NTIRE 2018 Challenge on Spectral Reconstruction from RGB Images

Boaz Arad Ohad Ben-Shahar Radu Timofte Luc Van Gool Lei Zhang
Ming-Hsuan Yang Zhiwei Xiong Chang Chen Zhan Shi Dong Liu Feng Wu
Charis Lanaras Silvano Galliani Konrad Schindler Tarek Stiebel Simon Koppers
Philipp Seltsam Ruofan Zhou Majed El Helou Fayez Lahoud Marjan Shahpaski
Ke Zheng Lianru Gao Bing Zhang Ximin Cui Haoyang Yu Yigit Baran Can
Aitor Alvarez-Gila Joost van de Weijer Estibaliz Garrote Adrian Galdran
Manoj Sharma Sriharsha Koundinya Avinash Upadhyay Raunak Manekar
Rudrabha Mukhopadhyay Himanshu Sharma Santanu Chaudhury
Koushik Nagasubramanian Sambuddha Ghosal Asheesh K. Singh Arti Singh
Baskar Ganapathysubramanian Soumik Sarkar

Abstract

This paper reviews the first challenge on spectral image reconstruction from RGB images, i.e.*, the recovery of whole-scene hyperspectral (HS) information from a 3-channel RGB image. The challenge was divided into 2 tracks: the "Clean" track sought HS recovery from noiseless RGB images obtained from a known response function (representing spectrally-calibrated camera) while the "Real World" track challenged participants to recover HS cubes from JPEG-compressed RGB images generated by an unknown response function. To facilitate the challenge, the BGU Hyperspectral Image Database [4] was extended to provide participants with 256 natural HS training images, and 5+10 additional images for validation and testing, respectively. The "Clean" and "Real World" tracks had 73 and 63 registered participants respectively, with 12 teams competing in the final testing phase. Proposed methods and their corresponding results are reported in this review.*

1. Introduction

Hyperspectral imaging systems (HISs) record the complete spectral signature reflected from each observable point in a given scene. While HISs have been available since the 1970s [8], recent technological advances have reduced their cost and made them accessible to a growing number of researchers and industrialists. Despite their increasingly lower cost, most HISs still rely on either spatial or spectral scanning (via push-broom or filter-wheel principles) in order to acquire complete hyperspectral (HS) images. This inherent limitation of traditional HISs makes them unsuitable for rapid acquisition, or acquiring scenes which contain moving objects. In addition, most HISs are still too physically large and heavy to fit most portable platforms such as drones, smartphones, and other hand-held devices.

A number of approaches have been employed in order to produce "snapshot" or video-capable HISs. They include computed-tomography imagers [22], mosaic cameras [17], hybrid RGB-HS systems [16] and others. In this challenge we focus on one of the more recent approaches: the recovery of visual-spectrum 'hyperspectral' images from RGB-only input.

The benefit of HS-from-RGB systems is twofold: (i) representing RGB images by their source HS signals allows the application of existing HS detection/analysis methods to data which could not be acquired by a HIS, while (ii) studying the failure cases of these systems can allow us to improve the spectral resolution of camera systems via improved design [5].

In natural images, reconstruction of hyperspectral images from RGB data is often accomplished by the use of sparse coding, learning via neural networks, or a combination of the two. While earlier methods relied on PCA basis to recover spectra from RGB or other multispectral data [20, 2], they were quickly outperformed by methods which leveraged sparse coding [23]. However, in recent years, natural hyperspectral image databases of growing size and resolution have become more prevalent (*e.g.*, 32 images recorded by Yasuma *et al.* [30], 66 im-

B. Arad (boazar@bgu.ac.il, Ben-Gurion University of the Negev), O. Ben-Sharar, R. Timofte, L. Van Gool, L. Zhang and M.-H. Yang are the NTIRE 2018 organizers, while the other authors participated in the challenge.
Appendix A contains the authors' teams and affiliations.
NTIRE webpage: http://www.vision.ee.ethz.ch/ntire18/

ages by Nguyen *et al*. [21], 77 images by Chakrabarti and Zickler [7], and 256 images recorded by Arad and Ben-Shahar [4]), thus allowing for trained neural net approaches, which became more popular.

Initially, shallow neural nets were used, such as the radial basis function (RBF) approach proposed by Nguyen*et al*. [21]. As increasing amounts of training data became available, much deeper networks were trained such the 18-layer GAN proposed by Alvarez-Gila *et al*. [3]. Alongside pure neural net approaches, sparse coding remains an active avenue of exploration. Robles-Kelly [25] proposed a hybrid sparse coding/neural net approach, while Aeschbacher *et al*. [1] demonstrated that sparse coding approaches (*i.e*. adjusted anchored neighborhood regression [28]) can achieve comparable performance to those based on neural nets.

Another notable trend in the previously mentioned works is a shift from evaluating performance on individual spectra samples (*i.e*., Parmar *et al*. [23]) to evaluating performance on whole images as well as sets of images (*i.e*., Arad and Ben-Shahar [4]). While performance evaluation of hyperspectral reconstruction algorithms has clearly become more comprehensive, performance metrics vary widely between researchers, in terms of both test data and evaluation metrics.

The NTIRE 2018 spectral reconstruction challenge offers the first large-scale, uniform benchmark for HS-from-RGB systems. Its two tracks aim to simulate HS reconstruction from a known, spectrally-calibrated system ("Clean"), as well as HS reconstruction "in the wild" from the processed output of an unknown camera saved in a lossy image format ("Real World"). The following sections will describe the challenge in detail, as well as the results and various methods used to attain them.

