Peer Reviewed Paper openaccess Paper Presented at IASIM 2022, July 2022, Esbjerg, Denmark

The hybrid approach—convolutional neural networks and expectation maximisation algorithm—for tomographic reconstruction of hyperspectral images

Mads Juul Ahlebæk,[a,*] Mads Svanborg Peters,[b,c] Wei-Chih Huang,[a] Mads Toudal Frandsen,[a] René Lynge Eriksen[c] and Bjarke Jørgensen[b]

[a]CP^3-Origins, Department of Physics, Chemistry and Pharmacy, University of Southern Denmark, Denmark

[b]Newtec Engineering A/S, 5230 Odense, Denmark

[c]Mads Clausen Institute, University of Southern Denmark, Denmark

Contact
M.J. Ahlebæk: ahle@sdu.dk
https://orcid.org/0000-0003-4938-4802
M.S. Peters: mape@newtec.dk
https://orcid.org/0000-0002-5761-4586
W.-C. Huang: wchi@newtec.dk
https://orcid.org/0000-0001-7939-3246
M.T. Frandsen: frandsen@cp3.sdu.dk
https://orcid.org/0000-0003-2061-562X
R.L. Eriksen: rle@mci.sdu.dk
https://orcid.org/0000-0003-4405-5831
B. Jørgensen: bjarke@newtec.dk
https://orcid.org/0000-0001-8559-9435

We present a simple, but novel, hybrid approach to hyperspectral data cube reconstruction from computed tomography imaging spectrometry (CTIS) images that sequentially combines neural networks and the iterative expectation maximisation (EM) algorithm. We train and test the ability of the method to reconstruct data cubes of $100 \times 100 \times 25$ and $100 \times 100 \times 100$ voxels, corresponding to 25 and 100 spectral channels, from simulated CTIS images generated by our CTIS simulator. The hybrid approach utilises the inherent strength of the Convolutional Neural Network (CNN) with regards to noise and its ability to yield consistent reconstructions and make use of the EM algorithm's ability to generalise to spectral images of any object without *training*. The hybrid approach achieves better performance than both the CNNs and EM alone for seen (included in CNN training) and unseen (excluded from CNN training) cubes for both the 25- and 100-channel cases. For the 25 spectral channels, the improvements from CNN to the hybrid model (CNN + EM) in terms of the mean-squared errors are between 14 % and 26 %. For 100 spectral channels, the improvements between 19 % and 40 % are attained with the largest improvement of 40 % for the unseen data, to which the CNNs are not exposed during the training.

Keywords: snapshot, hyperspectral imaging, artificial neural networks, convolutional neural networks, tomographic reconstruction

Correspondence
M.J. Ahlebæk: ahle@sdu.dk

Received: 1 November 2022
Revised: 8 December
Accepted: 9 December 2022
Publication: 31 January 2023
doi: 10.1255/jsi.2023.a1
ISSN: 2040-4565

Citation
M.J. Ahlebæk, M.S. Peters, W.-C. Huang, M.T. Frandsen, R.L. Eriksen and B. Jørgensen, "The hybrid approach—convolutional neural networks and expectation maximisation algorithm—for tomographic reconstruction of hyperspectral images", *J. Spectral Imaging* **12,** a1 (2023).
https://doi.org/10.1255/jsi.2023.a1

Introduction

Multispectral (MSI) and hyperspectral imaging (HSI)[1] are used in a wide range of applications in diverse fields. These include astronomy and space surveillance,[2] spectroscopic differentiation of materials in geoscience,[3] detection of foreign objects and weeds in precision agriculture[4] and optical sorting within the food industry.[5] HSI produces 3-dimensional (3-D) data cubes that capture light intensities in two spatial and one spectral dimension. Pushbroom (line scan)[6] HSI is the standard technique, but it requires steady movement of either the object or camera to acquire a hyperspectral image. Also, the equipment cost is typically high, and this creates barriers to broader applications of HSI.

On the other hand, the Computed Tomography Imaging Spectrometer (CTIS)[7–9] is a relatively simple and potentially compact and cheap snapshot HSI system which can capture an image within milliseconds or an even shorter time. There exist alternative snapshot spectral imaging technologies that capture (projections of) the 3-D data cube instantaneously using dispersive optics such as single-shot compressive spectral imaging with dual-disperser architecture (CASSI),[10] Hybrid camera Multispectral-Video Imaging System (HMVIS),[11] lenslet-array,[12] filter-on-chip imagers,[13] Image Mapping Spectrometers (IMS),[14] Image-replicating Imaging Spectrometers[15] and snapshot HSI Fourier transform spectrometers.[16] Nonetheless, the CTIS is investigated here.

The CTIS system acquires a 2-D image $\boldsymbol{g}$ by means of a diffractive optical element (DOE) that diffracts the 3-D hyperspectral cube $\boldsymbol{f}$ into the zeroth and surrounding first orders (Figure 1), corresponding to projections of the cube onto a 2-D plane. The 2-D projection is determined by the system matrix, denoted by $\boldsymbol{H}$ ($\boldsymbol{g} = \boldsymbol{Hf}$), which incorporates the optical parameters of the CTIS system as we shall discuss below.

For a captured CTIS image $\boldsymbol{g}$, one can either directly analyse it, e.g., for classification of apple scab lesions[17,18] or reconstruct a 3-D cube $\boldsymbol{f}$ from $\boldsymbol{g}$, which leads to wider applications, with the help of the inverse matrix $\boldsymbol{H}^{-1}$. However, $\boldsymbol{H}$ is a sparse, enormous and (usually) rectangular matrix, which makes the computation of the Moore–Penrose pseudo-inverse impractical in terms of both computation time and memory consumption. As a result, iterative algorithms[19–21] have been proposed to reconstruct $\boldsymbol{f}$. The reconstruction time is unfortunately quite long, and the accuracy is mediocre, especially for large images or high numbers of spectral channels, which hinders practical applications. Fast and precise real-time reconstruction is, therefore, an important but challenging goal.

In Reference 22, we, for the first time, applied convolutional neural networks (CNN)[23,24] to reconstruct the data cube from a simulated CTIS image. By comparison, we also used the expectation maximisation (EM) algorithm for cube reconstruction. The EM takes as input arguments $\boldsymbol{H}$, a CTIS image $\boldsymbol{g}$ and an initial guess of the reconstructed 3-D cube, and iteratively updated the cube until it approximately reproduces the true cube. Overall, the CNN performance is much better with a shorter reconstruction time than the EM. Moreover, the network can handle images of different objects, yielding a consistent accuracy, whereas the EM is challenged with objects of complex geometry (complex geometries having high spatial frequencies and varied spectral information). The EM algorithm is also susceptible to inherent noise in the CTIS images and fails to converge for noisy CTIS images. That is, the difference between the true and EM-reconstructed cube increases with more iterations.[25–27] An important caveat is that the CNN must see similar objects or geometries in the training phase to make reliable reconstructions. In other words, it cannot manage objects very different from those in the training data, whereas the EM applies to images of any objects.

In this work, we propose a simple but novel hybrid method that sequentially combines the neural networks and EM to circumvent the aforementioned shortcomings intrinsic to the EM algorithm and networks, respectively. Thus, a network is first employed to reconstruct a data cube from a CTIS image and then its output, as an initial guess of the data cube, is passed to the EM algorithm, which further improves the network predictability as depicted in Figure 1. In other words, the network provides a refined initial condition for the EM algorithm that is close to the correct data cube to guarantee convergence. Furthermore, the existence of the EM algorithm as the second step ensures that the hybrid model can be applied to new types of images even if those are not included in the training data of the network.

Introduction to CTIS imaging system and procedure of data generation

In this section, we introduce the CTIS imaging system, discuss the updated CTIS simulator and detail the preparation of data for 25 and 100 spectral channels used in the training and testing of the networks.

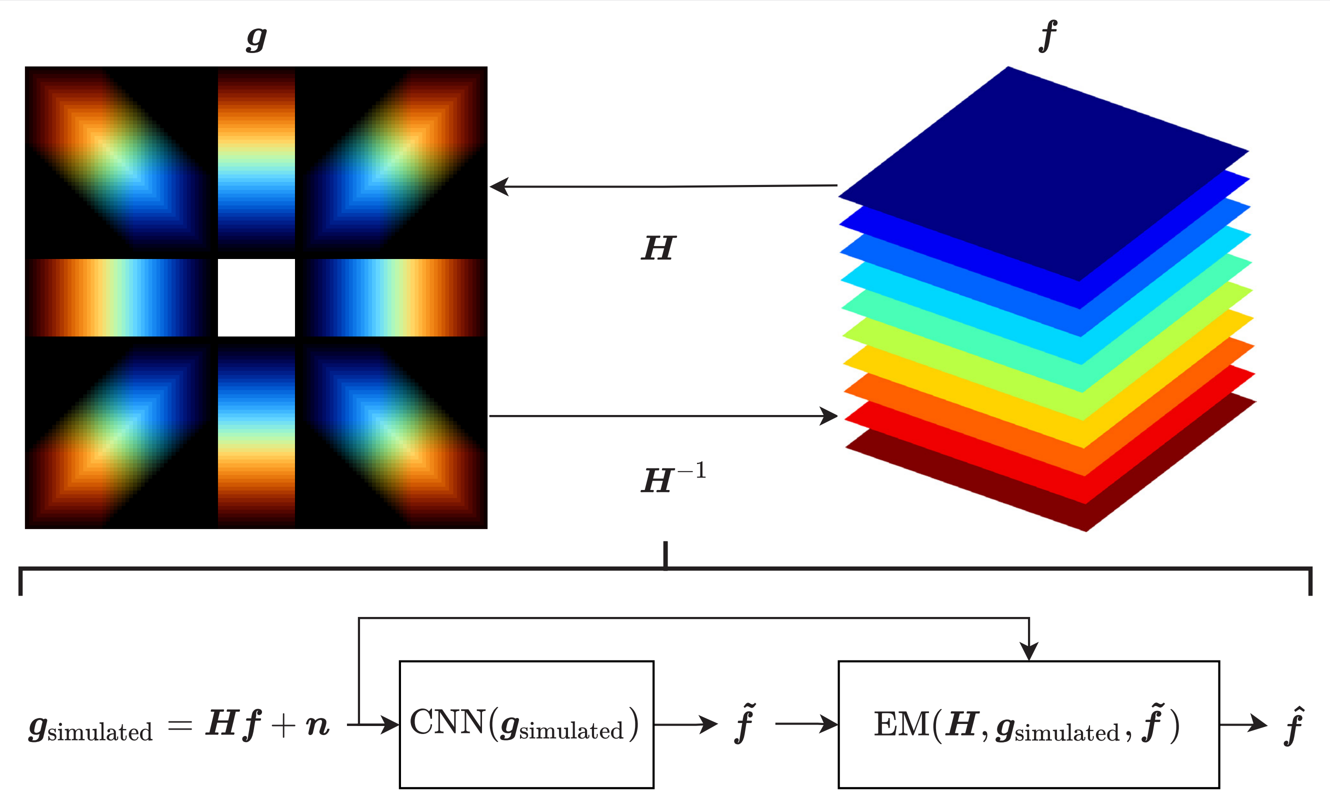
$\boldsymbol{g}$
$\boldsymbol{f}$
$\boldsymbol{H}$
$\boldsymbol{H}^{-1}$
$\boldsymbol{g}_{\text{simulated}} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{n}$
$\text{CNN}(\boldsymbol{g}_{\text{simulated}})$
$\tilde{\boldsymbol{f}}$
$\text{EM}(\boldsymbol{H}, \boldsymbol{g}_{\text{simulated}}, \tilde{\boldsymbol{f}})$
$\hat{\boldsymbol{f}}$

Figure 1. Top: illustration of the connection between a hyperspectral cube f and the corresponding CTIS image g via the H matrix. The image g consists of a central zeroth diffraction order and eight surrounding first diffraction orders, which are projections of the hyperspectral cube f. Bottom: the workflow of the hybrid approach, where $\tilde{f}$ and $\hat{f}$ are estimates of if reconstructed with the CNN and hybrid (CNN + EM), respectively; see explanations in the main text.

