Reconstruction of CASSI-Raman Images with Machine-Learning

Andreas Brorsson* Markus Nordberg†
David Gustafsson*
*FOI Swedish defence research agency, C4ISR, Linköping, Sweden.
†FOI Swedish defence research agency, Defence and Security, Systems and Technology, Stockholm, Sweden.
{andreas.brorsson, markus.nordberg}@foi.se

Abstract

Raman spectroscopy is a well-established method to detect small amounts of potentially dangerous substances. In a Coded Aperture Snapshot Spectral Imaging (CASSI) system spatial and spectral information are mixed resulting in an ensemble of compressed sensing measurements. A reconstruction method is applied to the Compressed Sensing (CS) measurement to reconstruct a hyperspectral cube containing the Raman spectra for the locations in the scene. Traditional reconstruction methods based on regularization such as Total Variation (TV) are time consuming which reduce the number of applications where the technology is applicable. A machine learning reconstruction approach using Convolutional Neural Network (CNN) is presented. The loss function for the CNN is a combination of reconstruction error and re-projection error of the reconstructed Raman spectra. Simulation of CS-measurements of samples containing different chemical substances and different concentration levels are reconstructed with high precision. The reconstruction time using the novel machine learning approach decreases several orders of magnitude.

1. Introduction

Improvised Explosive Devices (IEDs) are a common and increasing threat in the society. To find these, it is desirable to not only have a detection equipment capable of detecting them but also from a safe distance. Laser-based techniques e.g. Raman spectroscopy is a technique capable of this. By illuminating the explosive or residues with laser light and collect the scattered light these can be detected and identified. Today this technique has come a long way and for many substances very small amounts can be detected. There is a need to shortening the measurement times and also to further decrease the influence of the background noise. To achieve this Östmark et al. have used trace detectors based on multi spectral imaging Raman spectroscopy [16] and also hyper spectral imaging [15] based on Compressed Sensing (CS) [6]. Even though the measurement times are a few seconds for these detectors today's CS-algorithms are slow with computational times of several minutes. The consequence is that it is not practical possible to measure on large surfaces in real time. In this publication we investigate the possibility to use machine learning and especially Convolutional Neural Networks (CNN) to essentially speed up the decoding of the measured data.

2. Related Work

Using deep learning and other machine learning approaches to reconstruct high resolution images from compress sensing measurements is a very active research field. Higham et al. [11] propose a Deep Convolutional Auto-Encoder approach (DCAE) where the encoding step is the CS part i.e. applying a binary pattern to an image – and the decoding step is the reconstruction of an image. The non-learned hidden latent representation is a set of single pixel CS-measurements and the DCAE is learned to reconstruct the image from the latent CS-representation. They reconstruct 128x128 pixel gray scale images (film) using solely 333 binary patterns (2% sub-sampling ratio) at 30 frames per second. The reconstructed images are of high visual quality and the reconstruction time has been reduced in order of magnitudes.

Kulkarni et al. [14] present a two steps non-iterative deep learning approach for reconstruction of high resolution monochrome and color images. They propose a patch based reconstruction algorithm which reconstruct fixed sized image patches. The patches are then combined to reconstruct a full size image. The reconstructed images are denoised to improve the visual quality of the image. Kulkarni et al. [14] present results from simulated CS-measurements but also for CS-measurements from a real system.

Wang et al. [22] have developed a framework that jointly optimize the binary pattern used to obtain the CS-measurement and a CNN architecture for reconstruction the hyperspectral cube using the CS-measurements. The simulated CS-measurements are from a Coded Aperture Snapshot Spectral Imaging (CASSI) type of system were spatial and spectral information has been mixed. This is the first, and only as far as the authors know, attempt to use ML for reconstruction of a hyperspectral image from a CS-measurement from a (simulated) CASSI-system. Reconstruction of high resolution hyperspectral images require a large amount of memory, Wang et al. [22] therefore proposed a patch based reconstruction. The hyperspectral image is divided into patches and the patches are reconstructed to form a full resolution hyperspectral image. Two concatenated CNNs were used, a spatial CNN followed by a spectral CNN.

Sun et al. [20] propose a dual path reconstruction of images from CS-measurements. The dual path proposal is motivated by the issue that many reconstruction methods fail to reconstruct fine-scale details in the image – texture – while coarse scale details – structure – is reconstructed fairly well. The reconstruction is divided into two parts: geometry and texture. The reconstructed geometry and texture is combined to an image using ordinary addition. The structure-texture decomposition using the G-norm was developed by Aujol et al. [2]. The image structure is reconstructed using a CNN while the image texture is reconstructed using an attention based CNN, the reconstructed part are combined using ordinary addition.