2. NTIRE 2018 Challenge

The objectives of the NTIRE 2018 challenge on spectral reconstruction from RGB images are: (i) to gauge and push the state-of-the-art in HS reconstruction from RGB; (ii) to compare different solutions; (iii) to expand the available databases for training/testing; (iv) to suggest a uniform method of performance evaluation.

2.1. BGU HS Dataset

The BGU HS dataset is the largest and most detailed natural hyperspectral image database collected to date. For the purpose of this challenge, the database has been extended to include 256 images with 53 new public images, and 15 new unreleased validation/test images. Several test images can be seen in figure 1.

Recent HS-from-RGB methods are often introduced with reported results for one or more of the existing databases: the BGU HS [4] database (203 images), the Chakrabarti [7] database (77 images), or the much smaller Yasuma [30] database (32 studio images). This variability hampers attempts to quantitatively compare different approaches, a problem further compounded by the variety of error metrics used in each evaluation (MRAE, RMSE, PSNR and others).

This NTIRE 2018 challenge suggests a uniform method for the evaluation of HS-from-RGB algorithms, providing the first equal-grounds comparison and overview of state-of-the-art approaches.

During the challenge, 5 of the validation images were publicly released - increasing the number of publicly available images to 261.

Figure 1. Test images from the "Clean" track (top row) and "Real World" track (bottom row). In this figure, image brightness has been increased for display purposes.

2.2. Tracks

Track 1: "Clean" aimed to simulate recovery of HS information from a known and calibrated RGB imaging system. Participants were provided with uncompressed 8-bit RGB images created by applying the CIE-1964 color-matching function to ground truth hyperspectral information.

Track 1: "Real World" aimed to simulate recovery of HS information from the processed output of an unknown camera system. Participants were provided with JPEG-compressed 8-bit RGB images created by applying an unknown camera response function to ground truth hyperspectral information.

Competitions A competition on the CodaLab platform was available for each track of the NTIRE 2018 spectral reconstruction challenge. Each participant was required to register in order to access the data and submit their estimated HS images results to the evaluation server.

Challenge phases *(1) Development (training) phase:* the participants were provided with both HS and RGB training images (256 pairs), as well as RGB validation images (5 images); *(2) Validation phase:* the participants had the opportunity to test their solutions on the RGB validation

team	user	Track 1: Clean MRAE	Track 1: Clean RMSE	Track 2: Real World MRAE	Track 2: Real World RMSE
VIDAR[1] [26]	ChangC	$0.0137_{(1)}$	14.45	$0.0310_{(1)}$	24.06
VIDAR[2] [26]	contstriver	$0.0139_{(2)}$	13.98	$0.0320_{(2)}$	25.01
HypedPhoti	phoi	$0.0153_{(4)}$	16.07	$0.0332_{(3)}$	27.10
LFB [27]	Tasti	$0.0152_{(3)}$	16.19	$0.0335_{(4)}$	26.44
IVRL Prime	zrfan	$0.0155_{(5)}$	16.17	$0.0358_{(6)}$	28.23
sr402	sr402	$0.0164_{(6)}$	16.92	$0.0345_{(5)}$	26.97
CVL [6]	baran	$0.0174_{(7)}$	17.27	$0.0364_{(7)}$	27.09
adv_rgb2hs	shuffle	$0.0218_{(9)}$	24.81	$0.0396_{(8)}$	34.05
CEERI [15]	harshakoundinya	$0.0181_{(8)}$	19.41		
	prakhar.amba	$0.0231_{(10)}$	17.70	$0.0480_{(9)}$	32.63
SPEC_RC	koushikn	$0.0401_{(11)}$	24.81	$0.0817_{(10)}$	49.96
	grimVision	$0.5772_{(12)}$	404.44		

Table 1. NTIRE 2018 Spectral Reconstruction Challenge results and final rankings on the BGU HS test data.

images and receive immediate feedback by uploading their results to the online server. A validation leaderboard was available as well; *(3) Final evaluation (test) phase:* HS validation images were released (5 images), alongside RGB test images (5 different images for each of the two tracks). Participants were required to submit their HS estimation for the RGB test images and a description (factsheet) of their methods before the challenge deadline. One week later the final results were made available to participants.

Evaluation protocol Mean Relative Absolute Error (MRAE) computed between the submitted reconstruction results and the ground truth images was selected as the quantitative measure for this competition. Root Mean Square Error (RMSE) was reported as well, but not used to rank results. MRAE was selected over RMSE as the evaluation metric, in order to avoid overweighting errors in higher luminance areas of the test images vs. those in lower luminance areas. MRAE and RMSE are computed as follows:

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

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

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. Challenge Results

From 73/63 registered participants on the "Clean"/"Real World" tracks, respectively, 12 teams entered in the final phase and submitted results, codes/executables, and factsheets. Table 1 reports the final scoring results of the challenge, Figure 2 shows MRAE heat maps for each solution on the same sampled test image, and Table 2 reports the runtimes and the major details for each entry. Section 4 describes briefly the methods for each team while Appendix A details team members and affiliations.

Architectures and main ideas All proposed methods relied on some form of convolutional neural network [18], with 7 entries using deep convolutional neural nets (CNNs), 2 using a generative adversarial network (GAN) [10] architecture, and one employing a residual dense concatenate SE network . Notably absent from the challenge are methods based on sparse coding, which have been previously demonstrated as suitable for this task [1, 4].

