CUBE IT: TRAINING HYPERSPECTRAL DEMOSAICING MODELS USING SYNTHETIC DATASETS

Rafał Muszyński, Hiep Luong

IPI-URC-imec, Ghent University, Ghent, Belgium

ABSTRACT

Hyperspectral demosaicing aims to recover full spectral information at each pixel from a mosaicked image captured using a snapshot camera. Cameras vary in terms of used multispectral filter arrays (MSFA). SOTA demosaicing algorithms are evaluated and trained on a handful of publicly available datasets, and their performance does not transfer well to images captured with previously unseen cameras. Performing demosaicing for a specific MSFA requires training a new model, which is time-consuming and may require capturing new datasets, which hinders usability of SOTA models. We demonstrate, that demosaicing models with near SOTA performance can be trained using existing RGB datasets with simple hyperspectral augmentations. By performing random band reordering in the MSFA during training, our models seamlesly work with different MSFA. Conducted experiments show good quantitive and qualitative results. Our code will be made publicly available at `github.com/ramusz1/cube-it`.

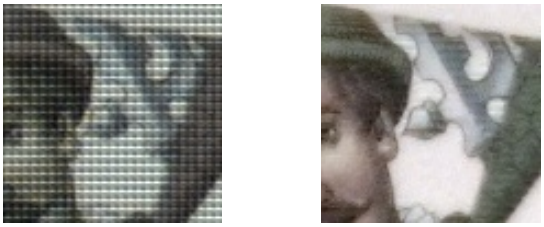
***Index Terms*—** Hyperspectral imaging, demosaicing, synthetic data

1. INTRODUCTION

Hyperspectral cameras capture rich information about the observed scene. This additional spectral information allows for the noninvasive distinction of chemicals and materials.

Current snapshot mosaic sensors produced by IMEC allow for hyperspectral videos with real time frame rate. This form factor leads to many applications such as: hyperspectral object tracking [1], corrosion detection [2], medicine [3], and many more. Snapshot cameras incorporate a MSFA to capture a specific wavelength at a given pixel. Similarly to RGB cameras, full per pixel spectral information can be reconstructed by performing demosaicing.

Recently, multispectral demosaicing has seen a lot of progress due to the NTIRE 2022 Spectral Demosaicing Challenge [4] which provided the largest publicly available dataset up to date. In practice, different cameras utilize different MSFA, with different bands and band layouts. This poses a practical problem, where many demosaicing algorithms do not perform well for cameras different than used for the training set (Figure 1). Therefore, currently, to perform demosaicing for a new MSFA, a new demosaicing model has to be retrained, which usually requires preparation of a new sensor specific dataset. On the other hand, with around 1000 images in the NTIRE dataset, we speculate that the models do not have sufficient training data to generalize well to unknown conditions observed in practice.



(a) Trained on NTIRE dataset (b) Trained on our synthetic data

Fig. 1: Models trained on our synthetic dataset are more flexible and achieve better results on our real data than models trained using conventional approaches.

In this paper, we hypothesize that demosaicing algorithms do not require physically accurate data, and can be instead trained on synthetic data generated from large scale RGB datasets. Our randomized dataset generation method, allowed us to train a robust demosaicing model which generalizes to unseen during training, real life sensors, and achieves good performance on standard benchmarks.

2. RELATED WORK

Table 1 lists selected publicly available datasets. Notably, the NTIRE 2022 Spectral Demosaicing Challenge [4] introduced a synthetic 16 band multispectral dataset for developing 4x4 demosaicing algorithms. With 950 available spectral cubes, this is currently the largest dataset. Images were calculated by simulating image capture using a certain 16 channel sensor prototype. That is, images contain the information about radiance. On the other hand, CAVE [5] and Tokyo Tech [6, 7] datasets contain reflectance information, which is a physical property of an object, and paired with arbitrary illumination can be used to simulate radiance. This leads to additional dataset augmentation techniques, such as applying different