Gedalin et al. [9] present a patch-based CNN that reconstruct hyperspectral images using CS-measurements from a so-called Compressive Sensing Miniature Ultra-Spectral Imager (CS-MUSI). DeepCubeNet, as they call their architecture, is based on a U-NET [19] CNN architecture.

Wu et al. [23] present a generative based approach for image reconstruction. A Generative Adversarial Network (GAN) is trained to generate image from a low dimensional set of parameters z i.e. $G(z)$ generates images using the parameters z. A transfer function F is used to transform images to CS-measurements. Let m be a vector of CS-measurement, let F be a transfer function corresponding to the CS-measurement and let $G(z)$ be an image generator. An image is reconstructed by finding the image using $G(z)$ that minimizing the distance between the measurements m and the $F(G(z))$. If the ground true image is in the span of the generator $G(z)$, theoretical bounds on the reconstruction error is known.

As far as the authors know machine learning has never been used to reconstruct Raman spectrum from an active CASSI-system.

3. Raman spectroscopy

When light fall onto a molecule the photons can be elastically scattered by Rayleigh scattering and also inelastically scattered by Raman scattering [18]. The Raman scattered light is very week, typically only one photon per million. Even though, it is still strong enough to be measured at tens or even hundreds of meters distance. Since the Raman light is scattered inelastically it carries information about the scattering molecule, specificity the vibrations in the molecule and since different types of molecules have different vibrations the information from Raman scattering is molecule specific. In a way it holds a sort of fingerprint from the molecules. Figure 1 shows example of Raman

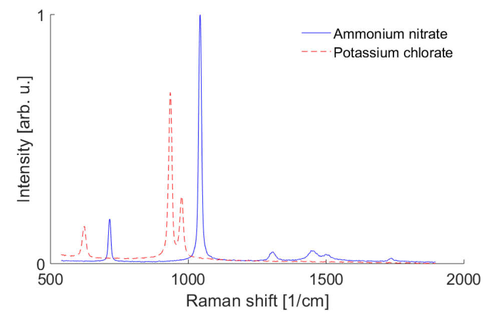

Figure 1. Raman spectra from ammonium nitrate and potassium chlorate measured with 1064 nm excitation wavelength.

spectra from ammonium nitrate and potassium chlorate. It can be seen that ammonium nitrate have two strong peaks while potassium chlorate have three.

4. Coded aperture snapshot spectral imaging

To measure a 3D data cube (two spatial and one spectral) with hyperspectral information without using a scanning system is challenging since available detectors are 1D or 2D arrays. In addition, the data collection is often a compromise between spectral resolution, spatial resolution, measurement time and light throughput. By using techniques such as CASSI [10, 21, 1], it is possible to mix spectral and spatial information and thereafter from this information recreate all of the data, without sacrificing resolution. Typical CS-algorithms used for this are Two-step Iterative Shrinkage/Thresholding (TwIST) [3] and Gradient Projection for Sparse Reconstruction (GPSR) [8]. The fundamental steps of the CASSI system is shown in Figure 2. The real hyperspectral data of the target (top-left) is imaged on a coded aperture. The coded aperture consists of a binary pattern that either block the light or pass the light that falls onto it. The parts of the imaged target that pass the coded aperture is dispersed by a grating i.e. a spectrograph (bottom-middle) and is recorded by a 2D detector (bottom-right). The image read by the detector is a mixture of spatial and spectral information of the target scene and can be decoded by a computer to retain a hyperspectral data cube (top-right). To increase the resolution (spatial and spectral) of the recreated hyperspectral cube several images with different coded apertures can be measured [13].

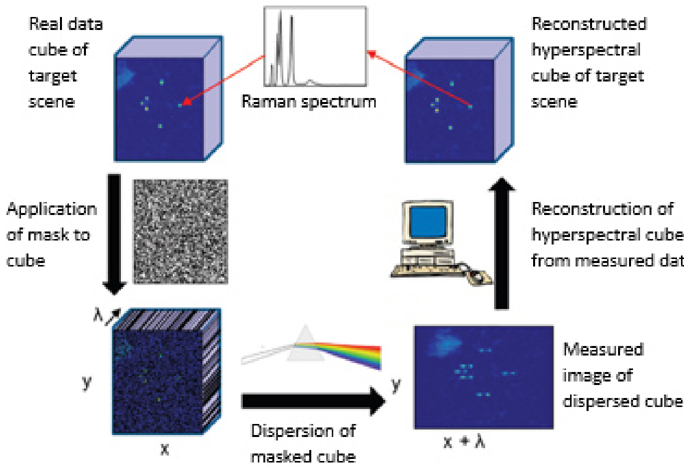

Figure 2. Fundamental steps of CASSI.