CTIS imaging system

The CTIS imaging system can be described by the linear imaging equation:[9]

$$g = Hf + n \tag{1}$$

where $\boldsymbol{g}$ is a vectorised CTIS image of $(q \times q = q^2)$ elements and $\boldsymbol{f}$ is the vectorised hyperspectral cube with $r = x \cdot y \cdot z$ voxels, where x, y, z denote the two spatial dimensions and the number of spectral channels, respectively, while n corresponds to a random noise vector. The $q^2 \times r$ system matrix $\boldsymbol{H}$ describes the projection of the i-th voxel in $\boldsymbol{f}$ to the j-th pixels in $\boldsymbol{g}$—equivalent to the nine projections in Figure 1, which consist of a central zeroth order, surrounded by eight first orders.

The system matrix $\boldsymbol{H}$ is constructed assuming spatial shift-invariance and a linear mapping between $\boldsymbol{f}$ and $\boldsymbol{g}$. It includes the point spread function (PSF), the illumination (wavelength-dependent intensity) and the diffraction sensitivity (which includes diffraction efficiency of the DOE, the transmission of the optical system and the sensor response). Both the illumination and diffraction sensitivity are wavelength dependent, while the PSF is assumed wavelength independent, in our limited wavelength range and with the used optics. Additionally, the diffraction sensitivity depends on the respective zeroth or first orders. See Supplementary Material Sections S3–S5 for additional details on determining system parameters. Thus, the i-th voxel, $\boldsymbol{f}_i$, is mapped into each diffraction order with a sensitivity given by the product of the diffraction sensitivity and the illumination for a specific wavelength and diffraction order. The resulting $q \times q$ CTIS image is convolved with the PSF, vectorised and arranged as columns in $\boldsymbol{H}$ for $i = 1,2, \ldots, r$. Due to the sparsity of $\boldsymbol{H}$, the memory requirements are large for large data cube dimensions ($x,y \geq 100$ and $z \geq 25$). Since a significant amount of CTIS images are needed for the training of the neural networks, a CTIS simulator was used to generate images without resorting to a system matrix $\boldsymbol{H}$.

Simulating CTIS images for 25 and 100 spectral channels

Our updated CTIS simulator was created with generalisability in mind. It generates a CTIS image from an input hyperspectral cube which can have arbitrary spatial and spectral dimensions while enabling control of both geometric and optical parameters. The main purpose of the simulator is to speed up the generation of CTIS images for the training of the neural networks and remove the need for a system matrix for large dimensions.

We have significantly improved the CTIS simulator employed in our previous work.[22] The updated simulator executes Equation (1), and similarly to the system matrix incorporates a PSF, spectral sensitivity corrections in terms of diffraction sensitivity and illumination as well as additive zero-mean Gaussian noise. It emulates our laboratory CTIS system,[28] from which the optical parameters used in the simulator have been measured; see Supplementary Materials Sections S2–S5 for more details. The scripts for the CTIS simulator and generation of $\boldsymbol{H}$ in MATLAB and Python are available on Github (https://github.com/madspeters/CTIS).

A simulated 450 × 450 pixels CTIS image is generated from a 100 × 100 × 25 data cube is shown on the left in Figure 2. All simulated CTIS images comprise the central zeroth order and eight neighbouring first orders. The geometric parameters of the simulator enable control of the cube dimensions, the distance between the zeroth and first orders and the pixel shift between projections of the spectral channels in the first orders.

In Figure 2 the geometric parameters are a shift of 2 pixels, a 27-pixel distance between the zeroth and first order, and the 100 × 100 × 25 dimensions of the data cube. Note that the simulated images from the chosen parameters are smaller than those captured by our CTIS camera. Additionally, the determined optical parameters are also incorporated as seen from the non-uniform intensities among the first orders. To assist the networks in identifying regions of interest on CTIS images, we pre-process the 2-D CTIS images with the division into nine smaller 150 × 150 images, each containing either the zeroth or a first order as shown in the left panel of Figure 2. For the 100-channel case (right panel of Figure 2), the optical parameters are updated correspondingly to match the higher number of spectral channels. That results in a 750 × 750 pixels CTIS image, which is divided into nine smaller 250 × 250 images as in the 25-channel case.

As our main goal is to reconstruct 3-D hyperspectral cubes from real CTIS images, the data cubes used to simulate CTIS images (and used as ground truth in training) are captured by our pushbroom HSI system. The system consists of a conveyor belt and an HSI camera, which contains an ImSpector V10E spectrograph (Specim), a 50 mm C Series vis-NIR objective (Edmund Optics) and a Qtechnology QT5022 system equipped with a CMV4000-E12 CMOS sensor (CMOSIS). The pushbroom system acquires 216 spectral channels between the wavelengths 384 nm and 972 nm with a spatial resolution of 0.33 mm pixel^{-1}.

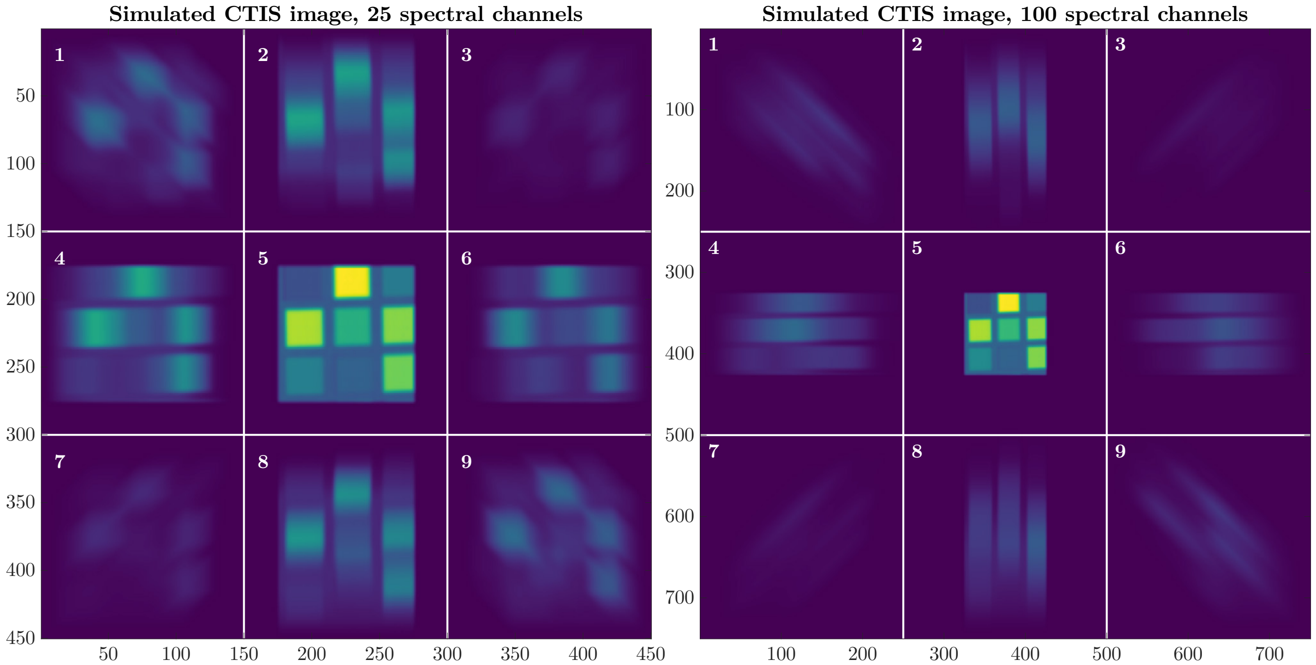

Figure 2. Left (right): a simulated 450 × 450 (750 × 750) pixels CTIS image for a (100 × 100 × 25) data cube of a Colorchecker divided into nine smaller 150 × 150 (250 × 250) pixels images, which are fed into the CNNs.

Data preparation

The data used in this work originate from 178 different pushbroom cubes of various objects, such as potatoes, a Colorchecker and books, with varying spatial dimensions, ranging from 200 × 200 to 499 × 400, all with 216 channels. Seven of these cubes are reserved as completely unseen cubes for testing the networks' capability of generalisation, while the remaining 171 cubes are used for training, validation and testing. These unseen cubes contain pears, potatoes with wireworm defects and carrots. RGB visualisations of some of the used pushbroom cubes created by combining three spectral channels at 470 nm (blue), 549 nm (green) and 650 nm (red) are displayed in Figure 3. RGB images of all 178 cubes are presented in Supplementary Material in Section S6.

The data preparation for the neural networks is illustrated in Figure 4: for each of the seen 171 pushbroom cubes, we crop 768 smaller cubes of dimensions (100,100,25). The 216 spectral channels are reduced to 25 by removing the first 10 and last 6 spectral channels, which have a low signal-to-noise ratio and averaging over 8 consecutive spectral channels for the remaining 200 spectral channels—resulting in 25 channels. Then, the simulator is applied to these smaller cubes to generate CTIS images. In total, there are 131,328 samples, which are divided into training (91,998), validation (19,665) and test (19,665) sets. The training set is used to train the networks, while the validation set is employed to prevent overfitting. We evaluate the models via the test set that has not been involved in the training process. Besides, we create 805 extra samples from each of the seven unseen cubes to assess how well the models can handle completely new data. All in all, each sample contains a 100 × 100 × 25, a full 450 × 450 pixels CTIS image and the corresponding pre-processed 150 × 150 × 9 CTIS image. The full CTIS images are used in the EM algorithm, while the neural networks take the pre-processed CTIS images as input and hyperspectral cubes as output. For the 100-channel case, the procedure of data generation is the same, where a sample contains a 100 × 100 × 100 data cube, a full 750 × 750 pixels CTIS image and the corresponding pre-processed 250 × 250 × 9 CTIS image.