Runtime / efficiency 10 out of 12 participants reported their runtimes. When implemented on a GPU, proposed methods reported runtimes ranging from 0.57 seconds to $\sim$ 4 seconds. CPU implementations required up to 3 minutes. The most efficient implementations (HypedPhoti, CVL, LFB, VIDAR) are also among the top performing; Methods requiring longer runtime did not enjoy any performance advantage. While time efficiency was not a competition metric, both on CPU and on GPU, none of the methods showed suitable performance for real-time video applications. The relatively shallow net of CVL or shallow net versions of HypedPhoti and VIDAR (at the expense of performance) are capable of (near) real-time on GPU.

Train data Participants were provided with a total of 256 training images, at 1392×1300 resolution for a total of $4.6 \cdot 10^8$ hyperspectral pixels. All participants found the amount of data sufficient for training their model, though the HypedPhoti team reported that they would require more data for a network with more than 42 convolutional layers. Some teams (CVL, sr402, adv_rgb2hs) further augmented the training data [29] by using rotated patches and/or ran-

[1] Two entrants did not reveal their architecture.

Team	Reported runtime per image: Clean	Reported runtime per image: Real World	Platform	CPU	GPU	Notes	Ensemble/Fusion
VIDAR[1] [26]	∼3m	∼3m	Tensorflow	E5-2650	8× Tesla M40	GPU used only for training	Multiple
VIDAR[2] [26]	0.96s	0.96s	PyTorch		GTX 1080Ti		Multiple
HypedPhoti	0.66s	0.57s	Keras/Tensorflow		GTX 1080Ti		
LFB [27]	0.71s	0.84s	PyTorch	E5-2630	GTX 1080Ti		
IVRL Prime	∼2.5m / 1.5s	∼2.5m / 1.5s	Tensorflow	E5-2680	Titan X	Timed on CPU/GPU	
sr402	11.89s	11.91s	PyTorch	i7-5930k	4× GTX 1080Ti		Flip/Rotation
CVL [6]	0.67s	0.67s	Tensorflow		Titan X		8×flip/rotation
adv_rgb2hs	3.64s	3.64s	PyTorch		Titan X Pascal		
CEERI [15]	2.3s		Keras/Tensorflow				
SPEC_RC	∼8s	∼8s	Keras/Tensorflow	i7-6700k	Titan X		

Table 2. Reported runtimes per image on the BGU HS test data and additional details from the factsheets.

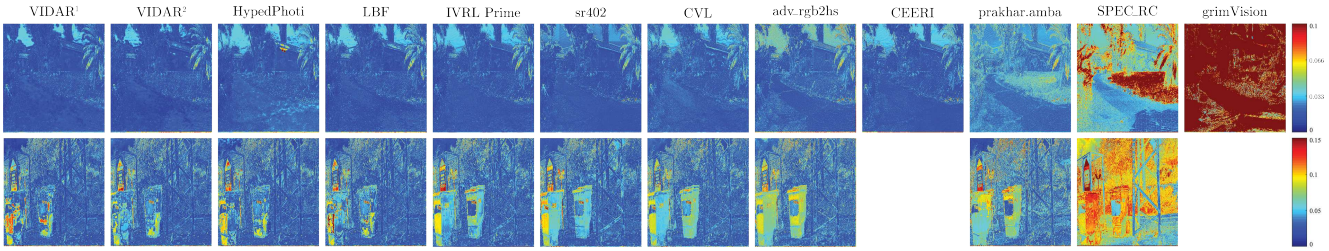


Figure 2. MRAE heat maps for all submitted methods relative to a sample image in both the "Clean" (top row) and the "Real World" (bottom row) tracks. Note that the error heat maps for each track have been scaled for optimal display.

dom crops to train their models.

Conclusions The solutions proposed by challenge competitors have not only improved upon state-of-the-art performance, but also present novel approaches which significantly expand upon previously published works. The challenge has succeeded in producing the first large-scale equal grounds comparison between hyperspectral-from-RGB reconstruction algorithms. And we hope it will serve as the basis for future comparison.

As the "Real World" track rankings are quite similar to those of the Clean track - it may be beneficial to simulate additional forms of camera noise and realistic combinations (*i.e.* shot-noise, "salt-and-pepper" noise, etc.) for this track in future challenges.

4. Challenge Methods and Teams

4.1. VIDAR [26]

4.1.1 Solution 1

An ensemble of three CNNs with densely-connected structures tailored for spectral reconstruction was used. Specifically, in each dense block, a novel fusion scheme was designed to widen the forward paths for higher capacity. An example of the network structure is shown in Fig. 3 and a typical setting of network parameters is listed in Table 3. The Adam solver was used for optimization with $\beta_1 = 0.9$ and the coefficient of weight decay (L_2 norm) was set as 0.0001. The global basic learning rate was 0.001 with a polynomial function as the decay policy. Training was stopped when no notable decay of training loss is observed. The algorithm proposed by He [12] was adopted for initializing weights and biases in each convolutional layer were initialized to zero. Training images were partitioned into sub-image patches with a resolution of 50×50 and a mini-batch number of 64 was selected empirically for stochastic gradient decent. The loss function adopted for training was MRAE. Training required approximately 38 hours using 8 Tesla M40 GPUs for a network with 38 dense blocks. During testing, a general CPU along with at least 32G memory is required for inference.