5. Experimental

A sketch of a CASSI setup can be seen in figure 3. A pulsed laser illuminates the target whereby the scattered light is collected through a lens and imaged onto a DMD (digital mirror array). The Rayleigh scattered light is blocked by a notch filter that pass the Raman scattered light through. A DMD consists of many small mirrors and

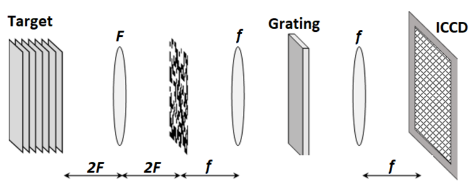

Figure 3. Sketch of the important parts in a CASSI system using a DMD.

can be positioned in two different angles. One of them reflects the light onto a CMOS-camera that shows a white light image of the target and the other angle reflects the light onto a imaging spectrograph where the light is dispersed and recorded by an intensified CCD-camera. The DMD also function as the coded aperture. A typical setting of the DMD can be seen in Figure 4, white means light going to the spectrograph while black means going to the CMOS-camera.

6. Machine learning based spectrum reconstruction

Lately, Machine Learning (ML) and Deep Learning (DL) have been applied to solve inverse problems with the goal to reconstruct information from transformed data or a set of transformed data. An example of inverse problem is to reconstruct a noise-free (clean) image from a noisy image or a high-resolution image from a set of low-resolution

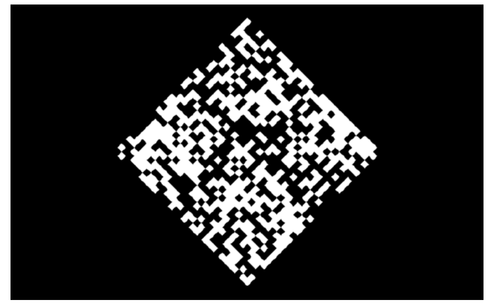

Figure 4. Example of a binary coded aperture shown by the DMD.

image (super-resolution). Inverse problems are commonly ill-posed and require some regularization assumption on the data to be solvable. Compressed sensing is an ill-posed inverse problem where one wants to reconstruct high resolution data using a set of compressed measurement.

CS in Raman spectroscopy using a CASSI system can be viewed as a regression problem where one wants to reconstruct the Raman spectrum from a set of CS-measurements. The problem is introduced in a simplified formulation where a single Raman spectrum is reconstructed and later reformulated to reconstruct a full Raman Spectrum matrix. Single Raman spectrum cannot be reconstructed instead a vector (a line on the sensor) must be reconstructed simultaneously.

Here we describe it seen from one spatial point to later include a whole line. Let x be a Raman spectrum (in a single spatial point), y be a CS-measurement from a CASSI-Raman system and let f be the transfer function from Raman spectrum to a CS-measurement i.e.

$$y = f(x) \tag{1}$$

Given a known transfer function f, as discussed in section 7, one can compute the CS-measurement y for a given Raman spectrum x. The goal is to train a DL-net that is able to reconstruct the Raman spectrum x from the CS-measurement y. One want to find a function that solves

$$\hat{x} = c(y; w) \tag{2}$$

where c is a CNN with parameters w that decode the CS-measurement y and reconstruct an estimate of the Raman spectrum $\hat{x}$. The reconstruction error is defined using the L_1-norm as

$$\|x - \hat{x}\|_1 \tag{3}$$

and express how well $c(y; w)$ can reconstruct the Raman spectrum x from the CS-measurements y. The L_1-norm is commonly used but other norms are also frequently used as error measure. Given the transfer function f and the reconstructed Raman spectrum $\hat{x}$, the transfer function can be applied to the reconstructed Raman spectrum resulting in an emulated CS-measurement

$$\hat{y} = f(\hat{x}) \tag{4}$$

where $\hat{y}$ is a emulated CS-measurement from the reconstructed Raman spectrum $\hat{x}$. The CS-measurement y from the Raman spectrum x is known and the distance between emulated CS-measurement $\hat{y}$ from $\hat{x}$ and y from x should be small. The re-projection error is defined using the L_1-norm as

$$\|y - \hat{y}\|_1 \tag{5}$$

which measure the distance between the original CS-measurement and the re-projected CS-measurement of the reconstructed Raman spectrum.

