SAND2019-9907C

Deep neural networks for compressive hyperspectral imaging

Dennis J. Lee

Sandia National Laboratories, 1515 Eubank Boulevard, Albuquerque, New Mexico 87123

ABSTRACT

We investigate deep neural networks to reconstruct and classify hyperspectral images from compressive sensing measurements. Hyperspectral sensors provide detailed spectral information to differentiate materials. However, traditional imagers require scanning to acquire spatial and spectral information, which increases collection time. Compressive sensing is a technique to encode signals into fewer measurements. It can speed acquisition time, but the reconstruction can be computationally intensive. First we describe multilayer perceptrons to reconstruct compressive hyperspectral images. Then we compare two different inputs to machine learning classifiers: compressive sensing measurements and the reconstructed hyperspectral image. The classifiers include support vector machines, K nearest neighbors, and three neural networks (3D convolutional neural networks and recurrent neural networks). The results show that deep neural networks can speed up the time for the acquisition, reconstruction, and classification of compressive hyperspectral images.

Keywords: Statistics, machine learning, hyperspectral imaging, compressive sensing

1. BACKGROUND: COMPRESSIVE HYPERSPECTRAL IMAGING

Compressive hyperspectral imaging reduces measurements

Hyperspectral imagers typically require scanning across a scene to collect spectral measurements. A snapshot hyperspectral imager reduces the number of measurements and avoids the need for spatial scanning. It requires reconstruction algorithms to recover the hyperspectral image from measurements. The reconstruction becomes an underdetermined inverse problem: the number of measurements is less than the number of bands. Regularization techniques such as total variation minimization help to make the problem less ill-posed by adding physical constraints such as smoothness along the spatial and spectral dimensions.[1] However, these compressive sensing algorithms demand heavy computation,[1–5] while greedy algorithms may converge to local minima.[6,7] In this work, we explore neural networks for two tasks:

1. Reconstruct the hyperspectral image from compressed measurements.
2. Classify the hyperspectral image from compressed measurements.

For Task 2, we compare two inputs for classification: the compressed measurements, and the reconstructed hyperspectral image from the compressed measurements. These tasks address our research questions:

1. Can neural networks reduce computation time for reconstruction?
2. Can neural networks improve classification accuracy for hyperspectral images, either from compressed measurements or reconstructed spectra, compared with traditional classifiers like support vector machines?
3. How does classification performance compare using compressed measurements as input, versus using reconstructed spectra as input?

Filter light by varying spectral transmissions

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

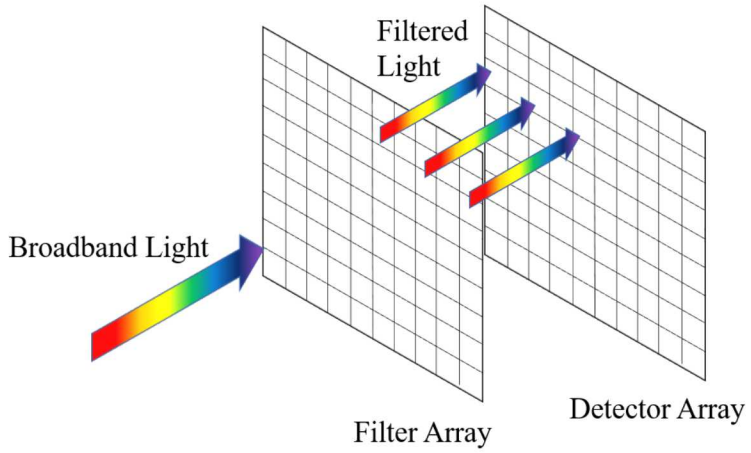


Figure 1: The proposed compressive hyperspectral imager filters light before the detector array.[1] Each measurement corresponds to a different spectral transmission on the filter array.

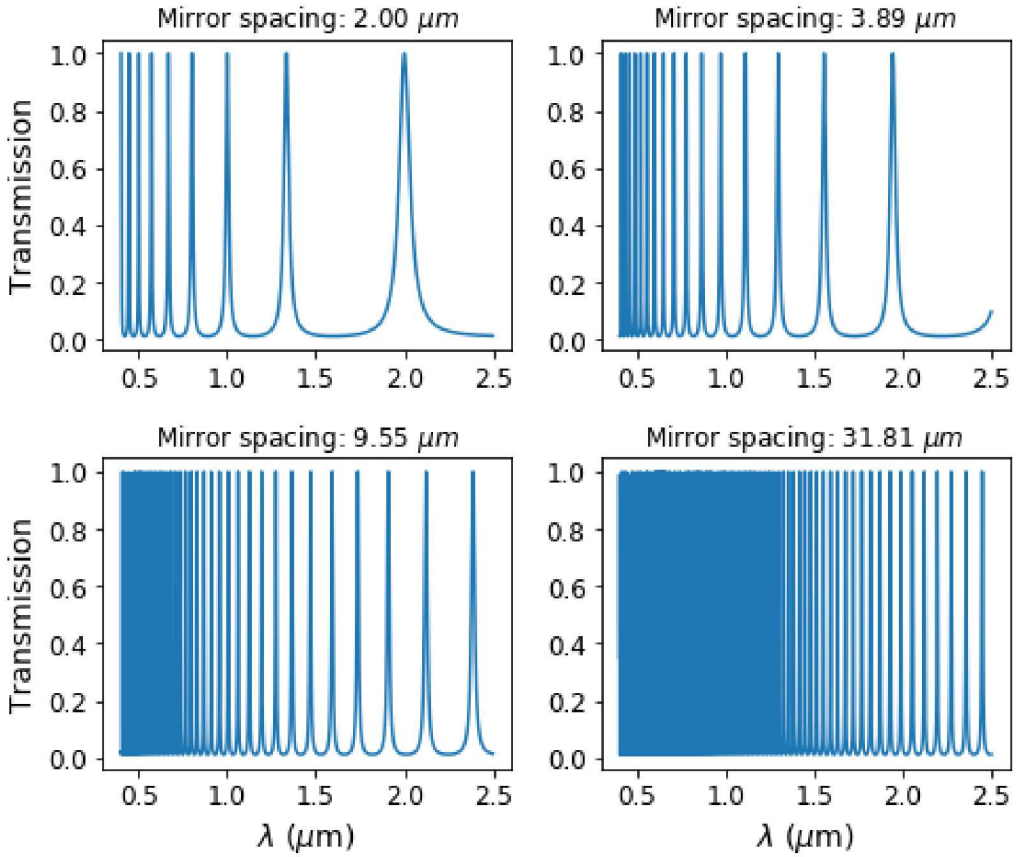


Figure 2: Examples of spectral transmissions of a Fabry-Perot resonator. The spectral transmission depends the mirror spacing in the resonator.

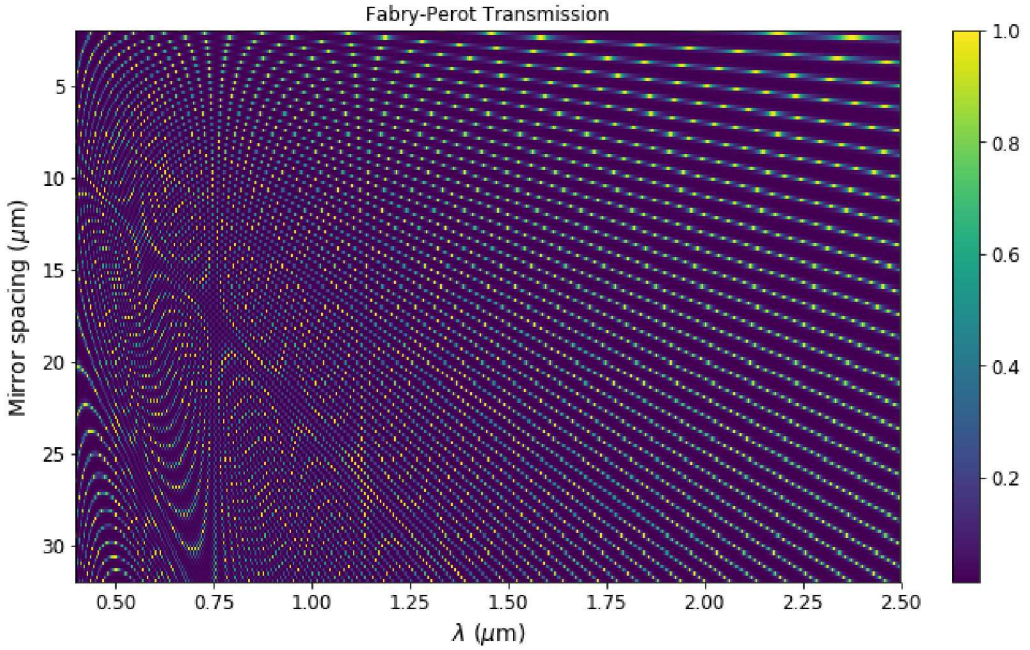


Figure 3: Spectral transmission as a function of mirror spacing and wavelength. The number of measurements equals the number of mirror spacings.

