, 2015. 1

[40] Andres Romero, Angela Castillo, Jose M Abril-Nova, Radu Timofte, et al. NTIRE 2022 image inpainting challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 15

[42] Zhan Shi, Chang Chen, Zhiwei Xiong, Dong Liu, and Feng Wu. Hscnn+: Advanced cnn-based hyperspectral recovery from rgb images. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, pages 939–947, 2018. 9

[43] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision*, pages 111–126. Springer, 2014. 1

[44] Radu Timofte, Rasmus Rothe, and Luc Van Gool. Seven ways to improve example-based single image super resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1865–1873, 2016. 7

[45] Longguang Wang, Yulan Guo, Yingqian Wang, Juncheng Li, Shuhang Gu, Radu Timofte, et al. NTIRE 2022 challenge on stereo image super-resolution: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2

[46] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 11531–11539, 2020. 9, 10

[47] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 7794–7803, 2018. 9

[48] Hao-Hsiang Yang, Chao-Han Huck Yang, and Yi-Chang James Tsai. Y-net: Multi-scale feature aggregation network with wavelet structure similarity loss function for single image dehazing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2628–2632. IEEE, 2020. 14

[49] Ren Yang, Radu Timofte, et al. NTIRE 2022 challenge on super-resolution and quality enhancement of compressed video: Dataset, methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 2

[50] Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K Nayar. Generalized assorted pixel camera: post-capture control of resolution, dynamic range, and spectrum. *IEEE transactions on image processing*, 19(9):2241–2253, 2010. 7

[51] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, 2022. 7, 10

[52] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *ECCV*, 2020. 7

[53] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021. 7

[54] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018. 10

[55] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328, 2018. 14