The loss-function to minimize with respect to the parameters w is a linear combination of the reconstruction error and the re-projection error and is defined as

$$E(y) = \|x - \hat{x}\|_1 + \lambda \|y - \hat{y}\|_1 \tag{6}$$

where λ is a weight term which balance the impact of the reconstruction error and re-projection error. $\hat{x}$ and $\hat{y}$ depends on w through the CNN net $c(y; w)$ and one wants to determine the weight w that minimize the error over a fixed training set. The parameters have been optimized using ADAM, gradient based search with adaptive step length [12].

The line-based reconstruction of the Raman spectrum from the vectors of CS-measurements is similar to the simplified presentation above. Let X be a $M \times N$ matrix where each column is a Raman spectrum containing M wavelengths. X contains N Raman spectra – the N columns in the matrix – with measurement of M wavelengths defined by the rows in the matrix. The transfer function $f(X)$ maps the matrix to a CS-measurement

$$Y_i = f(X; P_i) \tag{7}$$

where Y_i is a CS-measurement vector of length N where the binary pattern P_i has been used in the transfer function. One wants to reconstruct the matrix X from a set of CS-measurement vectors $Y = (Y_1, \cdots, Y_N)$ where different patterns P_i have been used in the transfer function. Let

$$X = C(Y; W) \tag{8}$$

where C is a CNN that reconstructs the $M \times N$ matrix X containing the Raman spectra, $Y = (Y_1, \cdots, Y_N)$ the corresponding CS-measurement vectors and W are the net (weight) parameters. Again the loss-function for $C(Y; W)$ is composed of two parts – the reconstruction error and the re-projection error. The reconstruction error is defined as the L_1-norm between the ground truth matrix X and the reconstructed matrix $\hat{X}$ as

$$\left\|X - \hat{X}\right\|_1 \tag{9}$$

where X and $\hat{X}$ are $M \times N$ matrices. Again the re-projection error is computed by applying the transfer function with the corresponding patterns P_i to the reconstructed matrix $\hat{X}$ i.e.

$$\hat{Y}_i = f(\hat{X}, P_i) \tag{10}$$

and $\hat{Y} = (\hat{Y}_1, \cdots, \hat{Y}_N)$. The re-projection error is defined as

$$\left\|Y - \hat{Y}\right\|_1. \tag{11}$$

The loss-function is a weighted combination of the reconstruction error and the re-projection error

$$E(Y) = \left\|X - \hat{X}\right\|_1 + \lambda \left\|Y - \hat{Y}\right\|_1 \tag{12}$$

where λ is a weight that is used to balance the two terms.

Given a fixed CNN net structure – number of layers, number and size of the convolutional kernels in each layer and activation function – a (local) optima can be found by minimizing the combined loss-function using adaptive step length gradient search method. Deciding the architecture of the CNN is a complex question that involves many questions based on hands-on knowledge and trial-and error. Crucial CNN issues include:

- Transfer learning – can pre-trained CNN be used to improve performance, decrease training time and limit the size of the training set?
- Number of layers, type of layers and number of nodes in each layer.
- Number of convolutional layers, size of the convolutional kernels and type of pooling layer.
- Connection structure between the layers – fully-connected, sparse-connected or skipping connections.
- Activation function.

In some cases well established recommendation for the different parameters exist that can be used as initial values. For inverse problem and especially for CS-problem the previous experience on parameter setting is limited and for CASSI-Raman system no previous results have been reported. Determine suitable parameter setting has been largely using trail-and-error. Explicit net configurations that has been evaluated are discussed in Section 8.

7. Method

One of the challenges with DL is acquiring or simulating large amounts of training data. The data used in the training of a DL network should preferably be obtained with the hardware for which the network is to be trained. The difficulty with system acquired data is that a large dataset with large variation needs to be recorded and annotated, which is time consuming and costly. The alternative is to simulate data, this opens up the possibility of generating a lot of (annotated) data, with large variation. The disadvantages of