The proposed compressive hyperspectral imager filters light before the detector array, as shown in Fig. 1. Each measurement corresponds to a different spectral transmission on the filter array. Examples of filter arrays include spatial light modulators (liquid crystal displays) and Fabry-Perot resonators, where spectral transmittance varies by applying a digital signal or voltage. This paper will consider Fabry-Perot resonators, in which mirror spacing can vary to control the spectral transmission. Figure 2 shows examples of spectral transmissions at four different mirror spacings. Figure 3 displays the spectral transmittance with mirror spacings ranging from 2 μm to 32 μm over wavelengths from 0.4 μm to 2.5 μm. *Compressive hyperspectral imagery* is hyperspectral imagery collected from compressive sensing measurements. We refer to Lee[1] for more details on the proposed compressive hyperspectral imager.

2. EXPERIMENTS

2.1 Dataset

The Indian Pines dataset is a hyperspectral image of 145 x 145 pixels with a 20 m spatial resolution and 10 nm spectral resolution over the range of 400–2500 nm, divided into 220 bands.[8] The filter array of the hyperspectral imager modulates all of the 220 bands, including the water absorption region. Each pixel label corresponds to a farm crop as shown in Fig. 4. Note that each crop exhibits intraclass variation.

2.2 Task 1: Reconstruction of compressive hyperspectral images

This task aims to reconstruct compressive hyperspectral images using neural networks, which will help reduce computation time with fast inference.

Data augmentation with random spectra and additive noise helps to prevent overfitting

Task 1 draws from two datasets:

Further author information: (Send correspondence to D.J.L.)
D.J.L.: E-mail: dlee1@sandia.gov

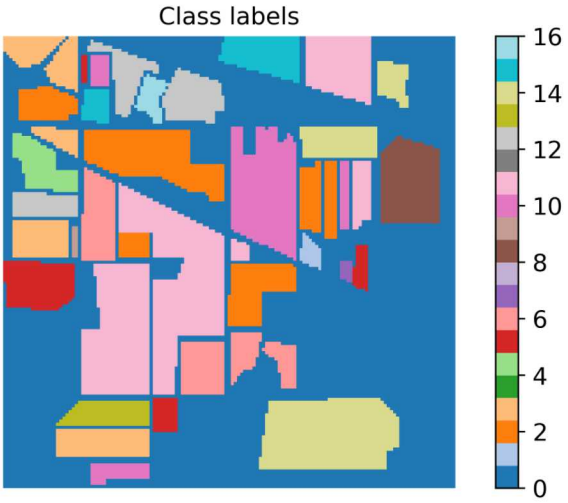


Figure 4: Labels of the Indian Pines hyperspectral image. Each label corresponds to a type of farm crop.

1. Indian Pines dataset
2. Random spectra dataset

For the second dataset, we generate random spectra with 220 bands from a normal distribution with zero mean and unit variance. Then we apply a Hanning filter with window sizes that vary from 11, 21, 31, and 41. This random dataset contains $145 \times 145 = 21025$ spectra, the same number as the Indian Pines dataset.

For each dataset, we reserve 60% for training, 20% for validation, and 20% for testing, maintaining the class imbalance between each split. Note that each split contains equal amounts of each dataset.

For the training split, we add random noise to the Indian Pines dataset, generated from the saeme process as the second dataset. The additive noise and the second dataset help to prevent overfitting, so that the neural network does not memorize the training data. We normalize each dataset to zero mean and unit variance.

Multilayer perceptrons reconstruct compressive hyperspectral images

In our experiments, the compressed signal length varies from 160, 80, 40, 20, to 10 measurements. The goal is to reconstruct 220 bands, so the number of measurements is less than the length of the original spectrum.

We investigate multilayer perceptrons to reconstruct compressive hyperspectral images. The input layer maps the compressed input signal to 220 outputs, with ReLU activation. Then K hidden, fully-connected layers follow, each with ReLU activation. The last layer is linear and outputs the reconstructed spectrum. Applications in video compression have utilized multilayer perceptrons for reconstructing compressive sensing measurements.[9]

In our experiments, we vary the number of hidden layers: $K = 1, 2, 4, 7, 14$. The multilayer perceptron iterates over each pixel to reconstruct the entire hyperspectral image.

2.3 Task 2: Classification of compressive hyperspectral images

Evaluate classifier performance on compressed inputs

This task aims to answer the questions

- Can neural networks improve classification accuracy for hyperspectral images, either from compressed measurements or reconstructed spectra, compared with traditional classifiers like support vector machines?

- How does classification performance compare using compressed measurements as input, versus using reconstructed spectra as input?

We evaluate a variety of classifiers: support vector machines, K nearest neighbors, and three different neural networks, as described below. The input size varies from 160, 80, 40, 20, to 10 measurements, which are less than the full size of 220 bands.

3D convolutions extract spatial and spectral features

We consider two different 3D convolutional neural networks (CNNs). The first 3D CNN extracts blocks of size $5 \times 5 \times I$, where I is the input size, which varies from 220, 160, 80, 40, 20, to 10 in our experiments. Then two convolutional layers follow, with sizes of $3 \times 3 \times 7$ and $3 \times 3 \times 3$, respectively, each followed by ReLU. Next a fully connected layer reads a flattened feature vector and outputs a feature vector with the classification scores for the 16 classes. We train the model with Adagrad and cross-entropy loss. We refer to Li[10] for more details.

The second model is a multiscale 3D CNN. The input patch has size $7 \times 7 \times I$, where I is defined above. It passes through a convolutional layer with 16 $3 \times 3 \times 11$ kernels, followed by ReLU. A special layer consists of 16 parallel 3D convolution blocks, each of sizes $1 \times 1 \times 1$, $1 \times 1 \times 3$, $1 \times 1 \times 5$, and $1 \times 1 \times 11$, which are summed and passed through ReLU. A second, identical parallel layer follows. Next is a convolutional layer with 16 kernels of size $2 \times 2 \times 3$, followed by ReLU, $2 \times 2 \times 3$ pooling, and dropout. A fully connected layer outputs a score for each class. We refer to He[11] for more details.

Previous approaches have applied principle components analysis to the spectral dimension, independently of the spatial dimension. Performing convolutions in 3D can simultaneously extract spatial and spectral features.

Recurrent networks characterize spectral correlations

The input to the recurrent neural network (RNN) is a hyperspectral pixel. The recurrent layer reads one band, while the next band is input simulataneously. The RNN predicts the label of the pixel after looping through the entire hyperspectral pixel sequence. We consider a model with gated recurrent units of size 64, followed by batch norm and tanh activation. A final fully connected layer outputs scores for each class. Note that this model considers spectral correlations but not spatial relations between neighboring pixels. We refer to Mou[12] for more details.

3. RESULTS

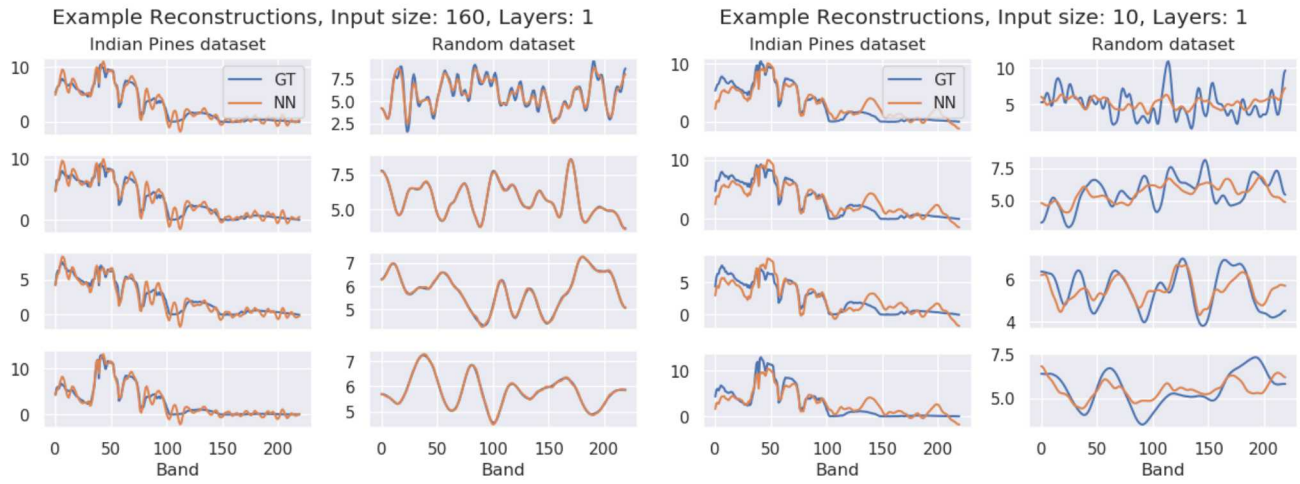
3.1 Task 1: Reconstruction of compressive hyperspectral images

Moderate compression (160 / 220) results in small reconstruction error

For Task 1, we study the reconstruction error as the number of measurements decreases from 160, 80, 40, 20, to 10, and the reconstructed spectra has 220 bands. The error also depends on the number of layers in the multilayer perceptron. Figure 5a shows examples of reconstructions with 160 measurements using a single layer perceptron. Both the Indian Pines and random spectra resemble ground truth closely at a moderate compression ratio (160 measurements of 220 bands).