Dataset	#Images	#Bands	Spectral range (nm)
NTIRE 2022 Spectral Demosaicing [4]	950	16	400-1000
NTIRE 2022 Spectral Recovery [9]	950	31	400-700
CAVE [5]	32	31	400-700
TokyoTech 31 [6]	30	31	420-720
TokyoTech 59 [7]	16	59	420-1000

Table 1: Most popular publicly available for hyperspectral demosaicing

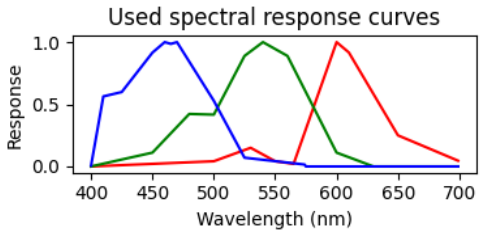


Fig. 2: Spectral response curves used for our RGB to hyperspectral conversion.

illumination to a training sample [8].

Multiple network architectures have been proposed for the task of multispectral demosaicing. MCAN [8] introduced mosaic convolutions, which adapt to the underlying MSFA, as well as mosaic attention module which utilize position sensitive feature aggregation. DAMCNet, winner of the NTIRE 2022 Spectral Demosaicing Challenge [4] is based upon channel attention modules. An example of a convolutional based architecture is [10], which uses ResNet [11] based network to refine results of bilinear interpolation.

Hyperspectral dataset synthesis has already been proposed, for example in [12], to train a hyperspectral object tracker using available RGB data converted to hyperspectral. We believe the proposed approach does not generate challenging cases for the demosaicing.

Hyperspectral information from RGB data is a currently researched topic. The winner of the NTIRE 2022 Spectral Recovery Challenge [9] - MST++ [13] utilizes spectral correlations and self similarity using a developed spectral-wise attention blocks. Alternatively in [14] spectral information is recovered based on a database of observed spectra.

Our work is not the first, to tackle problems with existing demosaicing methodologies and datasets. [15] proposes training a demosaicing network in an unsupervised manner. In [16] authors focus on practical solutions for problems encountered in real applications of demosaicing.

3. MULTISPECTRAL DATASET SYNTHESIS

In this section we describe tested approaches for hyperspectral dataset synthesis.

In hyperspectral images, the neighbouring bands are correlated and their relationship depends on captured material. The second part is especially important, since it implies that the band relationship changes at object boundaries. Those characteristics should be enabled by our synthesis framework.

We generate hyperspectral images based on a linear combination of RGB channels. Inspired by the reverse process of obtaining RGB images from hyperspectral data, we use per-channel response functions: S_r, S_g, S_b, for channels R,G and B respectively. For a pixel at position i, j in the input RGB image, we calculate the corresponding value of hyperspectral cube for wavelength l as: $R_{i,j}S_r(l) + G_{i,j}S_g(l) + B_{i,j}S_b(l)$. For our experiments, we approximated the spectral response of a Nikon D70 camera as reported in [17] (Figure 2). For each training sample. we randomly sample 16 wavelengths from interval 400-700nm.

As a more realistic approach, we generate hyperspectral images from RGB images using the MST++ approach. This approach should extend all of the priors from the NTIRE dataset into a much larger dataset.

In the NTIRE dataset, a specific MSFA was used to generate fake RAW images for the training dataset. Since the NTIRE dataset contains full spectral cubes, we can generate different RAW images for the training dataset, by mixing the order of bands in the MSFA. This is the simplest approach to generate a demosaicing dataset that does not depend on a specific mosaic pattern.

4. DEMOSAICING MODEL ARCHITECTURE

In this section, we describe the network architecture used in our experiments. In recent years, demosaicing models have become more and more complex. Since one of the advantages of snapshot cameras is high frame rate, we believe that more attention should be brought to processing speeds and hardware requirements. Therefore, similarly as [10], we propose ResD - a small and straightforward architecture, based on ESRGAN [18] and Resnet [11]. Fig. 3 presents our model.