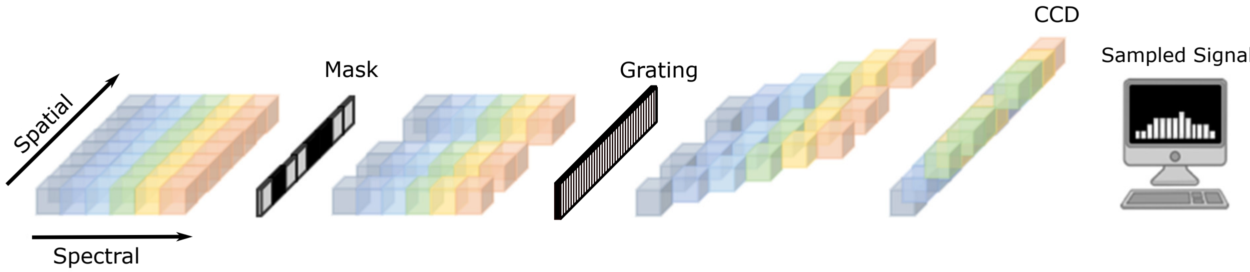

Figure 5. The simplified model of the CASSI system.

simulated data is the difficulty to model an existing system including noise and distortions in a satisfactory manner.

The benefits of using a simplified simulation at this early stage of research outweigh the use of recorded data and the complex model of the physical system. It is a big challenge how to apply DL models trained on simulated training-data to data from the developed CASSI system. By using a simplified model of the system the more theoretical inversion (reconstruction) is in focus.

To create training data, a simplified model of the CASSI system was generated. The simplifications that were made were:

- No "cross-talk" between adjacent rows, which means that one row at a time is reconstructed. This can be realized in reality by measuring every other row in the mask, but twice as many measurements needs to be recorded. The advantages of separating each row are that it reduces the size of the neural network and also reduces the time to train the network.
- Constant dispersion i.e. the mapping for a row is constant regardless of the wavelength.
- No optical aberration.
- Lowering the spatial resolution initally to 64×64, while keeping the spectral resolution to 340 bins.
- Four different masks, i.e. four measurements per scene.

This results in a CASSI model that each row in the hyperspectral cube can be illustrated according to Figure 5.

Figure 5 shows how the light, for different wavelengths, passes through the simulated CASSI system. The hyperspectral cube for a specific row (far left) hits the mask, blocking parts of the light. Then the partially blocked cube meets a dipersive element and is shifted one pixel for each wavelength. This light hits the detector (CCD) as spatially and spectrally mixed data. Because the detector is monochrome, it measures the intensity of the light regardless of the wavelength and the detected signal becomes similar to the image on the computer on the far right. This represents the measured data that is later used to recreate the hyperspectral cube for this specific row.

The generation of training data was controlled by an algorithm that randomly plotted Raman spectra on a 64x64 image. The spatial substances were in the form of rectangles or ellipses that were rotated and scaled randomly. The tool created both a hyperspectral cube and a measurement of this cube. With the help of the tool, a number of parameters could be controlled to obtain the desired training data:

- Number of substances in the image.
- Size of the substance.
- Which substances to use.
- Magnitude of the spectra.
- Magnitude of additive noise in the hyperspectral cube and measurement.

8. Reconstruction network

Since DL has not been applied to CASSI-Raman to the authors' knowledge, DL reference points were sought in a similar but different sub-area, namely Compressive Imaging (CI). Within CI, both pure Fully Connected (FC) and CNN networks [11, 14] and generative models [23, 4] have been used successfully to invert one-pixel camera measurements. Common to all these networks is to take the measurements y in an FC layer as a first step and transform the number of nodes to a desired amount which then goes into a CNN with several layers which gives out the reconstructed image.

Input to the model is a ($n_0 = K \times (N + M - 1) = 4 \times (64 + 340 - 1) = 1612$) long 1D array, where the spatial and spectral data is modulated. Output from the model is $n_1 = N \times M = 64 \times 340$ 2D matrix, where columns represent spatial position and rows spectral data. Thus every output from the model is a reconstructed row with one to one mapping to a row of the mask.

8.1. Encode-decode architecture

In [11, 14], both used a FC layer as input followed by a CNN part with the same spatial resolution as the target output. Considering the size of the data an unmodified pipeline