Larger compression (10 / 220) results in larger reconstruction error

Figure 5b shows examples of reconstructions with 10 measurements using a single layer perceptron. The error increases in both datasets: the random spectra shows the most error in the higher frequencies, and the Indian Pines dataset shows some bias in bands 0–30 and artifacts near the water absorption region near band 130. As expected, a larger compression (10 measurements of 220 bands) results in larger reconstruction errors.



(a) Reconstruction from 160 measurements.

(b) Reconstruction from 10 measurements.

Figure 5: Example reconstructions from varying numbers of measurements with a single layer perceptron.

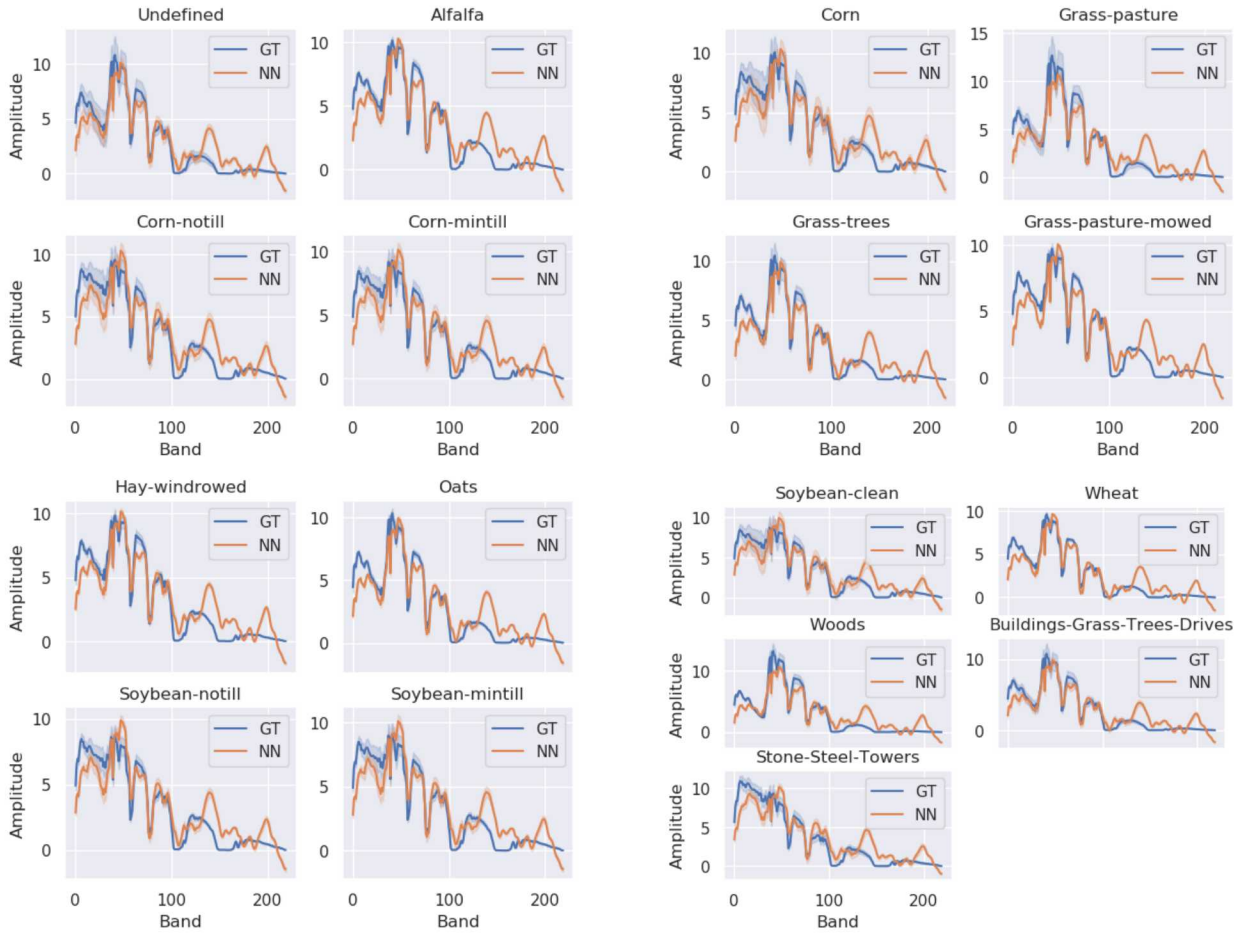


Figure 6: Reconstructions of each class/crop from 10 measurements using a single layer perceptron, after running 2000 epochs. The plots show the average spectra with confidence intervals of one standard deviation, which shows intraclass variation. GT: Ground Truth. NN: Single layer perceptron.

Figure 6 shows the reconstruction distribution of the Indian Pines classes from 10 measurements using a single layer perceptron, after running 2000 epochs. The plots show the average spectra of each class with confidence intervals of one standard deviation, which indicates intraclass variation in reconstruction error.

Overall R^2 shows too many layers increases error

Figures 7a and 7b show the overall R^2 with standard deviation as a measure of similarity of the reconstruction with the ground truth. The heatmap illustrates the reconstruction quality as the number of measurements and the number of layers vary. Note the training, validation, and testing sets remain fixed, and the standard deviation measures the variation in error over the entire testing set. If the number of layers is too large (e.g. 14 layers), the model begins to memorize the waveforms in the training set and does not generalize well to the testing set. Consequently, the overall R^2 degrades compared to models with fewer layers.

Single layer shows least overfitting

Figures 7c and 7d show R^2 with standard deviation over the Indian Pines dataset, excluding the random spectra dataset. Figures 7e and 7f show R^2 with standard deviation over the random spectra dataset, excluding the Indian Pines dataset. A high R^2 value in one dataset may indicate overfitting if the other dataset has a corresponding low R^2. For example, when the number of layers is 2, 4, and 7, the Indian Pines R^2 is relatively high for 10 measurements, but the corresponding R^2 over the random spectra dataset is much lower. We find that the single layer perceptron shows the least overfitting based on comparing R^2 values between the two datasets.

Regularize the model to further reduce overfitting

Regularization may help to reduce overfitting on the Indian Pines dataset. For example, dropout may be added to the multilayer perceptron. Other random spectra or additive noise may further augment the training dataset. Comparing the reconstruction of the random spectra with the Indian Pines dataset helps to measure how much the model overfits one dataset compared to the other.

3.2 Task 2: Classification of compressive hyperspectral images

Recurrent networks perform best on compressed inputs

We split the Indian Pines dataset into training, validation, and testing according to 60/20/20 proportions. All the experiments below use the same split throughout. The standard deviation in the accuracies reflect the variance caused dropout in the neural networks.

Figure 8 shows the overall accuracy with varying input size from 220, 160, 80, 40, 20, to 10 measurements. Note the number of measurements is less than or equal to the number of bands B in the hyperspectral image ($B = 220$ corresponds to the full or uncompressed spectral input). The heatmap compares different classifiers: support vector machines (SVM), K nearest neighbors (KNN), and three types of neural networks as described in Section 2.3. For SVM on compressed inputs, we use a radial basis function (RBF) kernel with $C = 1000$ and $\gamma = 0.001$, as determined by grid search. For KNN, we search over $K = 1, 3, 5, 10, 20$. In most cases, $K = 10$ performed best (e.g. for compressed sizes of 10 and 160).

The recurrent neural network (RNN) from Mou[12] performs the best on compressed inputs. Figure 9 shows the average class accuracies using the RNN from Mou.

3D CNNs perform best on reconstructed spectra

We test classifier performance on reconstructed inputs as a comparison with the compressed inputs. Figure 10 shows the classifiers' accuracies with varying numbers of measurements. The compressed signal length denotes the number of measurements used to reconstruct the hyperspectral images. The number of bands in the reconstructed hyperspectral image is 220, and this reconstructed image is the input to the classifiers. We use the single layer perceptron to calculate the reconstructed spectra, since it shows the least overfitting. Each model runs 20 times, with 100 epochs per run.