Models of data cube reconstruction with 25 spectral channels from CTIS images

In this section, we elaborate on three different methods of hyperspectral data cube reconstruction from CTIS images: the EM algorithm, CNNs and hybrid CNN-EM models. The models will be trained and tested on data consisting of 450 × 450 pixels CTIS images cropped into 9 regions of 150 × 150 pixels as input and hyperspectral cubes of 100 × 100 pixels with 25 spectral channels as output.

EM reconstruction algorithm

The EM algorithm[29] is routinely utilised in the CTIS reconstruction.[9,20,30] Since H is generally non-invertible, an estimate $\hat{f}$ of the hyperspectral cube is obtained using a sparse implementation of the iterative EM algorithm, which effectively attempts to solve Equation (1) for a given CTIS image $\boldsymbol{g}$. The EM algorithm first computes an estimated CTIS image $\hat{\boldsymbol{g}} = \quad \hat{}^{(k)}$ in the expectation step.

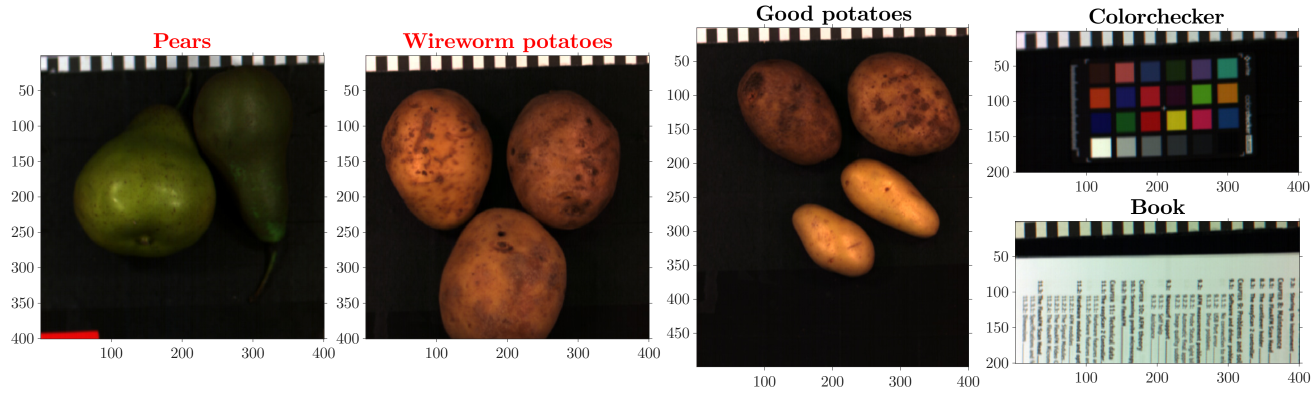

Figure 3. RGB visualisation examples of seen (unseen) hyperspectral cubes with black (red) titles captured by the pushbroom system. The examples of unseen cubes consist of pears and potatoes with wireworms, while the seen cubes consist of good potatoes, a Colorchecker and a book. RGB images are created by combining the 470 nm (blue), 549 nm (green) and 650 nm (red) spectral channels.

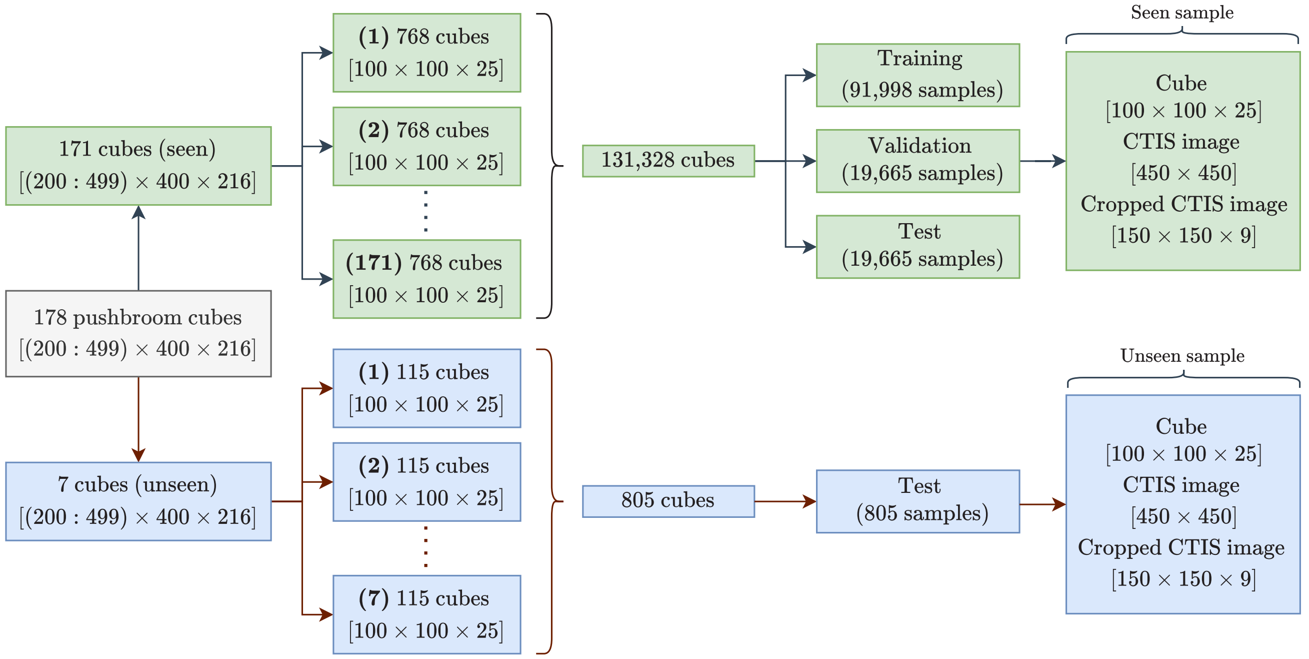

Figure 4. Overview of the data pipeline for the data generation with 25 spectral channels: the captured 178 pushbroom cubes are grouped into 171 seen and 7 unseen cubes, which are cropped into smaller 100 × 100 × 25 cubes. The samples are divided into training, validation and test sets for the neural networks.

The subsequent maximisation step computes a correction factor for all voxels in $\hat{\boldsymbol{f}}^{(k)}$ as a back-projection of the ratio of the acquired $\boldsymbol{g}$ and estimated CTIS image $\hat{\boldsymbol{g}}$, normalised by the summed rows of $\boldsymbol{H}$:

$$\hat{f}^{(k+1)} = \frac{\hat{f}^{(k)}}{\sum_{i=1}^{q^2} H_{ij}} \odot \left(H^T \frac{g}{H\hat{f}^{(k)}} \right) \qquad (2)$$

where k is the iteration index, $\hat{\boldsymbol{f}}^{(k)}$ is the k-th estimate of the hyperspectral cube, $\sum_{i=1}^{q^2} H_{ij}$ is the vectorised summation of rows in $\boldsymbol{H}$, $\boldsymbol{H}^{\mathsf{T}}$ is the transposed system matrix and the symbol $\odot$ denotes the Hadamard (elementwise) product. Notice that Equation (2) combines the expectation and maximisation steps into a single step.

Initialisation is typically either $\hat{\boldsymbol{f}}^{(0)} = ones(r,1)$[30] or $\hat{\boldsymbol{f}}^{(0)} = \boldsymbol{H}^{\mathsf{T}}\boldsymbol{g}$,[9] where the former is utilised in this work for EM reconstructions. As 10–30 EM iterations are typically required,[21] we chose to use 20 iterations for the standalone EM and apply only 10 iterations for the hybrid models.

Convolutional neural networks

To implement networks, we use TensorFlow,[31] an open-source machine learning platform that contains Keras,[32] a deep learning application programming interface. Since both the inputs and outputs are multi-channel images, it is natural to utilise only 2-D convolutional layers, denoted by Conv2D in Keras, without applying flattening which converts 2-D images or 3-D cubes into 1-D vectors as often done in CNN image classifications. The network architecture is presented in Figure 5, where the output dimensions for each layer are indicated. The left-most layer represents the input layer of dimensions (150, 150, 9). The input layer is followed by a sequence of multiple Conv2D layers without padding, each containing 25 kernels with varying kernel sizes. As the input is passed through the network, the dimensionality is gradually decreasing toward (100, 100, 25), the dimension of the output layer.

For all Conv2Ds, the convolution kernel (filter) moves one pixel rightwards or downwards over a 3-D image between two successive applications of the kernel. The kernel size for each Conv2D layer can be inferred from the difference between its output dimensions and that of the previous layer. The first Conv2D layer, for instance, has kernel size (6, 6) (height, width) that decreases the input from (150, 150, 9) into (145, 145, 25) in the absence of padding. The kernel sizes are gradually reduced throughout the network. Overall, the CNN consists of 174,100 trainable parameters in total. The motivation of such a tunnel-like architecture is to create a small network with relatively few parameters that features a short training time and fast predictions, namely a fast

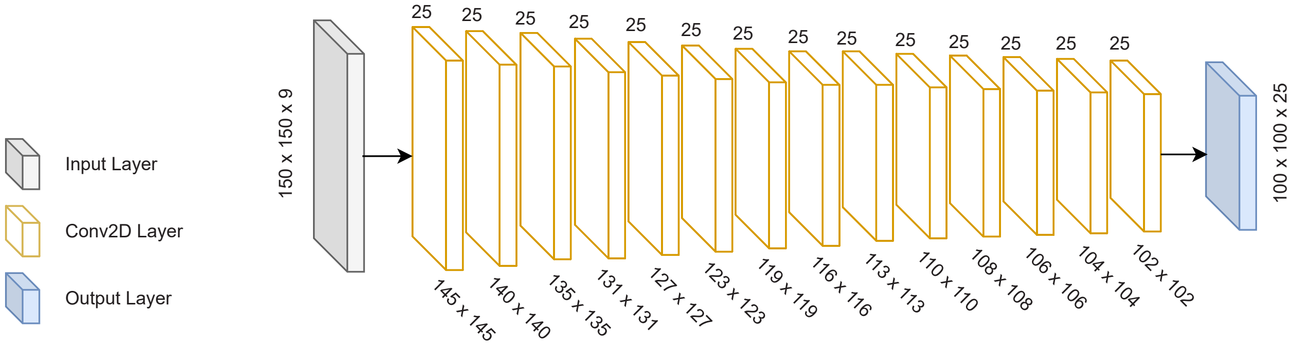

Figure 5. Illustration of the proposed CNN1 for reconstructing a 100 × 100 image with 25 spectral channels. The network has a total of 174,100 trainable parameters and consists of an input layer (which takes CTIS images), Conv2D layers and an output layer (which generates hyperspectral cubes). For the hidden layers, the output spatial dimensions are specified at the bottom with the output spectral dimensions on the top. The dimensions of the input and output layers are also denoted.

forward pass/propagation. The network is referred to as CNN1, which reconstructs cubes (network output) from CTIS images (input).