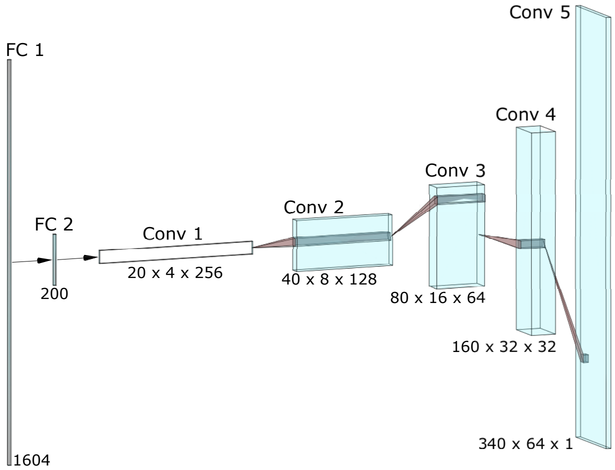

Figure 6. Network architecture, where *FC* is fully connected layers and *Conv* is transposed 2D convolution layers.

would yield an enormous first FC layer $> 30M$ parameters compared to $1 - 4M$ parameters in the CNN layers. To many parameters in the architecture increase the risk of over-fitting and increased time to train. Furthermore, the FC part will dominate the network in relation to the CNN part.

With the current simplified model, four masks are used for each measurement, which result in $\frac{n_0}{n_1} \simeq 7\%$ sub-sampling ratio. Our hypothesis is that the input data can be compressed further with trained parameters in the FC layer and this is based on two insights.

- The Raman spectra is more sparse than a regular image/HSI, that is for non-zero elements k in x, $k_{\text{raman}} < k_{\text{image}}$. In [7], the relationship $m \geq k \log N/k$, holds for CS-measurements, where N is the resolution of the hyperspectral image and m is the minimum number of measurements. In [5], images could be recovered with $m_{\text{image}} < 3\% \Rightarrow m_{\text{raman}} < m_{\text{image}}$ sub-sampling ratio.

- The second argument, which is more empirical, shows that generative networks such as GAN [23, 4] can reconstruct images with more information and size in parity with n_1, from a 1D vector as input to the network with a size of $n_g = 100$ variables ($n_g/n_1 \simeq 0.5\%$).

Therefore, to reduce the number of parameters in the FC part, an FC layer is added, with the change that the first FC samples down input $n_0 \rightarrow 100 - 200$ nodes and the second FC layer samples up to the desired number of nodes in the CNN part. This method reduces the parameters in the FC part from about 35M to about 2-4M depending on the number of nodes in the sampling step and the number of kernels of the input in the CNN part.

The decoding part of the architecture is inspired by [4] which used [17] architecture. The generator in [17] reconstructs the image from a 1D vector, uses transposed 2D convolution to upsample a 2D image to the desired size. The output from the encoding part of the network is reshaped to $(21, 4, k)$ where transposed 2D convolution layers upsample the data to $(340, 64)$. All filter kernels have size 3×3, stride $(2, 2)$ and batch normalization and ReLu activation after each convolutional layer except the last layer. The number of filter kernels are decreased to half the number from the previous layer, except for the last, which have one kernel, as seen in Figure 6.

9. Results

Two different datasets were generated, both containing up to 16 Raman spectra in one image but with the difference that in the first set all spectra were used in both the training and validation set (see Section 9.2) while in the second set only half of the spectra were used during the training i.e. during the validation unknown spectra could appear (see Section 9.3). Each dataset is built up from 4 600 hyperspectral images resulting in 260 000 rows in the test set and 33 000 rows in validation set.

9.1. Data Generation

When generating the target surface, a number of rectangles and ellipses were randomly positioned as targets and placed on the target surface. See Figure 7 for an example of a target surface with the different targets placed and simulated measurement. The targets were then assigned a Raman spectra. Raman spectra were randomized by letting a spectrum consist of three to six Raman peaks with Raman shifts in the range $300 - 700$ cm^{-1} and thereafter a white noise background was added. Later in the simulated measurement, a white noise measurement was also added. Figure 7 shows examples of Raman spectra generated with background noise added.

9.2. Training and reconstruction from limited Raman spectra dataset

In the first test, only 16 different Raman spectra were used in both the test and validation data sets. The background noise in the hyperspectral image was 0.5% and 1% in the simulated measurement. In the validation set, all Raman spectra were reconstructed correctly, with relatively correct magnitude and correct spatial position, except when the SNR became too low. The model was able to reconstruct Raman spectra, which were visible in a single spatial pixel, but the reconstruction was smeared to the neighboring pixels. The model was also able to reduce the additive noise. Magnitude picture, correlation of spectrum per spatial pixel and individual spectra are seen in Figure 8.