The multiscale 3D CNN from He performs the best over reconstructed inputs, based on the comparison from Fig. 10. Figure 11 shows the class accuracies on the reconstructed inputs using the multiscale 3D CNN from He.[11] Note that the values at 220 are empty in the table since the original signal length is 220.

Accuracy improves on reconstructed inputs

Figure 12 compares the accuracy of four different classifiers: support vector machines, K nearest neighbors, the RNN from Mou, and the multiscale 3D CNN from He. Each plot compares two inputs to the classifier: the compressed (non-reconstructed) measurements, and the reconstructed spectra. Note that the reconstructed spectra has full size (220 bands), and the x-axis denotes the number of compressed measurements before reconstruction. The size of 220 corresponds to the full spectral input to the classifier. Each model runs 20 times with 100 epochs per run.

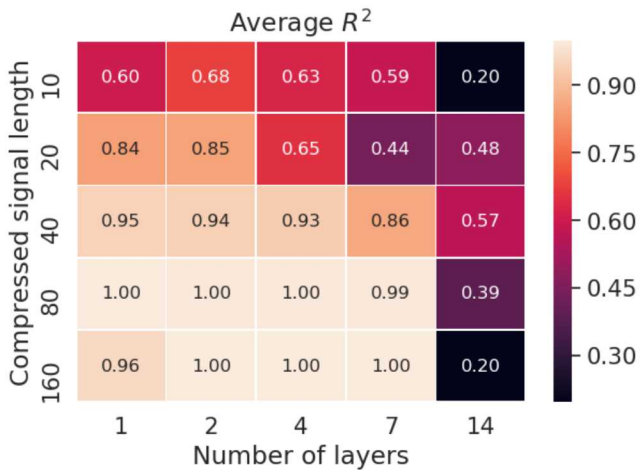
The compressed input may lose spatial context, so 3D convolutions may not be as effective. Neighboring pixels that share the same class may look different in compressed space. Compressive sensing can be an unstable, underdetermined inverse problem, where the number of measurements can be far less than the size of the signal to be reconstructed. As a result, the classifiers that do not account for spatial context perform better on compressed inputs, or the performance is comparable to reconstructed inputs. For example, support vector machines and the RNN from Mou only consider spectral correlations.

Classifiers that consider both spatial and spectral correlations, such as KNN or 3D CNNs, perform better on the reconstructed spectra, which more closely resemble physical signals. These classifiers exhibit a large divergence between reconstructed and compressed inputs, which indicates that the compressed inputs provide poor spatial context for these algorithms. The single layer perceptron from Task 1 produces fairly accurate reconstructions even with large compression ratios, boosting performance on Task 2.

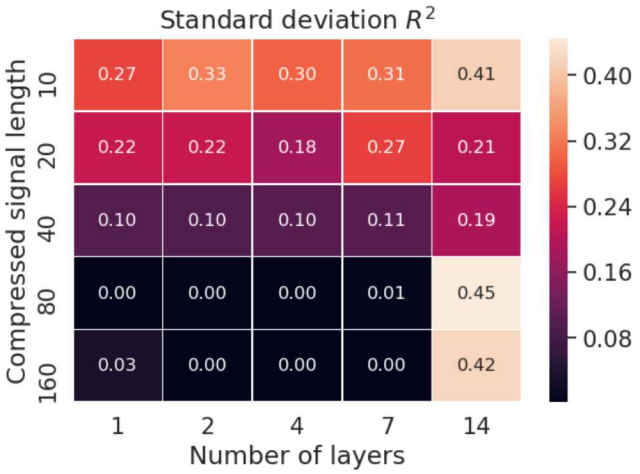
4. SUMMARY

Neural networks can reconstruct and classify compressive hyperspectral images

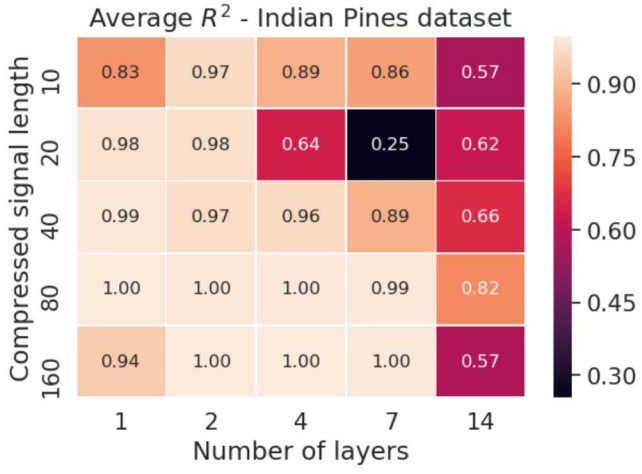
We have demonstrated a two step process for hyperspectral image classification using compressive sensing measurements. The first step is to reconstruct the hyperspectral image from compressive sensing measurements. We investigated varying the number of layers in a multiayer perceptron and found that a single layer minimizes overfitting. The second step is to classify the reconstructed image. We compared support vector machines, K nearest neighbors, and three neural networks (3D CNNs, multiscale 3D CNNs, RNNs). Classifier accuracy improves using reconstructed spectra compared to raw compressed measurements. This work shows how neural networks can reconstruct and classify compressive hyperspectral images, which reduces the number of measurements and speeds acquisition time.



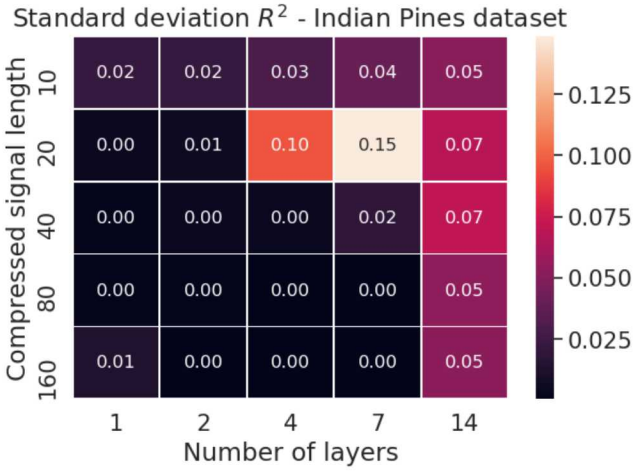
(a) Average R^2 for both datasets (Indian Pines and random spectra).



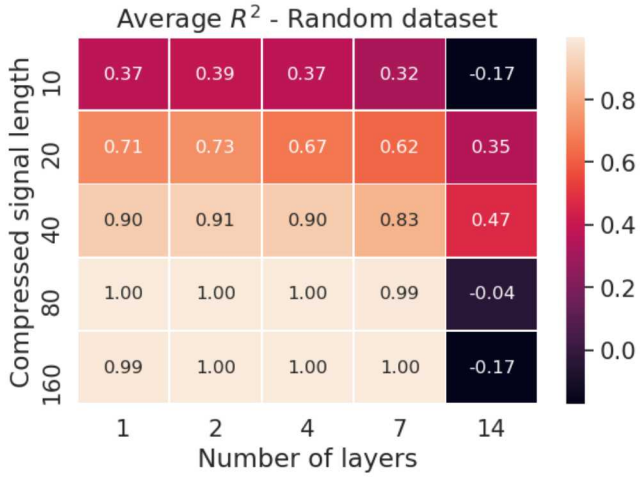
(b) Standard deviation of R^2 for both datasets.



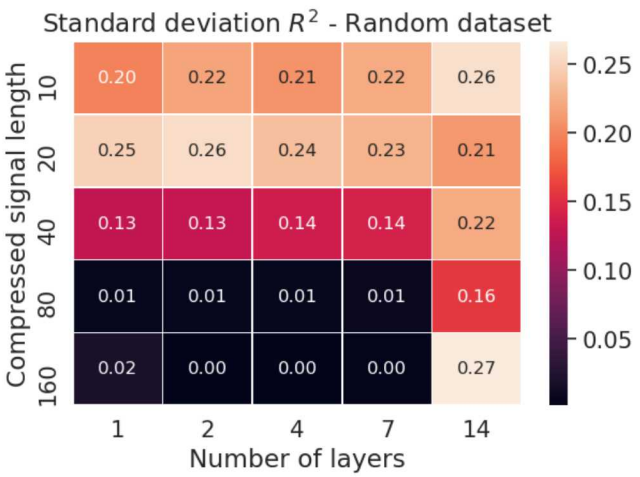
(c) Average R^2 for the Indian Pines dataset.



(d) Standard deviation of R^2 for the Indian Pines dataset.