Feature extraction	Feature mapping	Reconstruction
$\begin{bmatrix} C(3\times3\times16/16) \\ C(1\times1\times16/16) \end{bmatrix}$	$\left\{ \begin{bmatrix} C(1\times1\times64) \\ C(3\times3\times16/16) \\ C(1\times1\times8/8) \end{bmatrix} \right\} \times 38$	$C(1\times1\times1)$
	$C(1\times1\times16)$	

Table 3. A typical setting of hyper-parameters for the network in VIDAR[1] solution. $C(\cdot)$ stands for the convolution with (kernel size × kernel size × filter number). $[\cdot]\times$ and $\{\cdot\}\times$ stand for concatenation operators with certain blocks ($\times 1$ is omitted). The symbol "/" denotes the parallel operation for path-widening fusion.

4.1.2 Solution 2

An ensemble of three CNNs with residual blocks tailored for spectral reconstruction was used. An example of the network structure is shown in Fig. 3. Typically, the filter number of each convolutional layer was set to 64. The

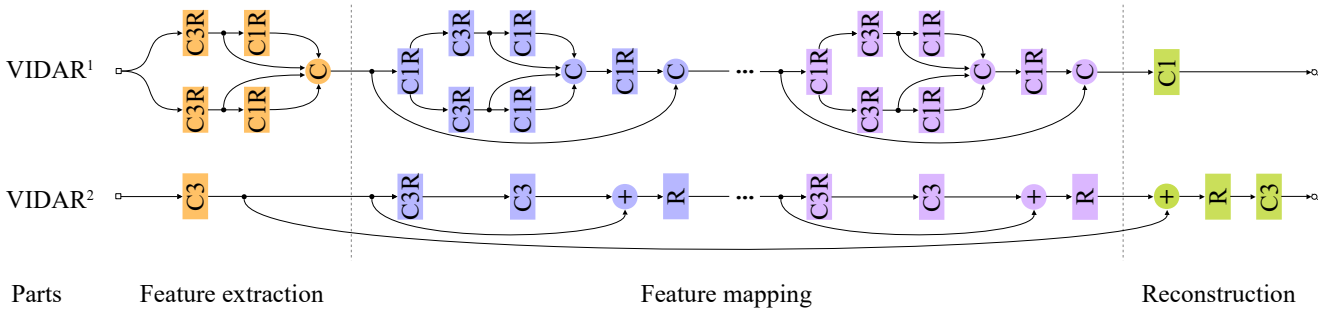


Figure 3. Network structures of the VIDAR team. The C with a rectangular block denotes convolution, and the following 1 and 3 denote the kernel size. The R represents an ReLU activation function. And the C with a circular block denotes concatenation.

Adam optimizer was used with $\beta_1 = 0.9$; $\beta_2 = 0.999$; and $e = 10^{-8}$. The initial learning rate was 2×10^{-4} with a polynomial function as the decay policy. Training was stopped after 1000 epochs. Weight initialization, patch size, batch size, and loss function were identical to those used in solution 1. The proposed network was trained on a single 1080Ti GPU. Approximately 60 hours were required for training a network with 16 residual blocks. During testing, at least 12G GPU memory is required for inference.

4.1.3 Ensemble

Two types of ensemble methods were adopted in the above two solutions. The first one is called self-ensemble. Specifically, the input image was flipped left/right to obtain a mirrored output. Then, the mirrored output and the original output were averaged into the target result. The second one is called model-ensemble, whose result is the linear combination of multiple models (e.g., three in the above two solutions) with different depths, filter numbers, or initializations. Please refer to [26] for complete technical details.

4.2. HypedPhoti

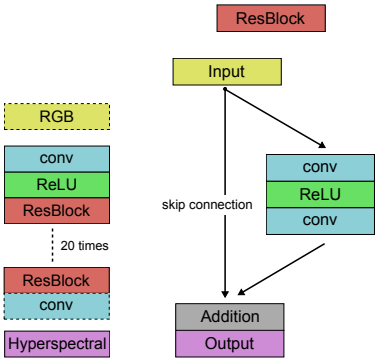


Figure 4. HypedPhoti network (left) and a residual block (right).

The HypedPhoti method consists of a deep, fully convolutional neural network that learns a patch-wise end-to-end mapping from RGB values to 31 spectral channels.

The network architecture is based on ResNet, but without any spatial striding or pooling. That choice was motivated by the intuition that fine-scale detail may be important and should not be lost through pooling. Moreover, for training 32×32 pixel patches were used, as no evidence was found to suggest that a larger context beyond a 32×32 spatial neighborhood improves spectral reconstruction. Batch normalization was not employed, since normalization reduces the network's ability to learn correlations between the spectral distribution and the local intensities (respectively, radiance values), potentially reducing its robustness against variations of individual images' intensity ranges. This reduced range flexibility was also reported in [19]. No data augmentation was performed. The complexity of the prediction appears to lie in the generalization across different spectral distributions, for which it is not obvious how to perform a synthetic, but nevertheless realistic augmentation. On the contrary, there seemed to be little use in augmenting with the usual spatial transformations, since invariance against them does not seem to be a limiting factor – and in fact could be potentially detrimental if there are directional effects in the spectral reflectance.

The employed network is a variant of the standard ResNet architecture. An illustration is given in Fig. 4. In particular, The network used in the final competition had a total of 42 convolutional layers, 40 of which are grouped into 20 so-called ResBlocks. Each ResBlock consists of 2 convolutions and a ReLU activation after the last convolution. Experiments with different numbers of ResBlocks were performed, and found that overfitting occurred when deeper networks were used. Signficantly shallower designs with only 6 ResBlocks performed quite well, too, whereas early stopping well before convergence on the training loss was necessary to get the best possible validation results with 20 ResBlocks. Hence, the possibility that the selected version with 20 ResBlocks is already overfitted to some bias of our rather small validation set (15 images) cannot be ruled out.