When the input contained a Raman spectra that had not been included in the test data, two reconstruction errors could occur, if a peak in the unknown spectra shared peak

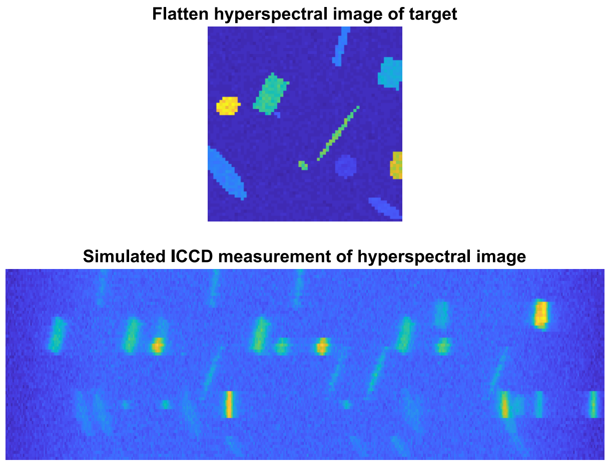

Figure 7. Examples of targets placed on the target surface (upper) and simulated ICCD measurement of the target surface (lower).

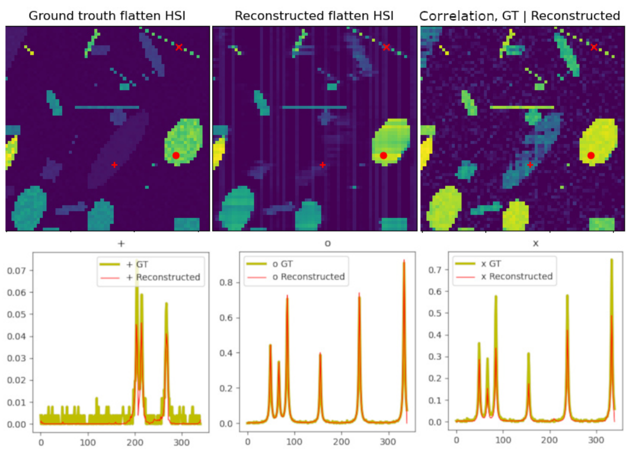

Figure 8. Flattened images of ground truth, reconstructed image and correlation between the two (upper part). Raman spectra at three different positions (lower part). For the dataset where all spectra were used during training.

with one of the known spectra, that part of the spectra was reconstructed but nothing more, i.e. a partial reconstruction of the part of the spectra the model knew. The second reconstruction error that could occur was that a peak in the unknown spectra roughly shared one or more peaks with a known spectra, then the whole substance was reconstructed as the nearest known spectra, i.e. a misclassification.

9.3. Training and reconstruction from broad Raman spectra dataset

	Counts	Distribution
PRS	16	25%
SRS	128	25%
RRS	-	50%

Table 1. Data distribution

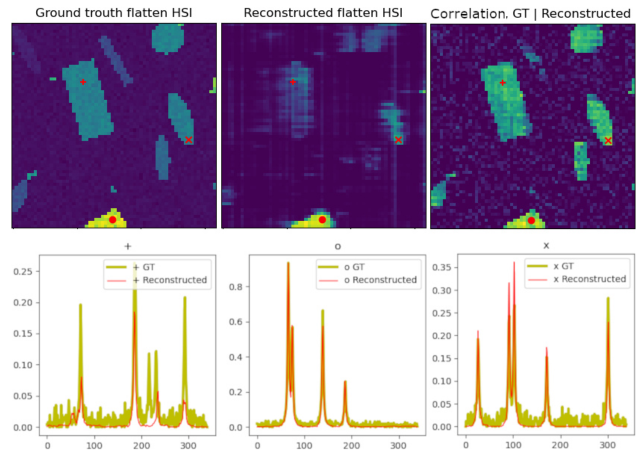

Figure 9. Flattened images of ground truth, reconstructed image and correlation between the two (upper part). Raman spectra at three different positions (lower part). For the dataset where half of the spectra were not used during training.

The second test consisted of 16 different primary Raman spectra (PRS), where four of these were used in each image, 128 different secondary Raman spectra (SRS) where four of them were in each image and eight completely random Raman spectra (RRS) in each image (Note possible to be included in an image), see Table 1. This meant that in the validation set, half of the spectra were not included in the training set (the network has never seen the Raman spectra). Both the background noise and the measurement noise were increased to 1% and 3.5%, respectively, which is considered a more realistic and more difficult noise model for the network to train on. This set was constructed partly to train a more robust and general reconstruction, but also to analyze the impact of reconstructing completely unknown spectrum but still maintain focus on ”important spectra” (PRS). Also note that many of the spectra in the test data with high probability will only appear for the model once or a few times.