To provide the convolutional layers with information about the position in the mosaic pattern and captured band's position in the output cube, we concatenate positional encoding with the input. For positional encoding we use a 3 channel, 4x4 pattern P repeated over the whole input. The value at position i, j of P is $P_{i,j} = (\frac{i}{4}, \frac{j}{4}, \frac{4i+j}{16})$.

5. EXPERIMENTS

In this section we describe used datasets, augmentation techniques, and training procedures used in our experiments.

Since the NTIRE 2022 demosaicing and spectral reconstruction datasets are based on the same scenes, we have ac-

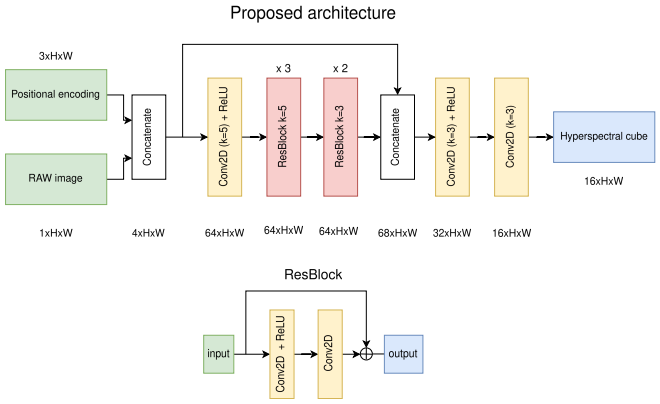


Fig. 3: Proposed demosaicing architecture of our ResD model

cess to the following data: 950 scenes (900 training, 50 test), with 16 band spectral HSI (NTIRE_{16}), 31 band spectral HSI (NTIRE_{31}) and RGB images of the captured scene. Additionally, we use COCO dataset [19] to generate larger hyperspectral datasets using our technique.

Our target dataset is a small sample of real images captured using IMEC snapshot VIS and NIR cameras.

Using MSFA properties of the IMEC VIS camera and selected illuminants, we transform reflectance data from the CAVE dataset into simulated cubes captured using our tested camera ($\text{CAVE}_{\text{imec}}$ dataset). This dataset should be the closest to our target dataset.

Using RGB images, we synthesize a fake HSI dataset using the following approaches:

- Using our method, we synthesize HSI cubes for each training batch (Synth_{1k},Synth_{10k},Synth_{100k})
- Recover HSI cubes using the MST++ model. MST++ outputs cubes with 31 bands. Because this process is time-consuming (around 1s for a 480x512 RGB image), we generate 10k cubes once, before the training (MST++_{10k}).

For each training sample, we performed augmentations such as random rotations and crops. For cubes in the NTIRE_{31} and MST++_{10k} we randomly select 16 bands. By randomly reordering bands in the cubes from the NTIRE_{16} and $\text{CAVE}_{\text{imec}}$, we generate new datasets: $\text{NTIRE}_{16-mixed}$ and $\text{CAVE}_{\text{imec}-mixed}$ dataset. For datasets with randomized band order, we select bands per each sample during training, but fix the order for the elements in the validation set.

Depending on the dataset size: 1k, 10k, 100k, we train our model for 1000, 100 and 20 epochs respectively, using Adam optimizer, with initial learning rate = 0.001 and decreased during learning. We use a smooth L1 loss function with $\beta = 0.001$. For the MCAN model, we adapted the original training procedure to different dataset size and added gradient clipping.

6. RESULTS

Table 2 presents results of our experiments. Since ground truth is not available for the IMEC dataset, we measure mean squared reprojection error and reprojection error range (minimal and maximal error). Reprojection error is the difference between input RAW pixels and the corresponding pixels in the output cube. Models trained on our synthetic datasets produce competitive results on the NTIRE_{16} dataset, while maintaining quality on the IMEC and CAVE data. MCAN architecture achieves the best results in almost all tests, apart from the $\text{CAVE}_{\text{imec}}$ dataset. While ResD produces worse results than MCAN, the test metrics are still very high.