Different configurations were used for tracks 1 and 2. In the "Clean" setup all additive biases were removed from the

convolutions, since the model in forward direction (from hyperspectral to RGB) is purely multiplicative. However, in the "Real World" setup additive biases were included to compensate for compression artifacts.

4.3. LFB [27]

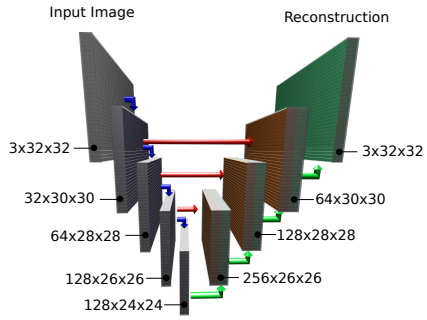


Figure 5. Visualization of the LFB network architecture.

A convolutional neural network is employed to solve the task of spectral reconstruction. The network architecture is based on a U-Net, which has been modified to perform a regression task. All pooling layers were removed and there is no batch normalization.

Figure 5 visualizes the architecture used for the Clean track. The red arrows represent a convolutional layer having a kernel size of 3, a unity stride and no zero-padding followed by a ReLU activation. The upward side of the network consists of corresponding transposed convolutions. The concatenation of such a transposed convolution and a ReLU activation is visualized by green arrows. Skip connections are added everywhere but in the uppermost layer and are visualized by red arrows. The very first applied convolution takes a three channel image as input, i.e. the RGB-image, and outputs 32 channels. The subsequent two convolutional layers each double the channel count up to a final count of 128. Afterwards, the channel count remains constant until it is reduced again in the upward path in a symmetric way to the downward path.

The architectures used for each track are slightly different. For the "Clean" track, a total amount of five layers was used, whereas a layer count of 6 was found to be optimal for the "Real World" track. In addition, a convolutional layer with a kernel size of 5 was added at the very start for the "Real World" track, taking an 3 channel RGB-image as input. This layer was added in order to increase robustness to noise and compression artifacts. The output of this preprocessing layer is fed into the actual network.

The final networks were trained from scratch on the entire data set provided within the challenge. A patch size of 32 and a batch size of 10 were used. All images were split into patches in a deterministic way, such that neighboring patches are located next to each other. Each model was trained for 5 epochs using the Adam optimizer with a learning rate of 0.0001 and, subsequently, for another 5 epochs using SGD with an initial Nesterov momentum of 0.9. The code was written in python using PyTorch.

4.4. IVRL Prime

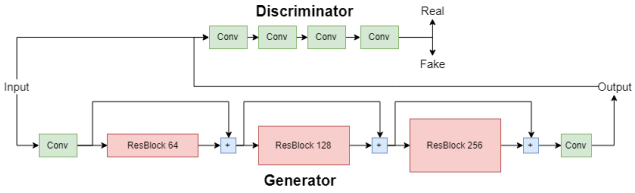


Figure 6. Illustration of the IVRL Prime GAN framework for spectral reconstruction. Note that the framework contains one generator network and two discriminator networks which help produce more realistic spectral reconstructions.

A generative adversarial framework [10] was used for spectral reconstruction as shown in Figure 6. This framework contained one generator network and two discriminator networks. 12 residual blocks [12] with increasing number of filters for the generator network were used to reconstruct spectral data from RGB images. The first residual blocks have 64 filters, and the number of filters is doubled after each 4 residual blocks. To help the generator network produce more realistic spectral reconstructions, 2 additiona discriminator networks were added to the framework. One discriminator network takes in all spectral bands of the output data and determines if it is realistic, the other discriminator network only "sees" the last 5 bands (660nm - 700nm) as these 5 bands are harder to reconstruct. The DCGAN [24] architecture was used for the discriminator network. A combination of MRAE and adversarial loss was used as the overall loss function of the framework. According to our experiments, adding adversarial loss provided 5% improvements in MRAE comparing to the pure residual network.

In the training dataset, 62 images which have a different context from that of the testing dataset were manually removed. 64×64 overlapping patches were cropped with a stride of 16 from the selected training dataset. Adam [14] was used for optimizing the network with $\beta_1 = 0.9$ and a learning rate of $1e-4$, the learning rate was halved after each 5000 batches.

4.5. sr402

To reconstruct the spectral image, a Residual Dense Concatenate SE Network was designed (Fig. 7). Inspired by DenseNet and SE-Net, the main idea is to collect the middle feature maps and pass these layers to the final convolution in order to maximize the amount of information used for reconstruction. Therefore, the model before final convolution can provide low-to-high feature maps.

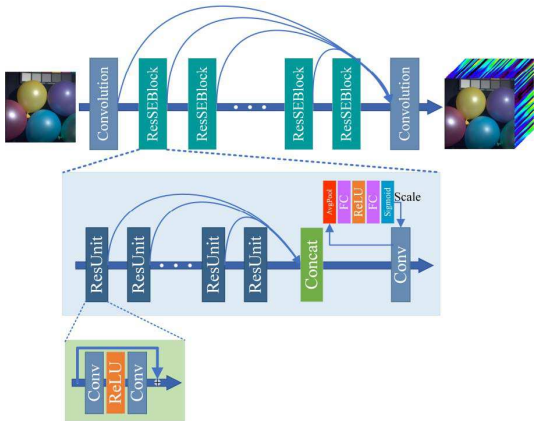
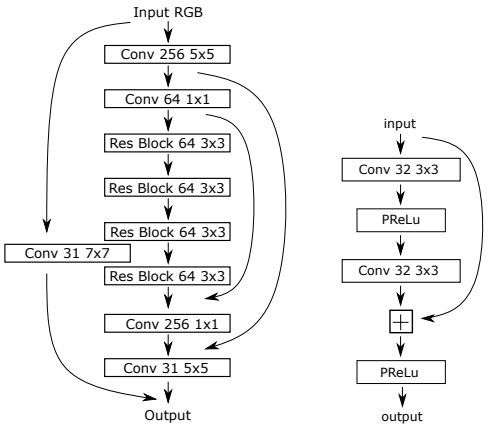


Figure 7. Residual Dense Concatenate SE-Net used by sr402.