Like the first test, this model was able to reconstruct the PRS given an adequate SNR. SRS was reconstructed either completely or partially, while RRS was more likely to be reconstructed only partially or not at all. In a small part from the validation dataset, spectra from RRS with low SNR, peaks or pairs of peaks that did not match the reference data appeared in the reconstruction, unlike the first test these incorrect peaks were relatively close to the reference peak and it is unclear how large impact the relatively large noise can generate ”ghost peaks” in reconstructed spectra.

In Figure 9, where an image from the validation set has been reconstructed, only two substances contain known spectra from PRS and SRS (noted with ”o” and ”x”) and the rest from RRS (not included in the training). In the flatten image of the reconstructed cube, it can be seen that RRS substances are suppressed, while the correlation image shows that partially correct reconstruction is achieved.

10. Conclusions

We have demonstrated how to use deep learning to reconstruct hyperspectral Raman cubes from simulated CASSI measurements. By using DL instead of conventional reconstruction algorithms the speed of the reconstruction has increased several orders of magnitudes (< 1 s for DL vs. > 20 min for conventional) making it possible to do real time measurements with CASSI Raman to cover large surfaces in reasonable time. The results show that it is possible to use a relative small and thereby fast trained convolutional neural network to reconstruct images with good spectral and spatial resolution and precision. By increasing the number of Raman spectra in the training we have showed that the network is capable of a more generalized reconstruction and is more robust against misclassified spectra.

References

[1] G. R. Arce, D. J. Brady, L. Carin, H. Arguello, and D. S. Kittle. Compressive coded aperture spectral imaging: An introduction. *IEEE Signal Processing Magazine*, 31(1):105–115, 2014.

[2] J-F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition–modeling, algorithms, and parameter selection. *International Journal of Computer Vision*, pages 111–136, 2006.

[3] J. M. Bioucas-Dias and M. A. T. Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, 2007.

[4] A. Bora, A. Jalal, E. Price, and A. Dimakis. Compressed sensing using generative models. In *ICML*, 2017.

[5] A. Brorsson, C. Brännlund, D. Bergström, and D. Gustafsson. Compressed imaging at long range in swir. In Michael Felsberg, Per-Erik Forssén, Ida-Maria Sintorn, and Jonas Unger, editors, *Image Analysis*, pages 115–127, Cham, 2019. Springer International Publishing.

[6] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[7] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008.

[8] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597, 2007.

[9] D. Gedalin, Y. Oiknine, and A. Stern. Deepcubenet: reconstruction of spectrally compressive sensed hyperspectral images with deep neural networks. *Opt. Express*, 27(24):35811–35822, Nov 2019.

[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):14013–14027, Oct 2007.

[11] F. C. Higham, R. Murray-Smith, M. J. Padgett, and M. P. Edgar. Deep learning for real-time single-pixel video. *Scientific Reports*, 2369(8), 2018.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[13] D. Kittle, K. Choi, A. Wagadarikar, and D. J. Brady. Multiframe image estimation for coded aperture snapshot spectral imagers. *Appl. Opt.*, 49(36):6824–6833, 2010.

[14] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] M. Nordberg and L. Landström. Uv raman chemical imaging using compressed sensing. *Analyst*, 144:1513–1518, 2019.

[16] H. Östmark, M. Nordberg, and T. E. Carlsson. Stand-off detection of explosives particles by multispectral imaging raman spectroscopy. *Appl. Opt.*, 50(28):5592–5599, Oct 2011.

[17] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2016.

[18] C. V. Raman and K. S. Krishnan. A new type of secondary radiation. *Nature*, 121:501–502, 1928.

[19] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[20] Y. Sun, J. Chen, Q. Liu, and G. Liu. Learning image compressed sensing with sub-pixel convolutional generative adversarial network. *Pattern Recognition*, 98:107051, 2020.

[21] A. Wagadarikar, R. John, R. Willett, and D. Brady. Single disperser design for coded aperture snapshot spectral imaging. *Appl. Opt.*, 47(10):B44–B51, Apr 2008.

[22] L. Wang, T. Zhang, Y. Fu, and H. Huang. Hyperreconnet: Joint coded aperture optimization and image reconstruction for compressive hyperspectral imaging. *IEEE Transactions on Image Processing*, 28(5):2257–2270, May 2019.

[23] Y. Wu, M. Rosca, and T. Lillicrap. Deep compressed sensing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6850–6860. PMLR, 09–15 Jun 2019.