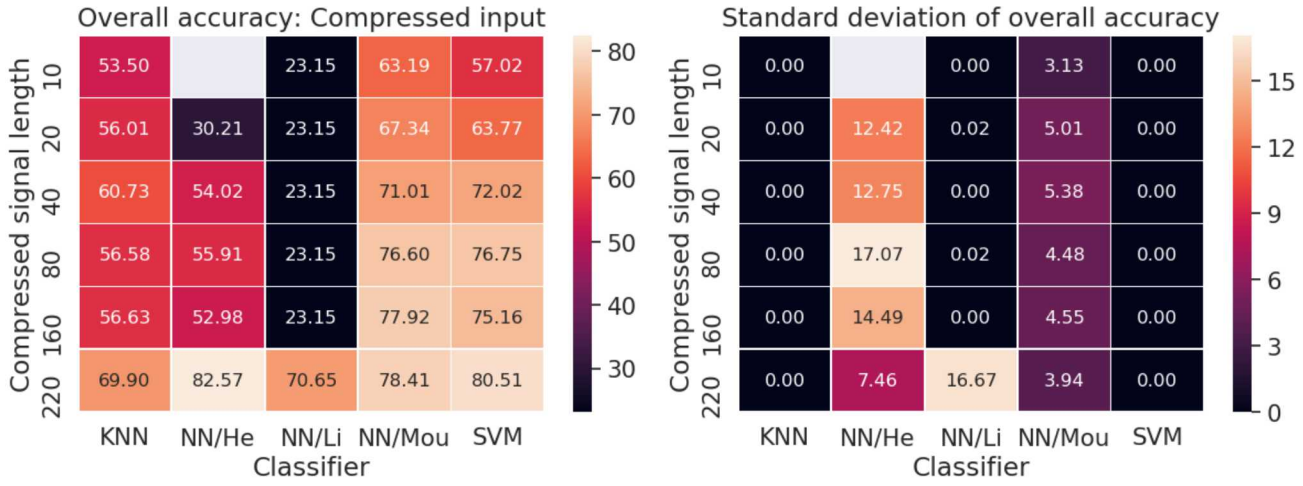


(e) Average R^2 for the random spectra dataset.



(f) Standard deviation of R^2 for the random spectra dataset.

Figure 7: R^2 for reconstructions over two datasets (Indian Pines and random spectra).



(a) Overall accuracy

(b) Standard deviation of the overall accuracy

Figure 8: Comparison of classifier accuracy with varying numbers of measurements (compressed signal length). The inputs to the classifiers are the compressed (non-reconstructed) measurements. The number of bands in the hyperspectral image is 220, and the compression ratio is the number of measurements divided by the number of bands. Note that the 3D CNN from He does not accept inputs of size 10, so these values are empty. Each model runs 20 times, with 100 epochs per run.

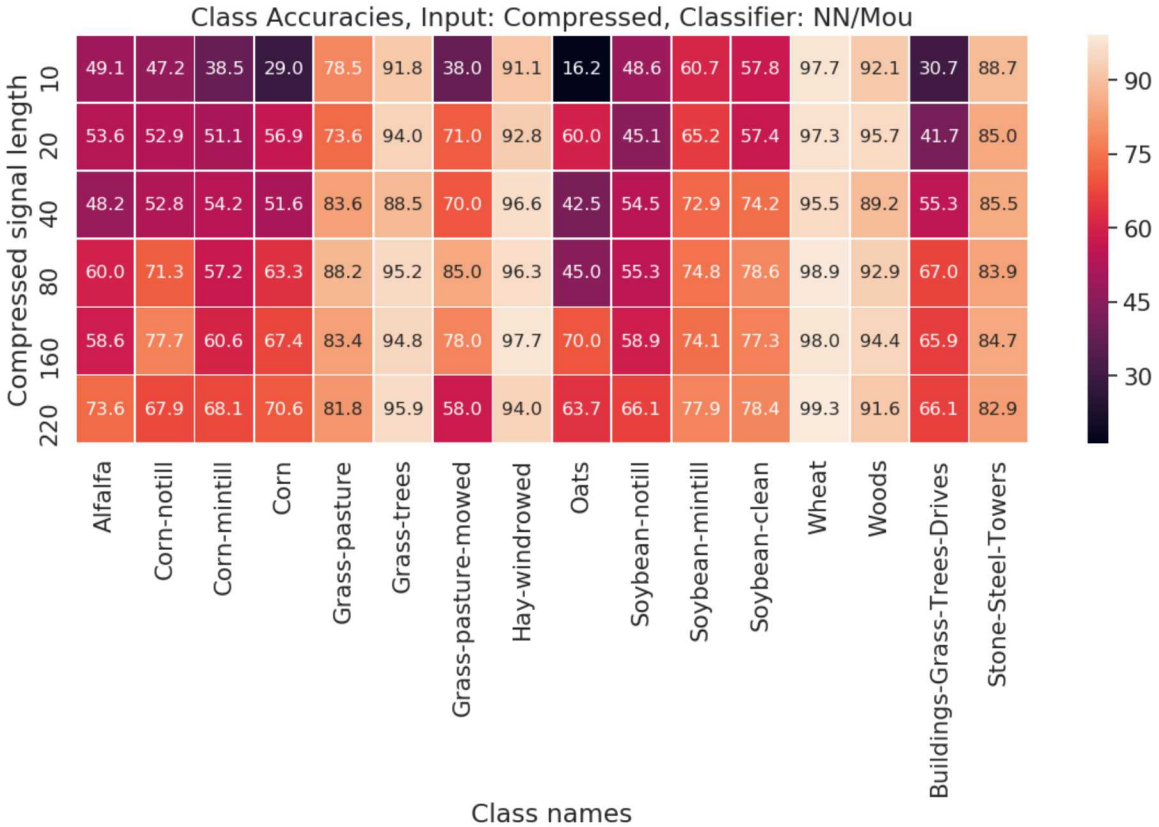
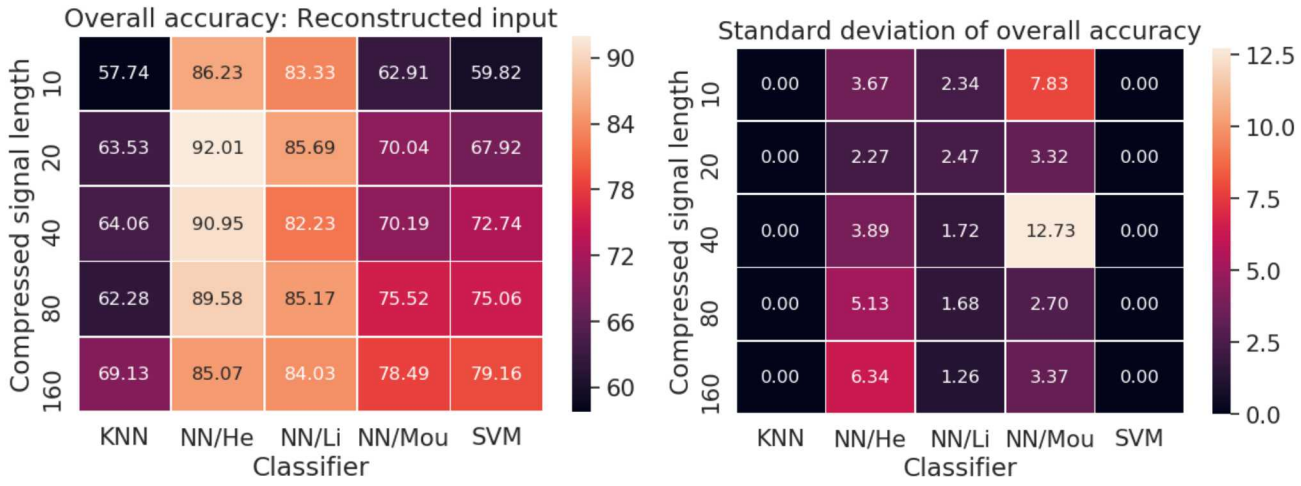


Figure 9: Class accuracies with varying numbers of measurements (compressed signal length) using the recurrent neural network from Mou. The inputs to the classifier are the compressed (non-reconstructed) measurements.



(a) Overall accuracy

(b) Standard deviation of the overall accuracy

Figure 10: Comparison of classifier accuracy with varying numbers of measurements (compressed signal length). The inputs to the classifiers are the reconstructed hyperspectral images. The compressed signal length denotes the number of measurements used to reconstruct the hyperspectral images. The number of bands in the reconstructed hyperspectral image is 220. Each model runs 20 times, with 100 epochs per run.