Lastly, we also test a U-Net,[33] which has been extensively used for image segmentation. The U-Net architecture is adapted to match the dimensions of the input and output in question (Figure 6) and consists of 22,225,329 trainable parameters, more than 100 times larger than the CNN1.

Hybrid models

From our previous work,[22] it has been demonstrated that neural networks can efficiently and accurately reconstruct cubes with smaller errors than the EM method, provided that the networks have been exposed to images of similar objects or geometries in the training phase. In addition, unlike the EM algorithm, the networks are not vulnerable to noise and complex spatial variation in the images, giving rise to consistent results.

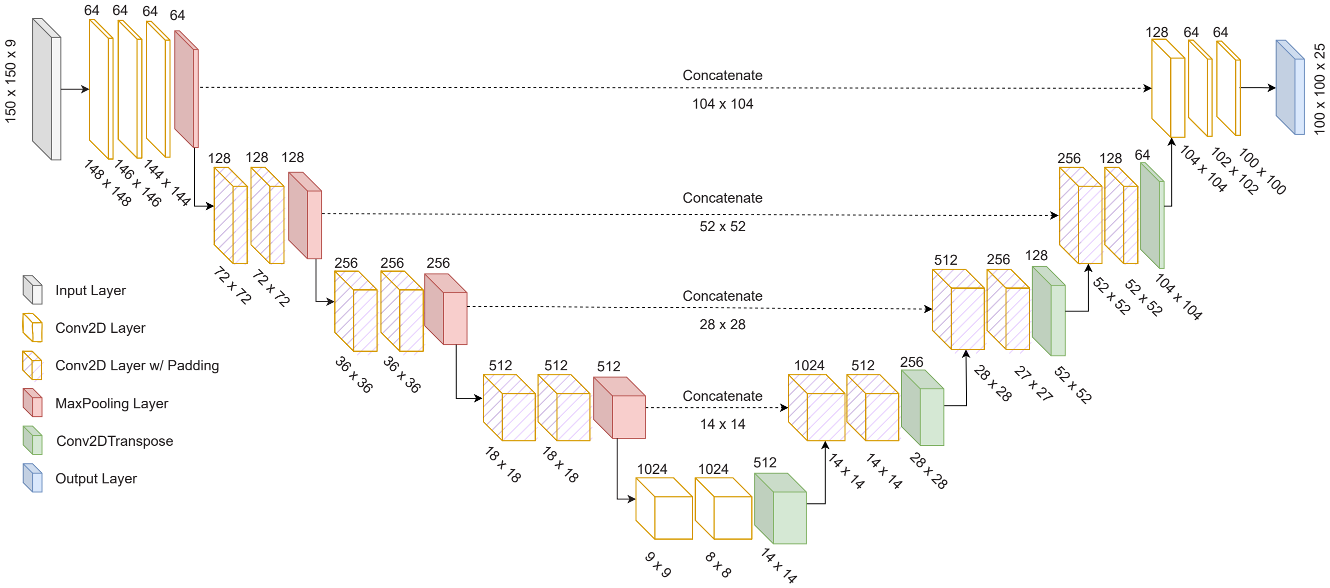

Figure 6. Illustration of the proposed U-Net for reconstructing a 100 × 100 image with 25 spectral channels. The network has a total of 22,226,329 trainable parameters and consists of an input and output layer as well as layers of Conv2D, Conv2D with padding, MaxPooling and Conv2DTranspose. The output spatial dimensions of the layers are specified at the bottom with the output spectral dimensions on the top. The skip-concatenation connections are indicated by the dotted arrows.

To improve the network's generalisability and overcome EM's weakness against noise and complex geometry, it is natural to combine the two methods sequentially. That is, one first uses a network, to reconstruct a cube from a CTIS image, which is passed to the iterative EM algorithm as an initial guess to further refine the network reconstruction. The Comparison of model performance section details how the hybrid models produce better results than the EM algorithm and networks alone for both seen (cubes that are part of training and test sets) and unseen (neither part of training nor test set) data cubes.

It should be pointed out that Reference 34 has proposed to solve ill-posed inverse problems using iterative deep neural networks, where a known, traditional algorithm, which takes inverse problems, is applied before neural networks. In the context of hyperspectral cube reconstruction, it corresponds to the reverse sequence: EM → Network. Because the network can handle the noise better than the EM, the sequence we suggest; Network → EM, will in principle provide a more consistent and stable reconstruction with better performance than the reverse one as demonstrated in the Results of 25 spectral channels section. Moreover, for the sequence Network → EM, one can experiment with different numbers of EM iterations to attain an optimal balance between the performance and execution time. By contrast, one must retrain the network in the EM → Network framework once the number of iterations changes, as the input of the network is the outcome of the EM algorithm which depends on the number of iterations. In other words, the Network → EM has more flexibility in the implementation of real-world data.

For the reverse case (EM → Network), the EM outputs hyperspectral cubes which are fed into the CNN. Therefore, both the input and the output of the CNN are cubes and thus have the dimensions $100 \times 100 \times 25$. We use the same number of kernels and the same kernel sizes as in CNN1 for a fair comparison. To match the input and output dimensions we use padding in each layer, to maintain the dimensionality throughout the network. This network consists of 188,500 trainable parameters, slightly more than the CNN1 due to padding, and is referred to as CNN2.

All in all, we have three hybrid models, CNN → EM, EM → CNN and U-Net → EM which are denoted as CNN1-EM, EM-CNN2 and U-Net-EM, respectively.

Network training

For the network training procedure for all three networks, we choose the Adam optimiser, Mean-Squared-Error (MSE) as the loss function and the (Keras) "EarlyStopping" callback, which ceases the training when the MSE of the validation set stops improving, to prevent over-training. Moreover, we set the batch size to 32 and 500 epochs are used, which are divided into 10, 10 and 480: a learning rate of 4×10^{-5} is assumed for the first 10 epochs, but is reduced by a factor of 2 for the second 10 epochs and a factor of 4 for the following 480 epochs. In addition, during the 480 epochs, the learning rate decays exponentially—it is reduced by a factor of 0.9 for every 50,000 steps. Moreover, 20 iterations are carried out for the standalone EM algorithm but only 10 iterations for the EM step in the hybrid models.

To quantify the performance of the different methods, we utilised the MSE and peak signal-to-noise ratio (PSNR) in decibels as error metrics:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_1)^2, \quad PSNR = 10\log_{10}\left(\frac{Max2}{MSE}\right) \tag{3}$$

where N is the data sample size, i is the sample index, $\hat{\boldsymbol{Y}}_i$ is the reconstructed cube and $\boldsymbol{Y}_i$ is the ground truth. Lastly, Max is the maximum pixel value which is 255 since the image format is 8-bit.

Noise influence

Before presenting our results, we should point out that the noise term n in Equation (1), parametrised by a zero-mean Gaussian distribution, can affect the EM reconstruction performance.[25] Zeng *et al.*[27] investigated the ill-conditioned image reconstruction problem in the presence of noise and showed that the EM algorithm at the beginning demonstrates a short convergent trend but then diverges from the desired solution. Empirically, for zero-mean Gaussian noise with a standard deviation of 0.5 at a maximal pixel value of 255, i.e., a noise level of ≈0.2 %, the application of the EM after neural networks sometimes increases MSE instead of refining the network predictions which is similar to the behaviour observed by Zeng *et al.* This is especially the case for network predictions that are very close to the ground truth, where the effect of a small mismatch is accumulated during the EM iterations, which involve matrix multiplications between tensors of large dimensions and perturbs the reconstructed cubes away from the true ones. The effects of noise on our models are briefly investigated in Appendix C.

For simplicity, in the following results section, we do not consider the noise term when generating data, effectively assuming the noise is either small enough or can be included into cubes $\boldsymbol{f}$.

Comparison of model performance and extension toward hyperspectral regimes

In this section, we first present our results by comparing the different reconstruction approaches. Second, we investigate whether the hybrid approach can be applied in a more challenging hyperspectral scenario with 100 spectral channels.

Results of 25 spectral channels

The results of applying the training procedure explained above and the EM algorithm are summarised in Table 1. Based on the results we make the following observations and comments.

- The hybrid models perform better than both the standalone EM and respective networks:
  ~26 % improvements on CNN1-EM compared to CNN1, and ~14 % improvements on U-Net-EM compared to U-Net. The improvements occur for both seen and unseen cubes, while the EM alone is by far worse than the network models. It illustrates the strength of the sequential combinations of the network and EM—the networks provide a good initial condition for the EM to further improve.
- The EM results for the seen and unseen cubes, MSE 122.50 versus 153.04, indicate the performance inconsistency associated with the spatial variation in the data. In fact, the fluctuation in MSE is even more pronounced when comparing samples from different individual pushbroom cubes.
- The model CNN1-EM outperforms EM-CNN2. That corroborated our previous argument that the EM is more vulnerable to noise and complex geometry and might yield inconsistent results as inputs to the network. In this case, the CNN2 must cope with different degrees of variation from the EM output and overall performs worse than the CNN1-EM.
- By contrast, in the CNN1-EM model, the vulnerability of EM has been mitigated by the network which decreases the noise and supplies a good starting point for the EM. Moreover, one can freely experiment with different numbers of EM iterations in the CNN1-EM to attain an optimal balance between the accuracy and reconstruction time, whereas the CNN2 must be retrained whenever the number of iterations is changed in the EM step.
- The U-Net as a much bigger network (more than 100 times larger than CNN1) has a more significant MSE increase, a factor of 11, from the seen to unseen cubes while the MSE only doubles for CNN1. The U-Net has, nonetheless, a smaller MSE than CNN1 for the unseen cubes. It illustrates first that it is very important to include different objects with various geometries into the training set for the network, especially large ones, to maintain consistent performance over all different objects. Second, a smaller network is more robust against new data and performs more consistently than a large network.

In Figure 7, we show the 10th channel of the ground truth (referred to as GT) and reconstructed cubes for the seen (top) and unseen (bottom) cubes as well as the residual error. The U-Net-EM (EM alone) has the best (worst) accuracy as seen from residual errors. The spectral information, characterised by the mean pixel value of a representative 5 × 5 area denoted by the white square in the images, is also shown. That is, the mean pixel value as a function of spectral channels. The U-Net-EM (pink dotted line) follows the GT best, while EM is unable to reproduce the spectral shape in detail. The advantage of applying EM after the networks is more visible for the unseen cubes. For example, around the middle channels, it moves the spectra close to the true ones. Additional

Table 1. MSE (PSNR) for the models under consideration as well as the EM algorithm, where the test set of the seen (19,665) samples and unseen (805) samples are used to evaluate the model performance. The average reconstruction time per cube is also shown for each model.