(a) network (b) residual block

Figure 8. CVL solution: (a) network layout and (b) residual block.

Training As shown in the Fig. 7, the model is an end-to-end Convolution Neural Network. For the challenge only up to 8 ResSEBlock were explored, and each ResSEBlock was composed of 7 ResUnits with 64 feature maps. At training time, 64×64 input and target patches were randomly cropped from the training data. During each epoch, 32 patches were randomly cropped from each image. Mini-batch size was set to 4. Horizontal/vertical flips and 90° rotations were randomly used for each path. The model was trained with MRAE loss using the Adam optimizer with an initial learning rate of $4 \cdot 10^{-5}$. The learning rate decays by 0.65 after every 50 epochs, for a total of 500 epochs. The same network was used for each Track. For the Track 1 (Clean), model parameters were randomly initialed using PyTorch's default function. After training over Track 1, the resulting pretrained model was used to learn the Track 2 dataset (Real World).

The network was implemented in Pytorch on 4 NVIDIA 1080Ti (11G) GPUs. The training time for each Track was 12h.

Testing Due to the limitation of GPU memory, the input images from the validation/test sets were split into small patches and their output inferred. Finally all patches were stitched together. The inference-time per image was 11.91s for both validation data and test data.

4.6. CVL [6]

CVL proposed a moderately deep fully-convolutional CNN method (FCN) [6] aimed to learn the RGB-to-HS latent mapping while avoiding overfitting as much as possible.

Fig. 8 provides a schematic representation of the proposed network. All layers except last one use PReLU as activation function and no batch normalization. The 7×7 convolutional layer can be considered skip connection and also learns the basic mapping from RGB input to HS jointly with the main subnetwork. The main subnetwork first shrinks the input then applies residual blocks with skip connections to then expand back to a 31-band output image added to the 7×7 conv subnet output. CVL used data augmentation [29] through scaling and flips/rotations at training and enhanced prediction [29] through flips/rotations at testing. The network was optimized for L2 loss. The larger was set the number of used residual blocks in the proposed FCN architecture, the better was the performance achieved but at the expense of slower inference time. More details are found in [6] where an efficient shallower design achieved top results on the common HS-from-RGB benchmarks: ICVL [4], CAVE [30], NUS [21].

4.7. adv_rgb2hs

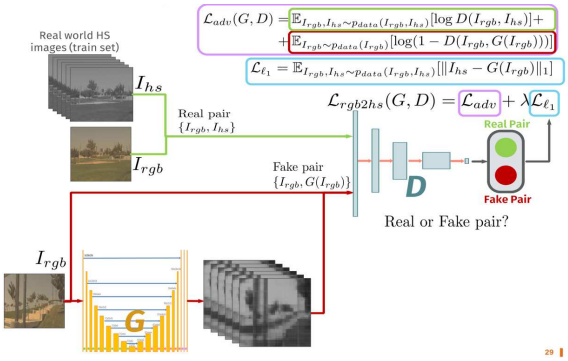


Figure 9. The adv_rgb2hs adversarial spatial context-aware spectral image reconstruction model.

Hyperspectral natural image reconstruction was posed as an image to image mapping learning problem, and a conditional generative adversarial framework (Fig. 9) was applied to help capture spatial semantics. In particular, [13] was adapted to this task, as described in [3]. Initially, the generator was defined as a modified U-net architecture, comprising eight successive 3×3 convolutions with stride 2 and a leaky ReLU after each of them and eight transposed

convolution blocks successively doubling the activation size up until the original 256×256 size, followed by two 1×1 convolutions in order to get the direct input images adequately combined with the upstream features. However, due to an empirical finding suggesting that increasing the receptive field over a certain value does not benefit reconstruction performance, a pruned version of the generator was used instead, with only three branches and a receptive field of 7×7. As for the discriminator, it was composed of five 3×3 convolutional layers with a stride a of 2. The reader is referred to [3] for further implementation details.

For this challenge, though, certain elements were modified as compared to [3]: MRAE was used as the non-adversarial component of the overall loss function, instead of L1. Training was performed from scratch over random 256×256 crops for 500 epochs, and the test-time reconstruction was done in a fully-convolutional way. The network was implemented using Pytorch, and a NVIDIA Titan X Pascal GPU was used for training and inference. Training required $\approx$ 16sec/epoch for a batch size of 1. The same approach was used for both Clean and Real World challenge branches, although separate models were trained for each of them.

4.8. CEERI [15]

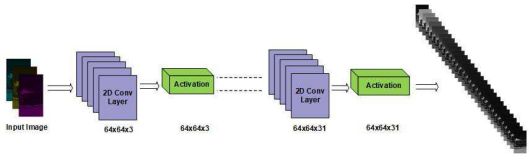


Figure 10. 2D-CNN model architecture used by the CEERI team.