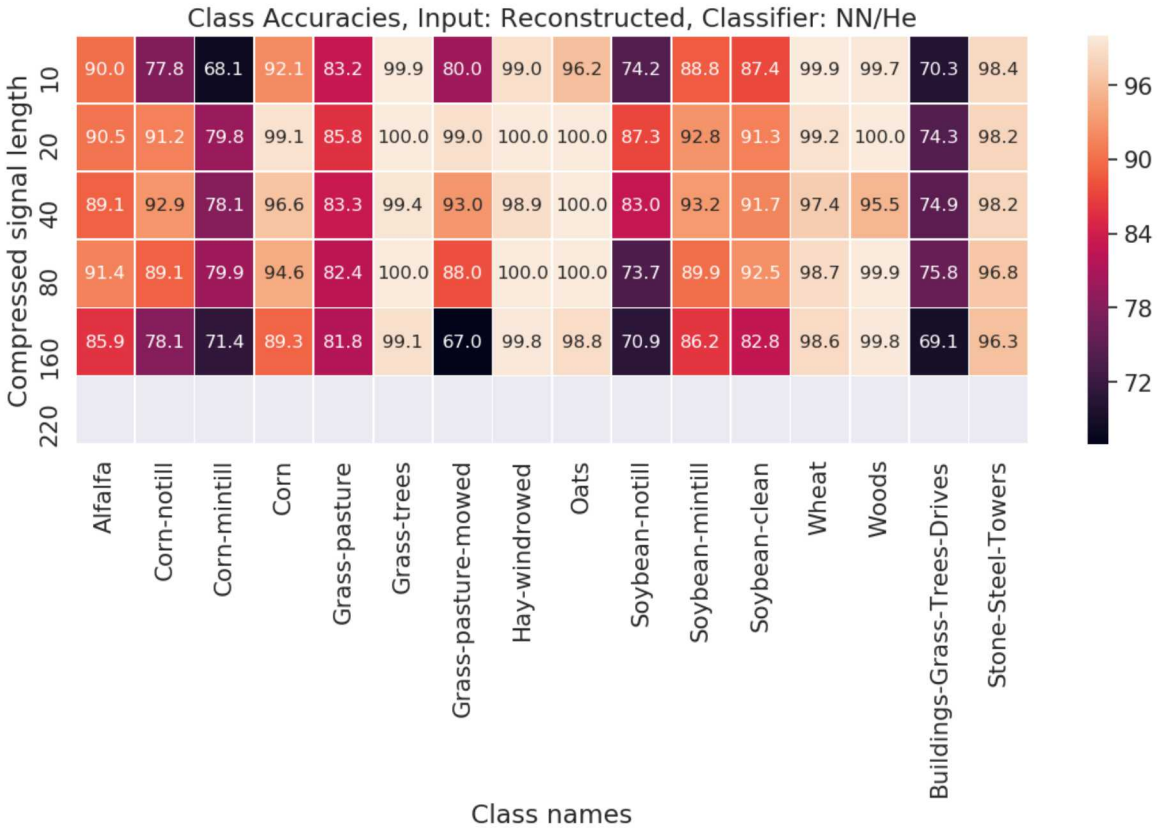
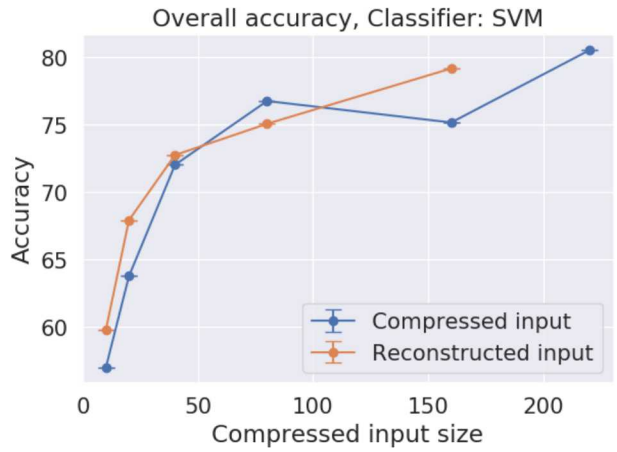
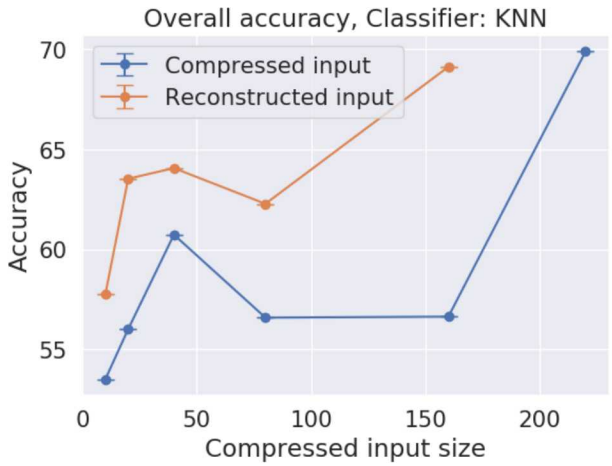


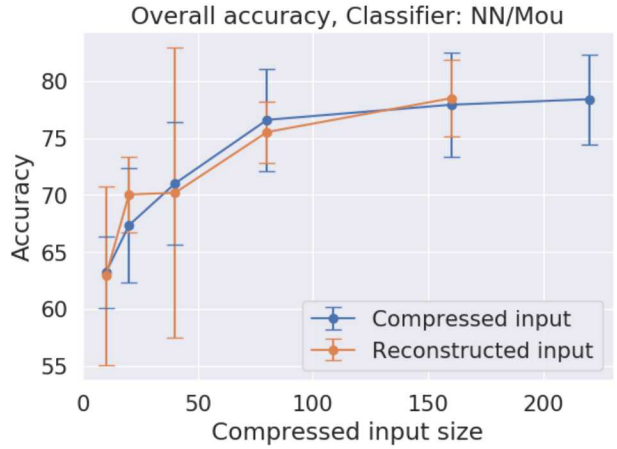
Figure 11: Class accuracies with varying numbers of measurements (compressed signal length) using the multi-scale 3D convolutional neural network from He. The inputs to the classifiers are the reconstructed hyperspectral images. The number of bands in the reconstructed hyperspectral image is 220. The compressed signal length denotes the number of measurements used to reconstruct the hyperspectral images, so the values at 220 are empty in the table since the original signal length is 220.



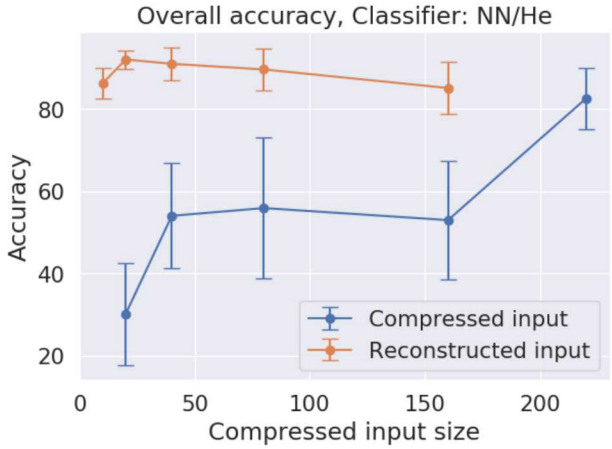
(a) The classifier is the support vector machine. The parameters C and γ are determined by grid search for each input size, and the kernel is the radial basis function. The classifier does not consider spectral similarities between adjacent pixels, and both input types seem to perform equally well.



(b) The classifier is K nearest neighbors. The size K is determined by grid search for each input size. This method does not seem to perform well on compressed measurements, as nearest neighbors may be dissimilar in compressed space.



(c) The classifier is the recurrent neural network from Mou. This method predicts spectral correlations but does not consider the spatial dimension. It seems to perform best on compressed measurements.

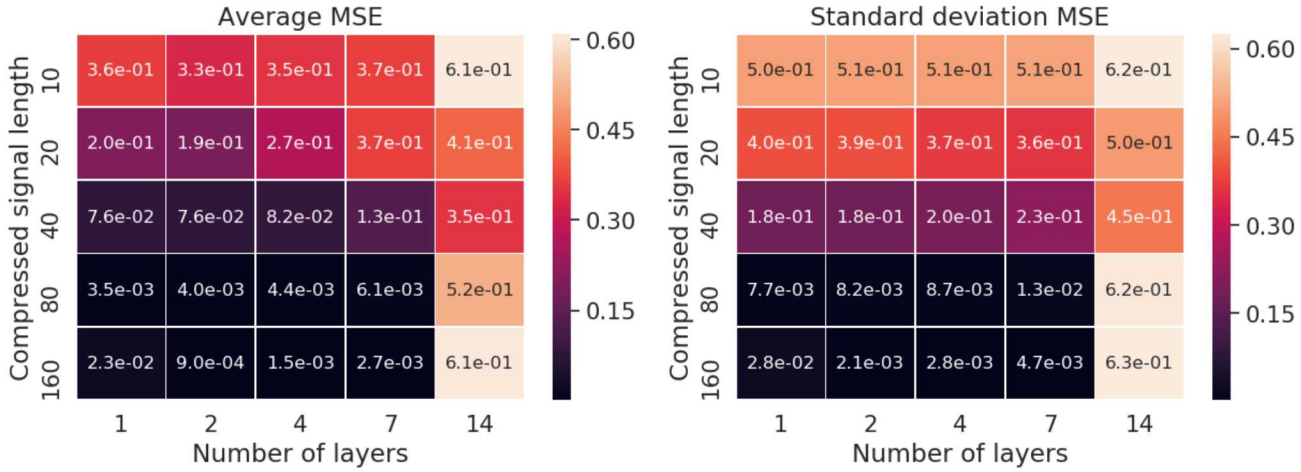


(d) The classifier is the multiscale 3D convolutional neural network. This method does not seem to perform well on compressed measurements, as adjacent pixels may look dissimilar in compressed space, making convolutional approaches less effective. It seems to perform best on reconstructed spectra.