	Seen cubes (19,665 samples)	Unseen cubes (805 samples)	Time (ms)
EM	121.50 (27.3)	153.04 (26.3)	48.05
CNN1	10.88 (37.8)	22.37 (34.6)	0.90
CNN1-EM	7.91 (39.2)	16.67 (35.9)	26.31
EM-CNN2	8.45 (38.9)	18.63 (35.4)	26.06
U-Net	0.91 (48.6)	10.34 (38.0)	1.69
U-Net-EM	0.78 (49.2)	8.83 (38.7)	27.06

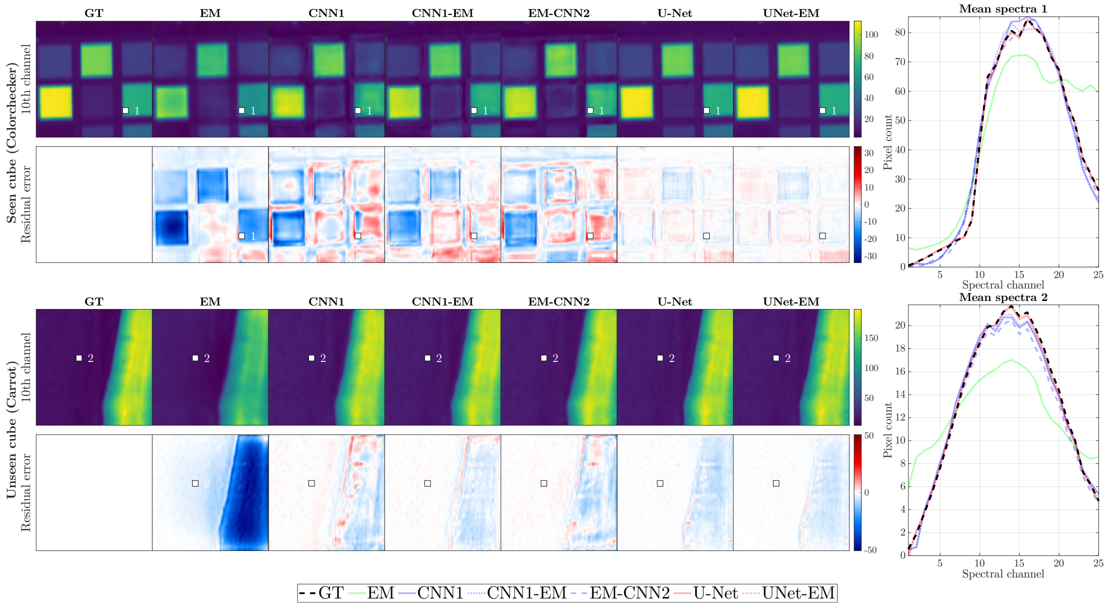

Figure 7. Comparison of 25-channel reconstructions of the seen (Colorchecker) and unseen (Carrot) cube for the ground truth (denoted by GT), EM (20 iterations), CNN1, CNN1-EM (10 iterations), EM-CNN2, U-Net and U-Net-EM. The 10th spectral channel and the corresponding residual error (reconstructed GT) are shown in the images, while the mean 5 × 5 pixels (indicated by the white squares) spectra are also plotted.

figures for different cubes are shown in Figures 9 and 10 in Appendix A. Finally, to easily visualise how closely the reconstructed cubes resemble the true ones, we show the RGB images of some representative reconstructed cubes in Supplemental Material in Section S7.

Results of 100 spectral channels

We are now in a position to tackle a more challenging scenario of 100 spectral channels with the hybrid approach. Only a U-Net network (modified to match dimensions accordingly) of 22.2 million parameters is considered as it has the best performance among the network models for 25 channels. We follow closely the data-creation and training procedure used in the 25-channel case with one major difference—100 epochs are assumed instead of 500 for the network training to reduce the training time as it takes much longer for each epoch with 100 spectral channels. Finally, we apply the EM algorithm with 10 iterations to further refine the network predictions. The performance of the standalone and hybrid models is summarised in Table 2.

For the seen cubes, the hybrid model is much better than the EM algorithm by a factor of 9.7 in MSE while the hybrid model is better than the U-Net by 19 %. For unseen cubes, although the performance of the U-Net significantly decreases, it still outperforms the EM. The EM step in the U-Net-EM improves the U-Net predictions by 40 %, which is significantly higher than the 25-channel case (only 15 %). It implies the EM step is more pivotal in the hyperspectral regime, especially for the completely unseen data.

Table 2. MSE (PSNR) for the EM algorithm, U-Net and hybrid network. Similar to Table 1, the test set of the seen (19,665) samples and unseen (805) samples are used to evaluate the model performance.

	Seen cubes (19,665 samples)	Unseen cubes (805 samples)
EM	27.62 (33.7)	45.99 (31.5)
U-Net	3.51 (42.7)	19.80 (35.2)
U-Net-EM	2.83 (43.6)	11.80 (37.4)

Similar to Figure 7, we show the 52nd channel of true and reconstructed cubes as well as the spectral information in Figure 8 for a seen and unseen cube. The EM alone can only reconstruct the overall spectral shape but again fails to capture small variations and details as seen by comparing the black (ground truth) and green (EM) lines in the spectral plots. From the 52nd channel, it is also evident that spatial information is lost for EM, where the text from the seen cube is no longer readable (top row of the left panel in Figure 8) and the finer spatial details of the unseen cube are missing. This smearing or smoothing of the high-frequency spatial and spectral components is characteristic of the EM algorithm. Moreover, the EM in the U-Net-EM visibly reduces the level of residual errors with respect to the U-Net predictions. Additional figures for different cubes are shown in Figures 11–13 in Appendix B. Similar to the 25-channels case, to visualise how closely the reconstructed cubes mimic the true ones, we show RGB visualisations of some representative reconstructed cubes in Section S8 in Supplemental Material.

To summarise, we have demonstrated that the hybrid model can make very good predictions on a variety of cubes for 100 channels—a hyperspectral regime—and can decently generalise to totally unseen samples. It outmatches both the standalone U-Net and the EM algorithm. The improvement by including the EM (compared to the U-Net alone) is more pronounced than in the 25-channel case: 19–40 % for 100 channels versus ~14 % for 25 channels. The hybrid model has proved promising with broad applications in the reconstruction of real-world hyperspectral imaging.

Conclusions

The CTIS, a snapshot hyperspectral imaging system, is a compact and efficient way of providing hyperspectral information. A 3-D hyperspectral cube can be

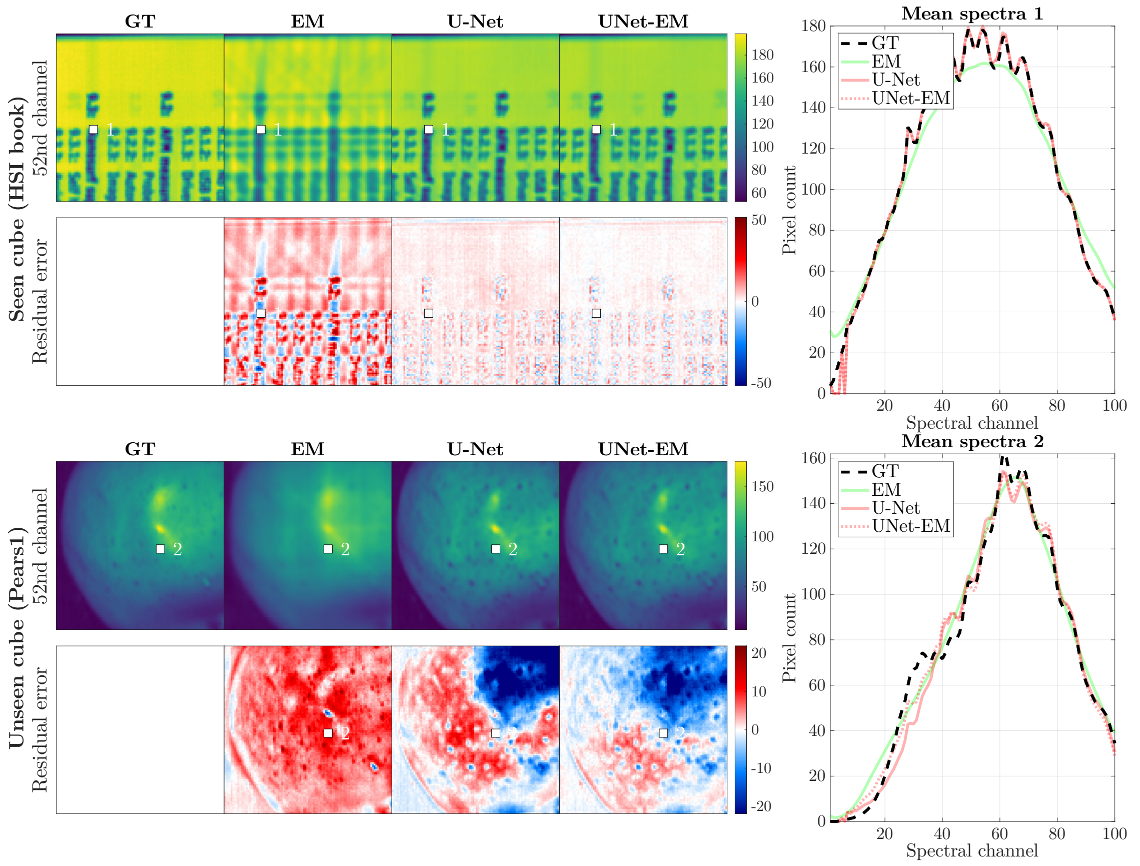

Figure 8. Comparison of 100-channel reconstructions of the seen (HSI book) and unseen (Pears1) cube for GT, EM (20 iterations), U-Net and U-Net-EM. The 52nd spectral channel and the error (reconstructed GT) for the respective reconstructions are shown in the images, while the mean 5 × 5 pixels (indicated by the white squares) spectra are shown in the plots.

reconstructed from a CTIS image that entails wider applications than the 2-D image itself. It has been shown[22] that CNNs can be employed for fast, reliable cube reconstruction, provided that the CNNs have been exposed to objects of similar geometry during the training. On the other hand, the iterative reconstruction algorithms, e.g., EM, need no training and can be applied to different CTIS images.

In this work we propose a very simple but novel way of cube reconstruction—a hybrid model. The model first utilises a network to reconstruct a hyperspectral cube from a CTIS image, and the reconstructed cube is fed into the EM algorithm as an initial value of the cube, which is then recursively updated. We have trained and tested our hybrid models based on real-world hyperspectral cubes from a pushbroom camera and CTIS images, generated by applying the cubes to a realistic CTIS simulator. The simulator (see Supplemental information) emulates a real CTIS system based on experimental measurements of the PSF, illumination and diffraction sensitivity as a function of the wavelength. We studied scenarios of 25 and 100 spectral channels. For both scenarios, the data consist of training (91,998 samples), validation (19,665) and test (19,665) sets, cropped from 171 different pushbroom cubes. The performance of models is evaluated based on the test set as well as an extra 805 samples, created from 7 unseen pushbroom cubes. For comparison, we investigated different methods of reconstruction.