A 2D convolution neural network based approach was used for hyperspectral image reconstruction from RGB. A 2D-CNN model primarily focuses on extracting spectral data by considering only spatial correlation of the channels in the image. The 2D-CNN model as shown in Figure 10, having 2D kernel extracts the hyperspectral information available in the spatial domain of the specific channel. The kernel convolves on individual channels (i.e. R, G, B) and the average of the values generated for each pixel of these channels are considered. The architecture is a simple 5 convolution layer architecture with 5×5 kernel size followed by ReLU activation. The four initial layers contain 64 feature maps and the final layer contains 31 feature maps which correspond to output images spectral channels. Training of the model was done on the BGU HS dataset. Patches of size $64 \times 64 \times 3$ were extracted from RGB input images and patches of size $64 \times 64 \times 31$ were extracted from corresponding hyperspectral images. The network was trained by feeding an RGB patch as input while the corresponding hyperspectral patch was used as ground-truth. A total of 84021 training patches were extracted from the provided data.

The 5-layer architecture was trained with learning rate of 10^{-4} using the Adam optimizer to minimize the mean absolute error between the model output and ground truth.

4.9. SPEC_RC

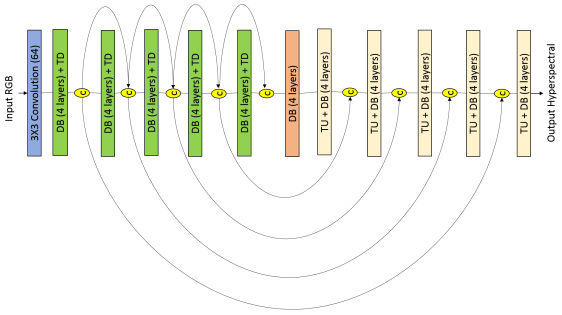


Figure 11. SPEC_RC's DenseNet architecture model. DB stands for Dense Block, TD for Transition Down,TU for Transition Up. The number of convolutional layers used in each block is shown within brackets.

Stacked layers of dense blocks were used for RGB to hyperspectral reconstruction. The DenseNet architectures have been used for segmentation and hyperspectral reconstruction previously [9]. In this implemetation each dense block consisted of four convolutional layers with a growth factor of 16. The model consisted of 56 convolution layers with 2,456,512 training parameters (Fig. 11). Six dense blocks containing [4,4,4,4,4,4] convolution layers are used for encoding. Each dense block has a growth rate of 16. Features were downsampled by a factor of 2 after each of the first five dense blocks using a Transition Down(TD) block where each TD block contains 1×1 convolution layer followed by a 2×2 maxpooling. The transition Up(TU) block consisted of two convolutional layers and a subpixel upsampling layer. The decoding part of the network consisted of 5 dense blocks each containing 4 convolution layers preceded by a TU block. The mean squared error between the predicted result and ground truth data was used as the loss function. The model was initialized with HeUniform[11] and trained with the Adam[14] optimizer (β_1=0.9, β_2=0.999 and $\epsilon = 10^{-8}$) for 300 epochs with a learning rate of 0.001 on mini batches of size 20.

Acknowledgements

We thank the NTIRE 2018 sponsors: Alibaba Group, NVIDIA Corp., SenseTime Group Ltd., Huawei Technologies Co. Ltd., Code Ocean, Google Inc., Disney Research, Amazon.com, Inc., and ETH Zurich. We also thank the Frankel Fund and the Helmsley Charitable Trust through the ABC Robotics Initiative, both at Ben-Gurion University of the Negev.

A. Teams and affiliations

NTIRE2018 team

Title: NTIRE 2018 Challenge on Spectral Reconstruction from RGB Images
Members: *Boaz Arad* [1]*(boazar@bgu.ac.il)*, Ohad Ben-Shahar [1], Radu Timofte [2,3], Luc Van Gool [2,4], Lei Zhang [5], Ming-Hsuan Yang [6]
Affiliations:
[1] Ben-Gurion University of the Negev, Israel
[2] Computer Vision Lab, ETH Zurich, Switzerland
[3] Merantix, Germany
[4] ESAT, KU Leuven, Belgium
[5] Polytechnic University of Hong Kong, China
[6] University of California at Merced, US

A.1. VIDAR team

Title: HSCNN+: Advanced CNN-Based Hyperspectral Recovery from RGB Images
Members: *Zhiwei Xiong (zwxiong@ustc.edu.cn)*, Chang Chen, Zhan Shi, Dong Liu, Feng Wu
Affiliations:
University of Science and Technology of China

A.2. HypedPhoti team

Title: Spectral Reconstruction from RGB with Fully Convolutional ResNet
Members: *Charis Lanaras (charis.lanaras@geod.baug.ethz.ch)*, Silvano Galliani, Konrad Schindler
Affiliations:
Photogrammetry and Remote Sensing, ETH Zurich, Switzerland

A.3. LFB team

Title: Reconstructing Spectral Images from RGB-images using a Convolutional Neural Network
Members: *Tarek Stiebel (Tarek.Stiebel@lfb.rwth-aachen.de)*, Simon Koppers, Philipp Seltsam
Affiliations:
Institute of Imaging and Computer Vision, RWTH Aachen University, Germany

A.4. IVRL Prime team

Title: Generative Adversarial Residual Network for Hyperspectral Reconstruction
Members: *Ruofan Zhou (ruofan.zhou@epfl.ch)*, Majed El Helou, Fayez Lahoud, Marjan Shahpaski
Affiliations:
Image and Visual Representation Lab, EPFL, Switzerland