Figure 12: Accuracy comparison of four different classifiers. Each plot compares two inputs to the classifier: the compressed (non-reconstructed) measurements, and the reconstructed spectra. Note that the reconstructed spectra has full size (220 bands), and the x-axis denotes the number of compressed measurements before reconstruction. The size of 220 corresponds to the full spectral input to the classifier. Each model runs 20 times with 100 epochs per run.

APPENDIX A. ADDITIONAL RECONSTRUCTION METRICS

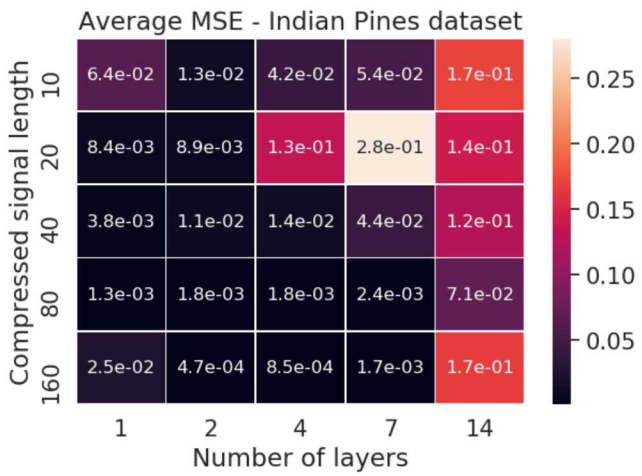
This appendix presents other metrics, in addition to R^2, for reconstructing hyperspectral images from compressive sensing measurements. The mean squared error (MSE) is another way to measure the difference between the reconstructed spectra and ground truth. Figure 13 shows the average MSE for both datasets, while Fig. 14 shows the average MSE for each dataset separately (Indian Pines and random spectra).



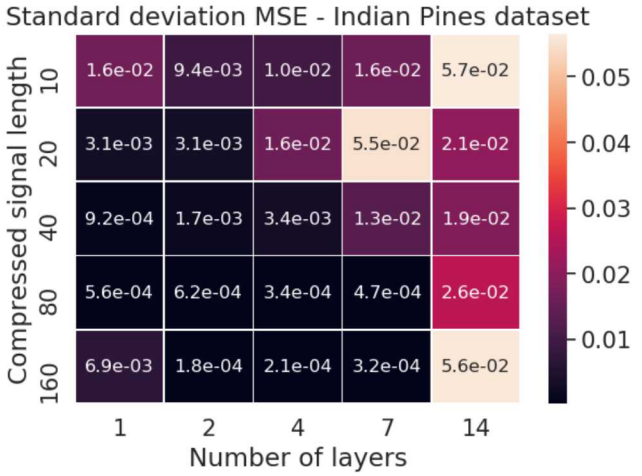
(a) Average MSE over both datasets (Indian Pines and random spectra).

(b) Standard deviation of the MSE for both datasets.

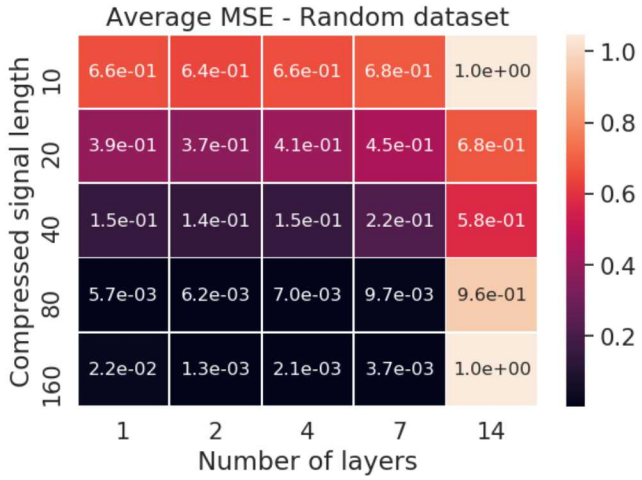
Figure 13: Mean squared error (MSE) in the reconstruction with varying input size and number of layers in the perceptron. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance across all spectra in the testing set.



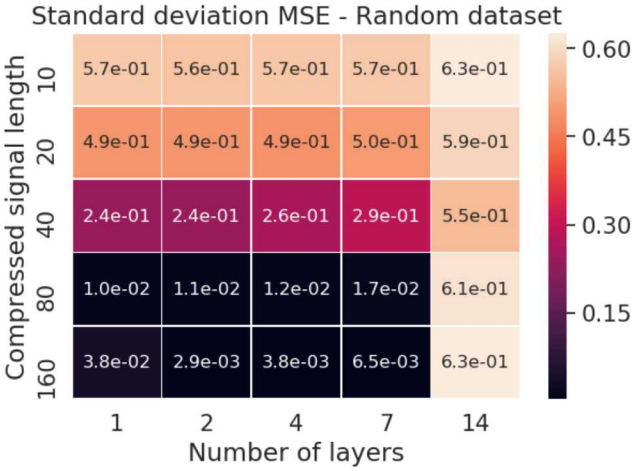
(a) Average MSE for the Indian Pines dataset.



(b) Standard deviation of MSE for the Indian Pines dataset.



(c) Average MSE for the random spectra dataset.

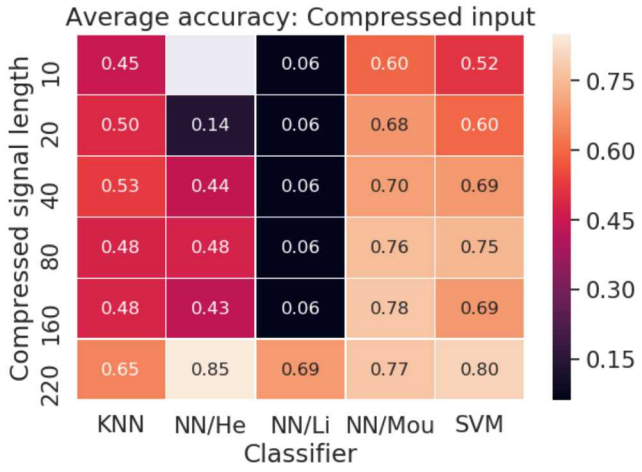


(d) Standard deviation of the MSE for the random spectra dataset.

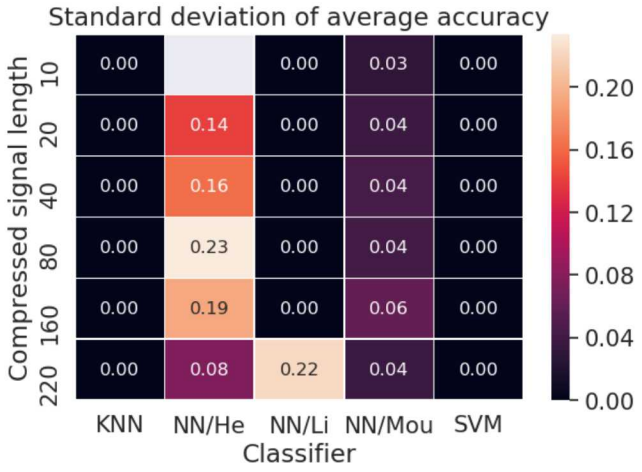
Figure 14: Mean squared error (MSE) for each dataset with varying input size and number of layers in the perceptron. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance across all spectra in the testing set.

APPENDIX B. ADDITIONAL CLASSIFICATION METRICS

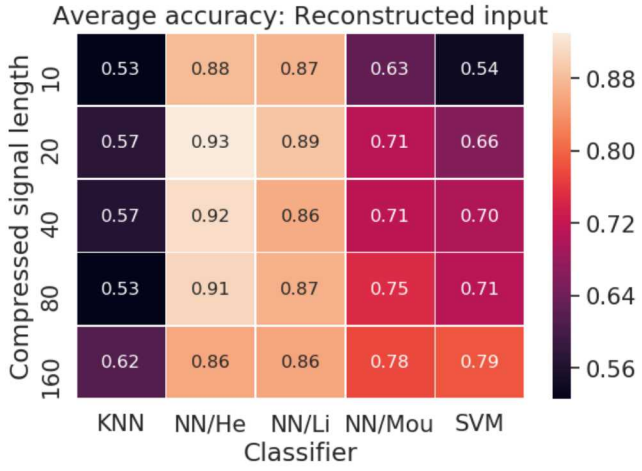
This appendix presents other classification metrics, in addition to overall accuracy, for classifying compressive hyperspectral images. The average accuracy is an average of the class accuracies. Figure 15 shows the average accuracy from two types of inputs: either compressed measurements or reconstructed spectra. Figure 16 shows the average kappa for these two types of inputs. Both figures compare different classifiers over a varying number of measurements.