Figure 4 presents graphical results on our custom dataset. Our model trained on the standard NTIRE_{16} doesn't work on our custom dataset. Mixing the bands during training ($\text{NTIRE}_{16-mixed}$) leads to great visual results. Models trained on the Synth datasets produce the sharpest results, but sometimes the mosaic pattern can become visible (cases a, c, e). Training ResD on the MST++_{10k} produces smoother results than using our synthetic dataset, but overall similar in quality, with cases a, b looking better, and c, d worse. In most cases, the MCAN model performs much worse for our dataset than when training on the NTIRE dataset. An exception is example d, where the model produces sharper results without producing artefacts.

Finally, we measure inference time for a 256×256 patch and a full resolution real image for both architectures (Table 3). Because our model requires less memory during inference, it fits on our graphics card, which allows for the fastest processing for real life demosaicing task.

7. CONCLUSIONS

In this paper, we show that demosaicing models with near SOTA performance can be trained using existing RGB datasets with simple hyperspectral augmentations. Time-intensive full spectral reconstruction is not needed. Moreover, performing random band reordering in the MSFA during training, increases the model's robustness for new MSFA. We combine our training methodology with a lightweight model, which is suitable for processing high resolution images. Our approach is suitable for researching new MSFA configurations and increasing the value of already existing hyperspectral data, which currently available SOTA demosaicing models struggle with.

8. REFERENCES

[1] Rafał Muszyński and Hiep Luong, "Helios: Hyperspectral hindsight ostracker," in *2023 13th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2023, pp. 1–5, ISSN: 2158-6276.

Model name	Training dataset	$\text{NTIRE}_{16-\text{validation}}$ PSNR	$\text{CAVE}_{\text{imec}-mixed}$ PSNR	$\text{CAVE}_{\text{imec}}$ PSNR	IMEC reprojection error	IMEC error range
ResD	NTIRE_{16}	45.1	24.1	25.8	1940.4	[-502.0, 649.0]
ResD	NTIRE_{31}	40.6	42.1	41.0	12.6	[-187.2, 233.7]
ResD	Synth_{1k}	40.2	43.0	41.9	10.0	[-120.8, 213.8]
ResD	Synth_{10k}	41.1	43.3	42.4	10.8	[-125.0, 144.1]
ResD	Synth_{100k}	42.1	44.4	**43.3**	6.7	[-206.4 96.0]
ResD	$\text{NTIRE}_{16-mixed}$	40.7	41.7	40.5	18.14	[-181.1, 275.5]
ResD	MST++_{10k}	40.0	41.8	39.9	18.3	[-163.2, 239.3]
MCAN	NTIRE_{16}	**48.4**	24.4	28.7	850.1	[-723.2, 592.7]
MCAN	$\text{NTIRE}_{16-mixed}$	43.8	43.8	42.3	2.6	[-106.4, 124.9]
MCAN	Synth_{100k}	43.0	**44.8**	42.1	**1.7**	**[-62.8, 95.1]**

Table 2: Comparing performance of models trained on different datasets. Our approach produces more flexible models with similar quality on public datasets to models trained on those datasets. For training on the NTIRE datasets, the training split is used.

Model	Device	Input resolution	Inference Time Mean	Inference Time STD
MCAN	CPU	256×256	0.002	0.003
		1024×2048	3.370	0.073
	GPU	256×256	0.002	0.003
		1024×2048	OOM	OOM
ResD	CPU	256×256	0.142	0.011
		1024×2048	5.202	0.124
	GPU	256×256	0.001	0.000
		1024×2048	0.520	0.005

Table 3: Comparison of inference speed of tested models. We were not able to process the full image using our GPU due to out of memory (OOM) error. ResD requires less memory, which enables GPU processing of higher resolution images (captured using imec cameras) using our graphics card. Used hardware: NVIDIA GeForce RTX 3080 GPU and 11th Gen Intel(R) Core(TM) i7-11700KF @ 3.60GHz CPU

[2] F. Mangialetto et al., "Hyperspectral imaging for the corrosion detection on metallic lattice towers," in *2024 CIGRE SESSION*. 2024, CIGRE.