For 25 spectral channels, we consider the standalone EM, CNN, U-Net and hybrid models of CNN1-EM (EM is applied after CNN), EM-CNN2 (CNN is applied after EM) and U-Net-EM, where the U-Net, CNN1 and CNN2 have 22.23, 0.17 and 0.19 million trainable parameters, respectively. The performance of the models is summarised in Table 1. First, it has been found that U-Net-EM is the best model whereas EM alone has the worst performance. That shows the advantages of neural networks over the traditional reconstruction algorithm, as demonstrated in our previous work.[22] Second, all hybrid models perform better than the corresponding networks alone with the improvement ranging from 14 % to 27 %. In addition, the CNN1-EM outperforms the reverse order EM-CNN2. It highlights the synergy between the networks and EM as follows: the EM can help networks to cope with unseen data as it can be applied to images of any objects. On the other hand, the network is less prone to noise and provides a good and stable initial guess for the EM to further improve the results. The reverse order EM-CNN2 will, by contrast, be subject to the noise and inconsistent results from the EM. Finally, the U-Net, as a much larger network, experiences a more noticeable performance loss (roughly a factor of 10) from the seen to the unseen cubes as opposed to the much smaller CNN1 (a factor of 2), although the U-Net still reconstructs cubes better. It indicates that a smaller network is more robust against new types of cubes. Additionally, the smaller network will also have a shorter forward pass time, i.e., faster predictions. Therefore, one should find the optimal configuration by considering the reconstruction time, performance and robustness when advancing to the real-time reconstruction.

We have also demonstrated that the hybrid approach works well in the hyperspectral regime by exploring a 100-channel case. The results are summarised in Table 2. The inclusion of the EM significantly improves the U-Net results, especially for the unseen data—the MSE is reduced by nearly a factor of 2.

To summarise, we have presented a very simple but novel hybrid model of hyperspectral cube reconstruction by applying the traditional EM algorithm after neural networks. These two methods are complementary—the network, which is less susceptible to noise and complex spatial variation in the images, provides a refined initial condition for the EM while the EM further improves the network's results and assists it to deal with very different kinds of objects.

Acknowledgements

MJA, RLE, MTF, WCH and BJ acknowledge partial funding from Food & Bio Cluster Denmark funded by The Danish Ministry of Higher Education and Science.

MTF and WCH acknowledge partial funding from The Villum Foundation and CenSec grant funded by The Danish Ministry of Higher Education and Science (CenSec).

MSP acknowledges partial funding from the Innovation Fund Denmark (IFD) under File No. 1044-00053B. We acknowledge partial support from Food & Bio Cluster Denmark. This work was performed using the UCloud computing and storage resources, managed and supported by eScience center at the University of Southern Denmark.

Appendix A. Additional comparisons of the different reconstructions, 25 channels

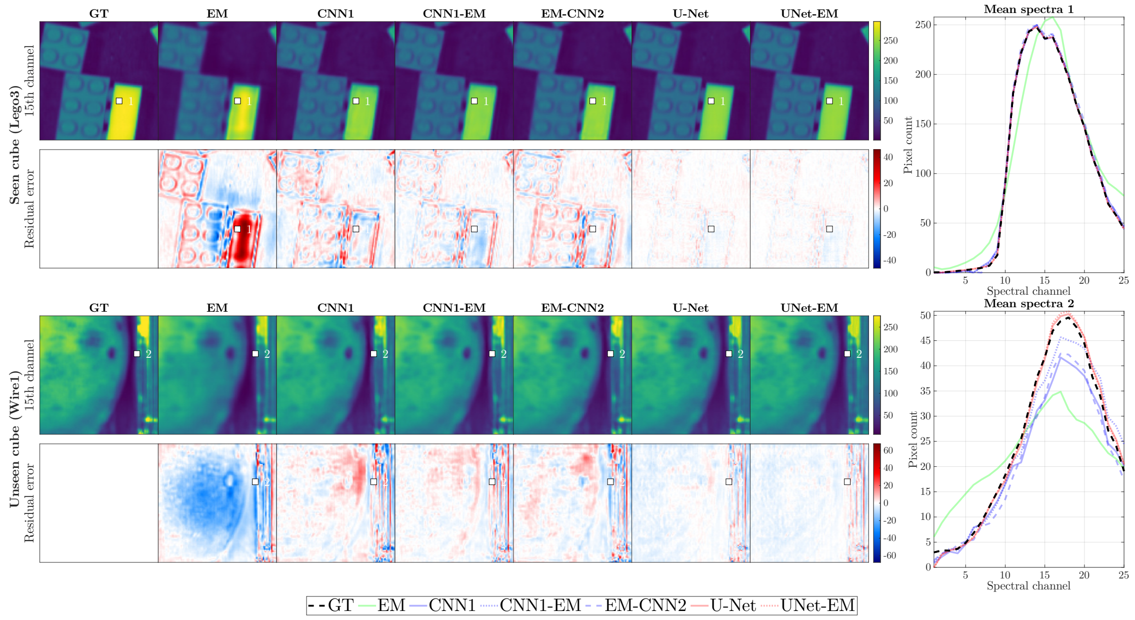

Figure 9. Comparison of 25-channel reconstructions of the seen (Lego3) and unseen (Wire1) cubes for GT, EM (20 iterations), CNN1, CNN1-EM, EM-CNN2, U-Net and U-Net-EM. The 15th spectral channel and the error (reconstructed GT) for the respective reconstructions are shown in the images, while the mean 5×5 pixels (indicated by the white squares) spectra are shown in the plots.

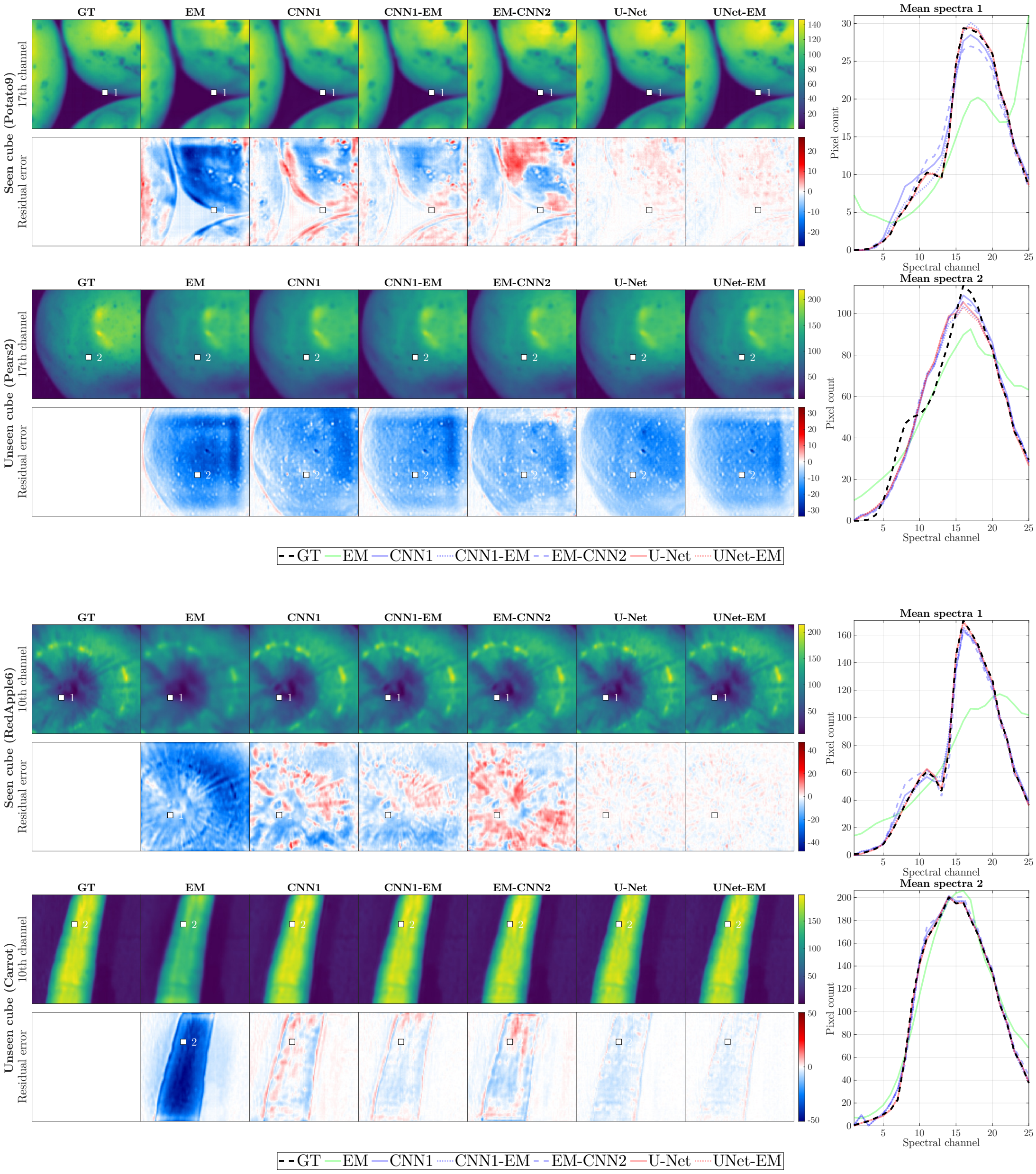

Figure 10. Comparison of 25-channel reconstruction of the seen (Potato9 and RedApple6) and unseen (Pears2 and Carrot) cubes for GT, EM (20 iterations), CNN1, CNN1-EM, EM-CNN2, U-Net and U-Net-EM. The 17th and 10th spectral channel and the error (reconstructed GT) for the respective reconstructions are shown in the images, while the mean 5 × 5 pixels (indicated by the white squares) spectra are shown in the plots.