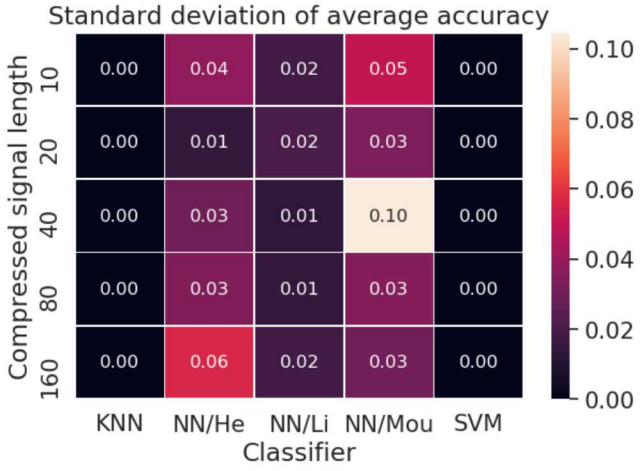
(a) Average accuracy from classifying compressed measurements.



(b) Standard deviation of average accuracy from classifying compressed measurements.

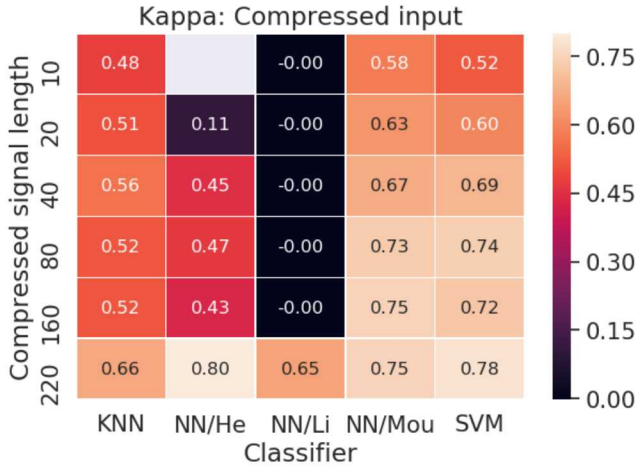


(c) Average accuracy from classifying reconstructed spectra.

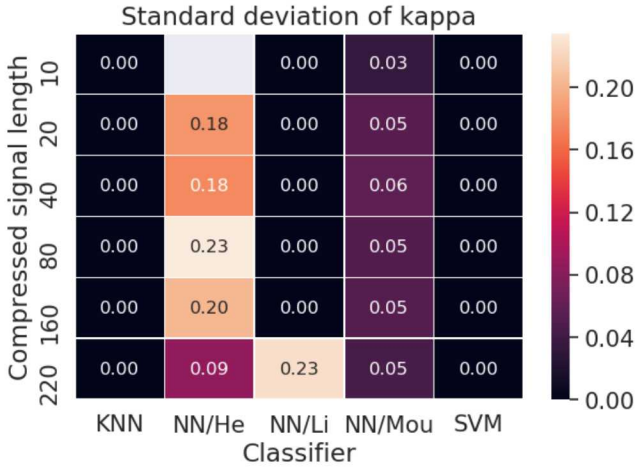


(d) Standard deviation of average accuracy from classifying reconstructed spectra.

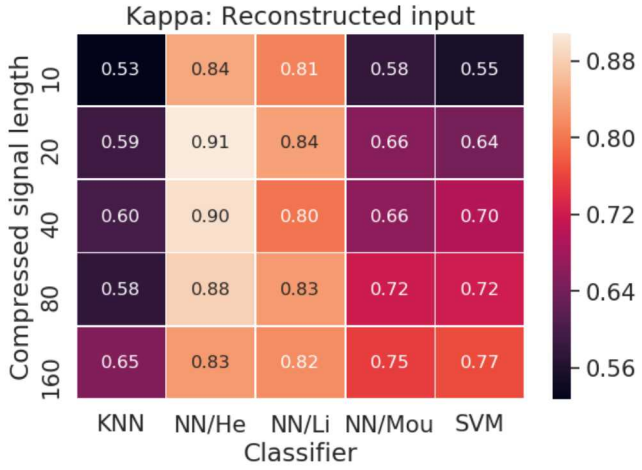
Figure 15: Average accuracy of classifying compressive hyperspectral images. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance of the models that use dropout. Each input (either compressed measurements or reconstructed spectra) compares different classifiers and varying input sizes. The reconstructed spectra have 220 bands, and the input size denotes the number of measurements before reconstruction.



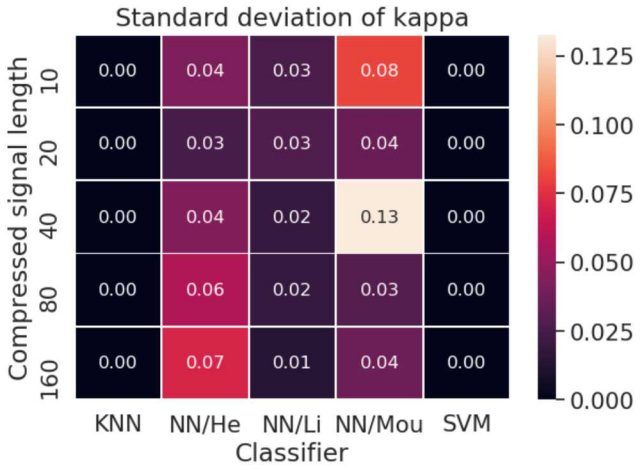
(a) Average kappa from classifying compressed measurements.



(b) Standard deviation of average kappa from classifying compressed measurements.



(c) Average kappa from classifying reconstructed spectra.



(d) Standard deviation of kappa from classifying reconstructed spectra.

Figure 16: Average kappa from classifying compressive hyperspectral images. The training/validation/testing split is fixed at 60/20/20, and the standard deviation measures the variance of the models that use dropout. Each input (either compressed measurements or reconstructed spectra) compares different classifiers and varying input sizes. The reconstructed spectra have 220 bands, and the input size denotes the number of measurements before reconstruction.

ACKNOWLEDGMENTS

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

REFERENCES

[1] D. J. Lee and E. A. Shields, "Compressive hyperspectral imaging using total variation minimization," Proc. SPIE 10768, 1076804 (2018).

[2] D. J. Lee, C. F. LaCasse, and J. M. Craven, "Compressed channeled spectropolarimetry," Opt. Express 25, 32041-32063 (2017).

[3] D. J. Lee, C. F. LaCasse, and J. M. Craven, "Compressed channeled linear imaging polarimetry," Proc. SPIE 10407, 104070D (2017)

[4] D. J. Lee, C. A. Bouman, and A. M. Weiner, "Single Shot Digital Holography Using Iterative Reconstruction with Alternating Updates of Amplitude and Phase," http://www.arxiv.org/abs/1609.02978 (2016).

[5] D. J. Lee, C. F. LaCasse, and J. M. Craven, "Channeled spectropolarimetry using iterative reconstruction," Proc. SPIE 9853, 98530V (2016).

[6] D. J. Lee and A. M. Weiner, "Optical phase imaging using a synthetic aperture phase retrieval technique," Opt. Express **22**(8), 9380-9394 (2014).

[7] D. J. Lee, K. Han, H. J. Lee, and A. M. Weiner, "Synthetic aperture microscopy based on referenceless phase retrieval with an electrically tunable lens," Appl. Opt. **54**(17), 5346-5352 (2015).

[8] M. Baumgardner, L. Biehl, D. Landgrebe. 220 Band AVIRIS Hyperspectral Image Data Set: June 12, 1992 Indian Pine Test Site 3. Purdue University Research Repository. doi:10.4231/R7RX991C (2015).

[9] M. Iliadis, L. Spinoulas, A. Katsaggelos, "Deep fully-connected networks for video compressive sensing," arXiv:1603.04930v2 (2017).

[10] Y. Li, H. Zhang, Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3D convolutional neural network," Remote Sens. 9, 67 (2017)

[11] M. He, B. Li, H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," ICIP (2017).

[12] L. Mou, P. Ghamisi, X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," IEEE Trans. Geos. Remote Sens. 55, 7 (2017).