[3] Jens De Winne, Anoek Strumane, Danilo Babin, Siri Luthman, Hiep Luong, and Wilfried Philips, "Multispectral indices for real-time and non-invasive tissue ischemia monitoring using snapshot cameras," *BIOMEDICAL OPTICS EXPRESS*, vol. 15, no. 2, pp. 641–655, 2024.

[4] B. Arad et al., "NTIRE 2022 spectral demosaicing challenge and data set," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2022, pp. 881–895, IEEE.

[5] Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K Nayar, "Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum," *IEEE Trans. on Image Process.*, vol. 19, no. 9, pp. 2241–2253, 2010.

[6] Yusuke Monno, Sunao Kikuchi, Masayuki Tanaka, and Masatoshi Okutomi, "A practical one-shot multispectral imaging system using a single image sensor," *IEEE Trans. on Image Process.*, vol. 24, no. 10, pp. 3048–3059, 2015.

[7] Yusuke Monno, Hayato Teranaka, Kazunori Yoshizaki, Masayuki Tanaka, and Masatoshi Okutomi, "Single-sensor RGB-NIR imaging: High-quality system design and prototype implementation," *IEEE Sensors J.*, vol. 19, no. 2, pp. 497–507, 2019.

[8] Kai Feng, Yongqiang Zhao, Jonathan C-W Chan, Seong Kong, Xun Zhang, and Binglu Wang, "Mosaic convolution-attention network for demosaicing multispectral filter array images," *IEEE Trans. Comput. Imaging*, vol. 7, pp. 864–878, 2021.

[9] B. Arad et al., "NTIRE 2022 spectral recovery challenge and data set," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2022, pp. 862–880, IEEE.

[10] Kazuma Shinoda, Shoichiro Yoshiba, and Madoka Hasegawa, "Deep demosaicking for multispectral filter arrays," 2018, arXiv:1808.08021.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778, IEEE.

[12] Z. Li et al., "Rawtrack: Toward single object tracking on mosaic hyperspectral raw data," in *2023 13th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 2023, pp. 1–5.