Appendix B. Additional comparisons of the different reconstructions, 100 channels

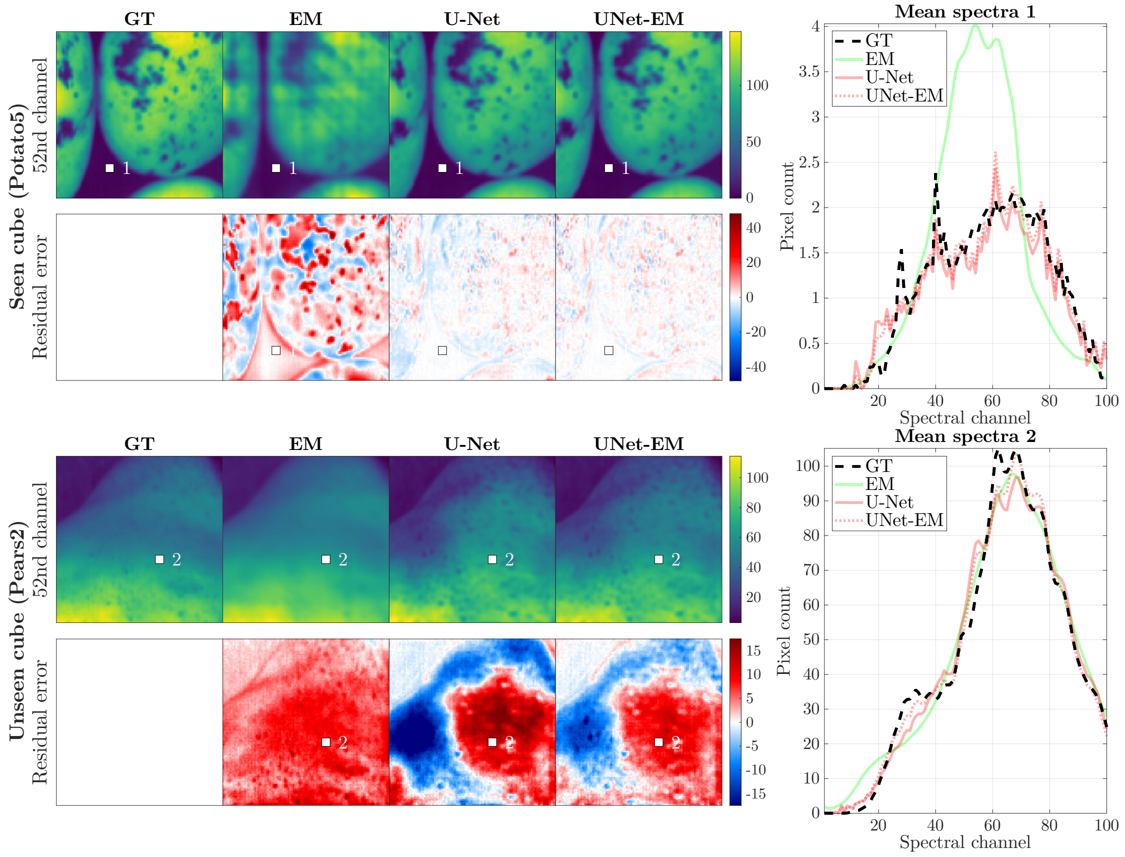

Figure 11. Comparison of 100-channel reconstruction of the seen (Potato5) and unseen (Pears2) cubes for GT, EM (20 iterations), CNN1, CNN1-EM, EM-CNN2, U-Net and U-Net-EM. The 52nd spectral channel and the error (reconstructed GT) for the respective reconstructions are shown in the images, while the mean 5 × 5 pixels (indicated by the white squares) spectra are shown in the plots.

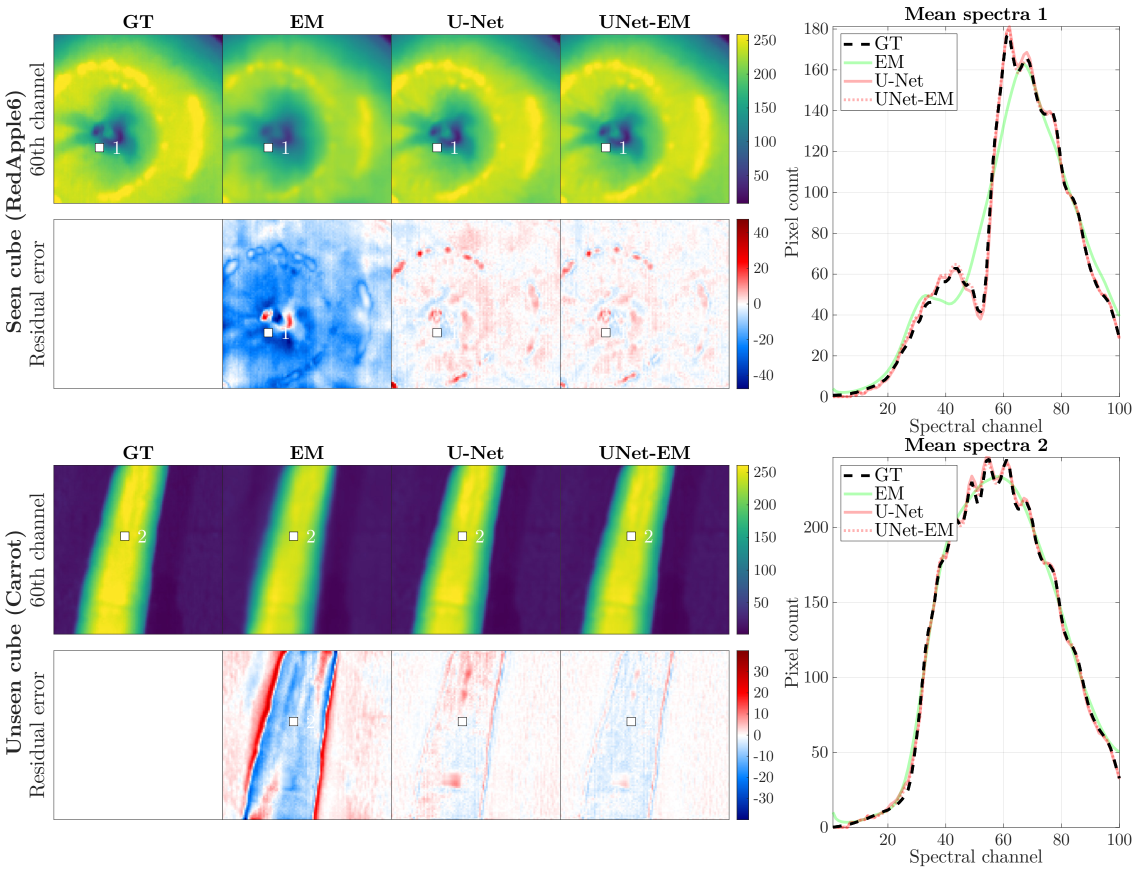

Figure 12. Comparison of 100-channel reconstruction of the seen (RedApple6) and unseen (Carrot) cubes for GT, EM (20 iterations), CNN1, CNN1-EM, EM-CNN2, U-Net and U-Net-EM. The 60th spectral channel and the error (reconstructed GT) for the respective reconstructions are shown in the images, while the mean 5 × 5 pixels (indicated by the white squares) spectra are shown in the plots.

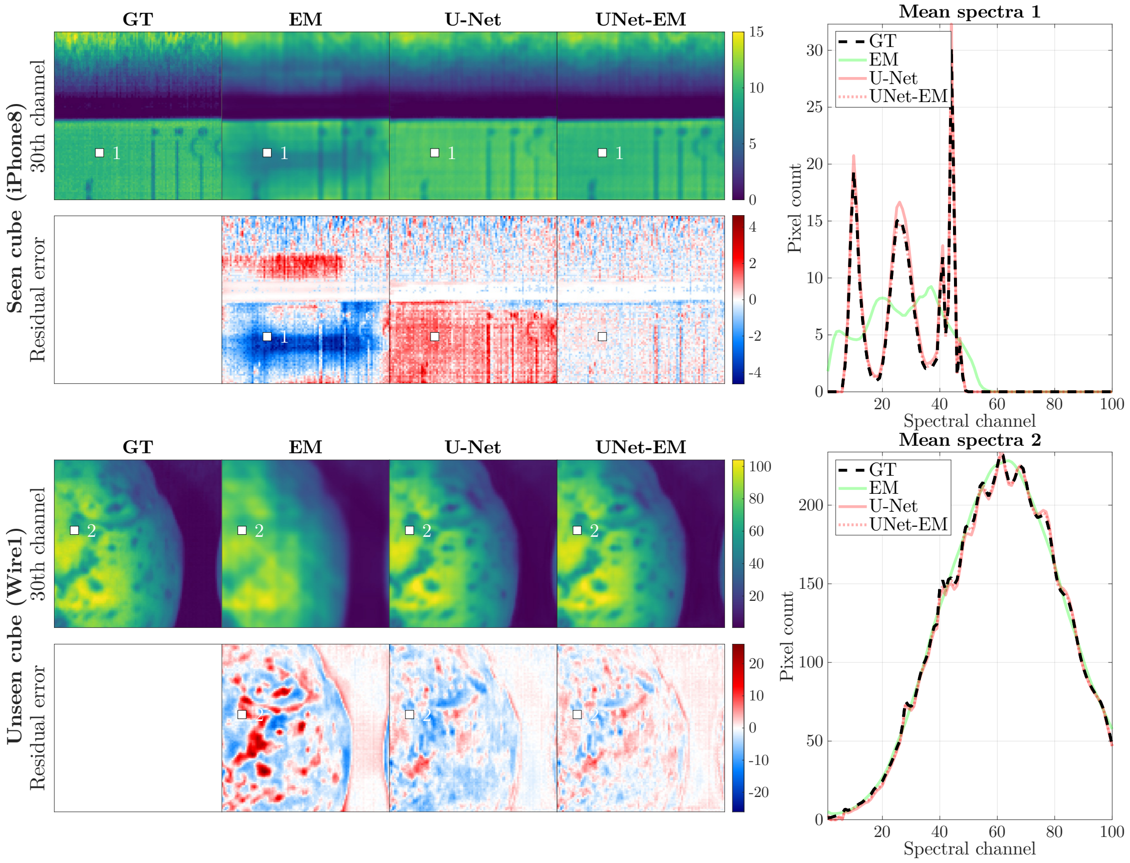

Figure 13. Comparison of 100-channel reconstructions of the seen (iPhone 8) and unseen (Wire1) cubes for GT, EM (20 iterations), CNN1, CNN1-EM, EM-CNN2, U-Net and U-Net-EM. The 30th spectral channel and the error (reconstructed GT) for the respective reconstructions are shown in the images, while the mean 5 × 5 pixels (indicated by the white squares) spectra are shown in the plots.

Noise investigations

The following is a brief investigation into the effects of noise and the proposed hybrid approach. To quantify the effect of noise, white Gaussian noise is added to the training validation and test data, which consists of both seen and unseen cubes, and we here focus only on the hybrid CNN1-EM model. The applied noise is characterised by a zero-mean Gaussian distribution with a standard deviation of $\sigma = 0.5$ (approximately corresponding to the noise level found on our CTIS camera). All negative pixel values resulting from the addition of the Gaussian noise are replaced by zero. Notice that the added noise is not incorporated in the $\boldsymbol{H}$ matrix of the EM algorithm.

We have found that the performance of CNN1 is not affected by the presence of the noise. Instead in some cases, CNN1 performs better on unseen cubes when trained on noisy data. In other words, the CNN benefits from being exposed to noise and thus becomes more robust. This is already a known feature of the CNNs.[35,36]

On the other hand, the performance of CNN1-EM on the noisy data is shown in Figure 14, which successfully reproduces the converging—followed by the diverging—behaviour observed by Zeng *et al.*[27] Furthermore, the quality of the initial guesses provided by CNN1 also has an impact on the performance of the second EM step. With a decent initial guess (close to the ground truth such as the blue line in Figure 14 for seen cubes), the EM results will diverge quickly, whereas with a mediocre initial guess (such as the green line for unseen cubes) the EM results stay convergent for a longer time.