A.5. sr402 team

Title: Residual Dense Concatenate SE Network for Spectral Reconstruction
Members: *Ke Zheng (zhengkevic@gmail.com)*[1,2], Lianru Gao[1], Bing Zhang[1], Ximin Cui[2], Haoyang Yu[1]
Affiliations:
[1] Key Laboratory of Digital Earth Science, Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, Beijing 100094, China
[2] College of Geoscience and Surveying Engineering, China University of Mining and Technology (Beijing), Beijing 100083, China

A.6. CVL team

Title: An efficient CNN for spectral reconstruction from RGB images
Members: *Yigit Baran Can (ybarancan@gmail.com)*, Radu Timofte
Affiliations:
Computer Vision Lab, ETH Zurich, Switzerland

A.7. adv_rgb2hs team

Title: Adversarial Networks for Spatial Context-Aware Spectral Image Reconstruction from RGB
Members: *Aitor Alvarez-Gila (aitor.alvarez@tecnalia.com)*[1,2], Joost van de Weijer[2], Estibaliz Garrote[1], Adrian Galdran[3]
Affiliations:
[1] Tecnalia, Spain
[2] Computer Vision Center/Universitat Autonoma de Barcelona, Spain
[3] INESC-TEC Porto, Portugal

A.8. CEERI team

Title: RGB to Hyperspectral conversion using deep convolutional neural network
Members: *Manoj Sharma (sriharsharaja@gmail.com)*, Sriharsha Koundinya, Avinash Upadhyay, Raunak Manekar, Rudrabha Mukhopadhyay, Himanshu Sharma, Santanu Chaudhury
Affiliations: CSIR-CEERI

A.9. SPEC_RC team

Members: *Koushik Nagasubramanian (koushikn@iastate.edu)*, Sambuddha Ghosal, Asheesh K. Singh, Arti Singh, Baskar Ganapathysubramanian, Soumik Sarkar
Affiliations: Iowa State University, US

References

[1] J. Aeschbacher, J. Wu, and R. Timofte. In defense of shallow learned spectral reconstruction from rgb images. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017. 2, 3

[2] F. Agahian, S. A. Amirshahi, and S. H. Amirshahi. Reconstruction of reflectance spectra using weighted principal component analysis. *Color Research & Application*, 33(5):360–371, 2008. 1

[3] A. Alvarez-Gila, J. v. d. Weijer, and E. Garrote. Adversarial Networks for Spatial Context-Aware Spectral Image Reconstruction from RGB. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 480–490, Oct. 2017. 2, 7, 8

[4] B. Arad and O. Ben-Shahar. Sparse recovery of hyperspectral signal from natural rgb images. In *European Conference on Computer Vision*, pages 19–34. Springer, 2016. 1, 2, 3, 7

[5] B. Arad and O. Ben-Shahar. Filter selection for hyperspectral estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3153–3161, 2017. 1

[6] Y. B. Can and R. Timofte. An efficient CNN for spectral reconstruction from RGB images. In *arXiv:1804.04647*, March 2018. 3, 4, 7

[7] A. Chakrabarti and T. Zickler. Statistics of real-world hyperspectral images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 193–200. IEEE, 2011. 2

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

[9] S. Galliani, C. Lanaras, D. Marmanis, E. Baltsavias, and K. Schindler. Learned spectral super-resolution. *arXiv preprint arXiv:1703.09470*, 2017. 8

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative Adversarial Networks. *CoRR*, 2014. 3, 6

[11] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 8

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 6

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017. 7

[14] Kingma, Diederik P and Ba, Jimmy. Adam: A Method for Stochastic Optimization. *arXiv.org*, Dec. 2014. 6, 8

[15] S. Koundinya. 2d-3d cnn based architectures for spectral reconstruction from rgb images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 3, 4, 8

[16] H. Kwon and Y.-W. Tai. Rgb-guided hyperspectral image upsampling. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 307–315. IEEE, 2015. 1

[17] A. Lambrechts, P. Gonzalez, B. Geelen, P. Soussan, K. Tack, and M. Jayapala. A cmos-compatible, integrated approach to hyper-and multispectral imaging. In *Electron Devices Meeting (IEDM), 2014 IEEE International*, pages 10–5. IEEE, 2014. 1

[18] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. 3

[19] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017. 5

[20] L. T. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *JOSA A*, 3(10):1673–1683, 1986. 1

[21] R. M. Nguyen, D. K. Prasad, and M. S. Brown. Training-based spectral reconstruction from a single rgb image. In *European Conference on Computer Vision*, pages 186–201. Springer, 2014. 2, 7

[22] T. Okamoto and I. Yamaguchi. Simultaneous acquisition of spectral image information. *Optics letters*, 16(16):1277–1279, 1991. 1

[23] M. Parmar, S. Lansel, and B. A. Wandell. Spatio-spectral reconstruction of the multispectral datacube using sparse recovery. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 473–476. IEEE, 2008. 1, 2

[24] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR*, cs.LG, 2015. 6

[25] A. Robles-Kelly. Single image spectral reconstruction for multimedia applications. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 251–260. ACM, 2015. 2

[26] Z. Shi, C. Chen, Z. Xiong, D. Liu, and F. Wu. Hscnn+: Advanced cnn-based hyperspectral recovery from rgb images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018. 3, 4, 5

[27] T. Stiebel, S. Koppers, P. Seltsam, and D. Merhof. Reconstructing spectral images from rgb-images using a convolutional neural network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 3, 4, 6

[28] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision*, pages 111–126. Springer, 2014. 2

[29] R. Timofte, R. Rothe, and L. Van Gool. Seven ways to improve example-based single image super resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3, 7

[30] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE transactions on image processing*, 19(9):2241–2253, 2010. 1, 2, 7