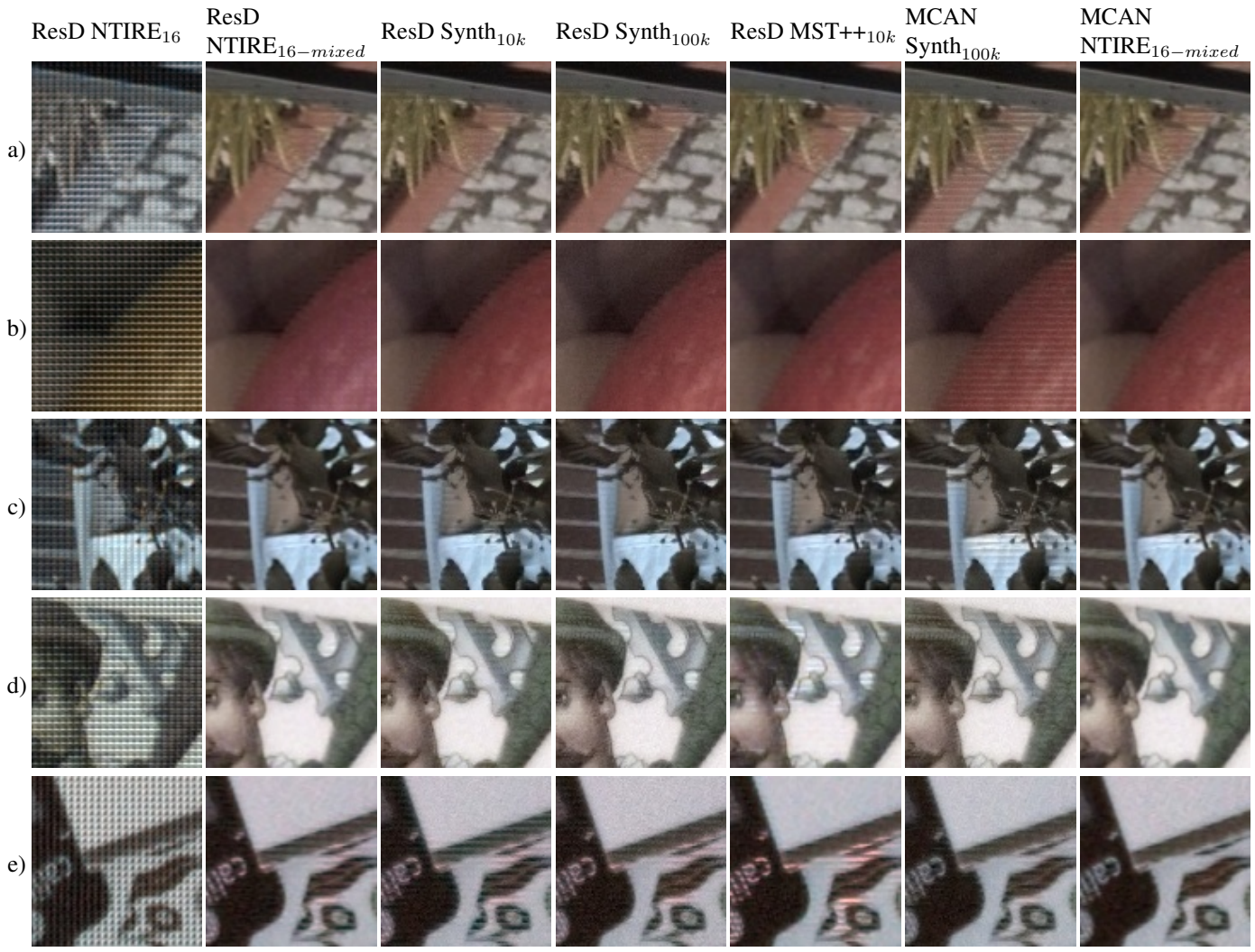


Fig. 4: Visual inspection on our target dataset. Our model ResD trained on different datasets, MCAN

[13] Y. Cai et al., "MST++: Multi-stage spectral-wise transformer for efficient spectral reconstruction," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2022, pp. 744–754, IEEE.

[14] Boaz Arad and Ohad Ben-Shahar, "Sparse recovery of hyperspectral signal from natural rgb images," in *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, Eds., Cham, 2016, pp. 19–34, Springer International Publishing.

[15] Kai Feng, Haijin Zeng, Yongqiang Zhao, Seong G. Kong, and Yuanyang Bu, "Unsupervised spectral demosaicing with lightweight spectral attention networks," *IEEE Transactions on Image Processing*, vol. 33, pp. 1655–1669, 2024.

[16] Eric L. Wisotzky, Lara Wallburg, Anna Hilsmann, Peter Eisert, Thomas Wittenberg, and Stephan Göb, "Efficient and accurate hyperspectral image demosaicing with neural network architectures," 2023, arXiv:2403.12050.

[17] Nikolay Petrov, Victor Bespalov, and Andrei Gorodetsky, "Phase retrieval method for multiple wavelength speckle patterns," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 7387, 2010.

[18] X. Wang et al., "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Computer Vision – ECCV 2018 Workshops*, Laura Leal-Taixé and Stefan Roth, Eds., vol. 11133, pp. 63–79. Springer International Publishing, 2019, Series Title: Lecture Notes in Computer Science.

[19] T. Lin et al., "Microsoft COCO: Common objects in context," 2015, arXiv:1405.0312.