In fact, the existence of sizable noise implies a noticeable mismatch between the real CTIS image and the approximated, $\hat{g} = H\hat{f}$. In this case, the $\boldsymbol{H}$ matrix used in the EM algorithm does not correctly map the hyperspectral cubes to the CTIS image—that violates the assumption on which the EM algorithm [described in Equation (2)] is based, thus making the EM unable to attain the true cubes.

There are at least two ways to circumvent the noise issue. First, one can properly model systematic errors of the CTIS system and subtract them from the CTIS image such that the remaining random noise is small and under control. Alternatively, one could capture more than one CTIS image of the same object and average out the systematic errors. Second, the cube $\boldsymbol{f}$ in Equation (1) can be redefined to include the random noise:

$$g = Hf + n = H\hat{f} \tag{4}$$

In this case, one can obtain an estimated cube $\hat{f}$, which contains the noise, by the EM algorithm in a consistent way. As long as the noise term is small enough, e.g., the standard deviation of the noise is less than one pixel count, the reconstructed cubes will be a good approximation to the real cube $\boldsymbol{f}$.

Conclusively, to utilise the proposed hybrid models, one must very carefully model $\boldsymbol{H}$, such that the level of noise is under control.

References

1. A.F.H. Goetz, G. Vane, J.E. Solomon and B.N. Rock, "Imaging spectrometry for Earth remote sensing", *Science* **228(4704),** 1147–1153 (1985). https://doi.org/10.1126/science.228.4704.1147
2. K. Hege, D. O'Connell, W. Johnson, S. Basty and E. Dereniak, "Hyperspectral imaging for astronomy and space surveillance", *Proc. SPIE* **5159,** 01 (2004). https://doi.org/10.1117/12.506426
3. N. Keshava, "Distance metrics and band selection in hyperspectral processing with applications to material identification and spectral libraries", *IEEE Trans Geosci Remote Sens.* **42(7),** 1552–1565 (2004). https://doi.org/10.1109/TGRS.2004.830549
4. H. Lee, M.S. Kim, Y.-R. Song, C.-S. Oh, H.-S. Lim, W.-H. Lee, J.-S. Kang and B.-K. Cho, "Non-destructive evaluation of bacteria-infected watermelon seeds using visible/near-infrared hyperspectral imaging: Bacterial-infected watermelon seed detection using hyperspectral image", *J. Sci. Food Agric.* **97(4),** 1084–1092 (2017). https://doi.org/10.1002/jsfa.7832
5. Y.-Y. Pu, Y.-Z. Feng and D.-W. Sun, "Recent progress of hyperspectral imaging on quality and safety inspection of fruits and vegetables: A review", *Compr. Rev. Food Sci. Food Saf.* **14(2),** 176–188 (2015). https://doi.org/10.1111/1541-4337.12123
6. B. Boldrini, W. Kessler, K. Rebner and R. Kessler, "Hyperspectral imaging: a review of best practice, performance, and pitfalls for inline and online applications", *J. Near Infrared Spectrosc.* **20(5),** 438 (2012). https://doi.org/10.1255/jnirs.1003
7. T. Okamoto and I. Yamaguchi, "Simultaneous acquisition of spectral image information", *Opt. Lett.* **16(16),** 1277–1279 (1991). https://doi.org/10.1364/OL.16.001277
8. T.V. Bulygin and G.N. Vishnyakov, "Spectrotomography: a new method of obtaining spectrograms of two-dimensional objects", in *Analytical Methods for Optical Tomography*, Ed by

G.G. Levin. International Society for Optics and Photonics, SPIE, Vol. 1843, pp. 315–322 (1992). https://doi.org/10.1117/12.131904

9. M. Descour and E. Dereniak, "Computed-tomography imaging spectrometer: experimental calibration and reconstruction results", *Appl. Opt.* **34(22),** 4817 (1995). https://doi.org/10.1364/AO.34.004817
10. M.E. Gehm, R. John, D.J. Brady, R.M. Willett and T.J. Schulz, "Single-shot compressive spectral imaging with a dual-disperser architecture", *Opt. Express* **15(21),** 14013 (2007). https://doi.org/10.1364/OE.15.014013
11. X. Cao, X. Tong, Q. Dai and S. Lin, "High resolution multispectral video capture with a hybrid camera system", *CVPR 2011*. IEEE, pp. 297–304 (2011). https://doi.org/10.1109/CVPR.2011.5995418
12. A. Bodkin, A. Sheinis, A. Norton, J. Daly, S. Beaven and J. Weinheimer, "Snapshot hyperspectral imaging: the hyperpixel array camera", in *SPIE Defense, Security, and Sensing*, Ed by S.S. Shen and P.E. Lewis. SPIE, p. 73340H (2009). https://doi.org/10.1117/12.818929
13. B. Geelen, N. Tack and A. Lambrechts, "A compact snapshot multispectral imager with a monolithically integrated per-pixel filter mosaic", in *SPIE MOEMS-MEMS*, Ed by G. von Freymann, W.V. Schoenfeld and R.C. Rumpf. SPIE, p. 89740L (2014). https://doi.org/10.1117/12.2037607
14. L. Gao, R.T. Kester, N. Hagen and T.S. Tkaczyk, "Snapshot image mapping spectrometer (IMS) with high sampling density for hyperspectral microscopy", *Opt. Express* **18(14),** 14330–14344 (2010). https://doi.org/10.1364/OE.18.014330
15. A.R. Harvey, D.W. Fletcher-Holmes, S.S. Kudesia and C. Beggan, "Imaging spectrometry at visible and infrared wavelengths using image replication", in *Electro-Optical and Infrared Systems: Technology and Applications*, Ed by R.G. Driggers and D.A. Huckridge. International Society for Optics and Photonics, SPIE, Vol. 5612, pp. 190–198 (2004). https://doi.org/10.1117/12.580059
16. M.W. Kudenov and E.L. Dereniak, "Compact real-time birefringent imaging spectrometer", *Opt. Express* **20(16),** 17973–17986 (2012). https://doi.org/10.1364/OE.20.017973
17. C. Douarre, C.F. Crispim-Junior, A. Gelibert, L. Tougne and D. Rousseau, "On the value of CTIS imagery for neural-network-based classification: a simulation perspective", *Appl. Opt.* **59(28),** 8697–8710 (2020). https://doi.org/10.1364/AO.394868
18. C. Douarre, C. Crispim-Junior, A. Gelibert, G. Germain, L. Tougne and D. Rousseau, "CTIS-net: A neural network architecture for compressed learning based on computed tomography imaging spectrometers", *IEEE Trans. Comput. Imaging* **7,** 572–583 (2021). https://doi.org/10.1109/TCI.2021.3083215
19. M.D. Vose and M.D. Horton, "A heuristic technique for CTIS image reconstruction", *Appl. Opt.* **46(26),** 6498 (2007). https://doi.org/10.1364/AO.46.006498
20. N. Hagen, E.L. Dereniak and D.T. Sass, "Fourier methods of improving reconstruction speed for CTIS imaging spectrometers", *Proc. SPIE 6661, Imaging Spectrometry XII*, 666103 (2007). https://doi.org/10.1117/12.732669
21. L. White, W.B. Bell and R. Haygood, "Accelerating computed tomographic imaging spectrometer reconstruction using a parallel algorithm exploiting spatial shift-invariance", *Opt. Eng.* **59(5),** 055110 (2020). https://doi.org/10.1117/1.OE.59.5.055110
22. W.-C. Huang, M.S. Peters, M.J. Ahlebæk, M.T. Frandsen, R.L. Eriksen and B. Jørgensen, "The application of convolutional neural networks for tomographic reconstruction of hyperspectral images", *Displays* **74,** 102218 (2022). https://doi.org/10.1016/j.displa.2022.102218
23. Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard and L.D. Jackel, "Backpropagation applied to handwritten zip code recognition", *Neural Comput.* **1(4),** 541–551 (1989). https://doi.org/10.1162/neco.1989.1.4.541
24. Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition", *Proc. IEEE* **86(11),** 2278–2324 (1998). https://doi.org/10.1109/5.726791
25. D.L. Snyder, M.I. Miller, L.J. Thomas and D.G. Politte, "Noise and edge artifacts in maximum-likelihood reconstructions for emission tomography", *IEEE Trans. Med. Imaging* **6(3),** 228–238 (1987). https://doi.org/10.1109/TMI.1987.4307831
26. J.F. Scholl, *The Design and Analysis of Computed Tomographic Imaging Spectrometers (CTIS) Using Fourier and Wavelet Crosstalk Matrices*. PhD thesis, University of Arizona (2010).
27. G.L. Zeng and G.T. Gullberg, "Unmatched projector/backprojector pairs in an iterative reconstruction algorithm", *IEEE Trans. Med. Imaging* **19(5),** 548–555 (2000). https://doi.org/10.1109/42.870265

28. M.S. Peters, R.L. Eriksen and B. Jørgensen, “High-resolution snapshot hyperspectral computed tomography imaging spectrometer: real-world applications”, *Proc. SPIE 12136, Unconventional Optical Imaging III* 199–204 (2022). https://doi.org/10.1117/12.2621128
29. L.A. Shepp and Y. Vardi, “Maximum likelihood reconstruction for emission tomography”, *IEEE Trans. Med. Imaging* **1(2),** 113–122 (1982). https://doi.org/10.1109/TMI.1982.4307558
30. D.W. Wilson, P.D. Maker and R.E. Muller, “Reconstructions of computed-tomography imaging spectrometer image cubes using calculated system matrices”, *SPIE 3118, Imaging Spectrometry III* 184–193 (1997). https://doi.org/10.1117/12.283827
31. TensorFlow Developers, *TensorFlow*.
32. F. Chollet, *Keras*.
33. O. Ronneberger, P. Fischer and T. Brox, “U-net: Convolutional networks for biomedical image segmentation”, in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*, Ed by N. Navab, J. Hornegger, W.M. Wells and A.F. Frangi. Springer International Publishing, pp. 234–241 (2015). https://doi.org/10.1007/978-3-319-24574-4_28
34. J. Adler and O. Öktem, “Solving ill-posed inverse problems using iterative deep neural networks”, *Inverse Problems* **33(12),** 124007 (2017). https://doi.org/10.1088/1361-6420/aa9581
35. T.S. Nazaré, G.B. Costa, W.A. Contato and M. Ponti, “Deep convolutional neural networks and noisy images”, *Iberoamerican Congress on Pattern Recognition*. Springer, pp. 416–424 (2017). https://doi.org/10.1007/978-3-319-75193-1_50
36. Y. Qian, M. Bi, T. Tan and K. Yu, “Very deep convolutional neural networks for noise robust speech recognition”, *IEEE/ACM Trans. Audio, Speech Lang. Process.* **24(12),** 2263–2276 (2016). https://doi.org/10.1109/TASLP.2016.2602884