Robust Spectral Based Compression of Hyperspectral Images using LSTM Autoencoders

Kyle Webster
Gianforte School of Computing
Montana State University
Bozeman, MT 59717
kyle.webster@student.montana.edu

John Sheppard
Gianforte School of Computing
Montana State University
Bozeman, MT 59717
john.sheppard@montana.edu

***Abstract*—The large size of hyperspectral images limits the applicable uses and necessitates effective compression methods. While there has been great success in a combined spectral and spatial compression approach, the reconstruction error rate for lossy compression methods with higher compression rates is still relatively large. Inspired by recent successes in spectral based deep learning for classification, we propose a spectral based Long Short Term Memory Autoencoder (LSTM-AE) to compress the spectral dimension alone. The obtained results show that not only can LSTM-AE achieve similar compression rates to existing methods, but also a large reduction in reconstruction error. We have also demonstrated the robustness of the approach in being able to generalize a single model for use in multiple scenes without being retrained. The model was also demonstrated to be successful in compressing unseen images at higher rates than existing methods that have trained on those images.**

***Index Terms*—Recurrent Neural Networks, Compression, Autoencoder**

I. Introduction

Remote sensing systems have many diverse applications; including food quality and safety [1], land use and land cover classification [2], and crop analysis [3]. Hyperspectral images (HSI) are a specific categorization of remote sensing images that contain many continuous bands that represent a narrow range of captured wavelengths. These images are represented as 3D data cubes with pixels in the x and y axis and the spectral bands in the z axis. HSIs are known to have high levels of spatial and spectral information and can often have high levels of correlation within the bands. These sensors capture a wide array of reflected radiation wavelengths as a three dimensional image. Through the exploitation of spatial and spectral information in these images, powerful classifiers have been built to analyze the images.

Due to the large level of spectral and spatial information that hyperspectral images can capture, HSIs have a tendency to be very large when stored in a database. Their sizes can easily exceed 50 Megabytes (MB) since each pixel stores information in the format of 12 bits per pixel per band (bpppb) or 16 bpppb with bands often exceeding 100 bands. For example, say an image that has a pixel size of 64×64 and each pixel contains 300 bands with each band being stored in 16 bits. The total size of this image would be $(64 \times 64 \times 300 \times 16) = 19.66$ MB.

Given the large size of the images, it is unrealistic to utilize the images without a compression process in both data transfer as well as when stored in a database. Lossless compressing can be very powerful with the guarantee of no information loss. One of the biggest downsides with lossless compression, however, is the reduction in the compression ratio with recent attempts rarely surpassing a compression ratio of 25% of the original size [4]. For specific applications, this might be a worthwhile trade off, but for most general applications, lossy compression algorithms are acceptable due to the significantly higher compression rates, and acceptable loss of information.

There has been little work analyzing a purely spectral based compression strategy due to the increased compression rates of a spectral and spatial strategy. Methods that have traditionally implemented a spectral strategy further compressed by a spatial strategy often suffer from a significant increase in reconstruction error the further compressed an image is [5], [6]. The combined compression of an HSI's spatial and spectral dimensions simultaneously has shown some improvements with regards to the weaknesses of the previous strategies, but they still suffer from the same trends [7], [8].

Recent work in hyperspectral compression techniques demonstrate that a spatial and spectral approach may not be necessary to achieve similar results. In particular, recent work with Long Short Term Memory Networks to learn the spectral information and autoencoders performing feature extraction of hyperspectral images indicate that a combined strategy may be able to represent the bands and extract enough feature information from the bands to allow for a competitive HSI compression strategy. [9]–[12].

In this paper, we propose an autoencoder based compression method to compress the spectral dimension of the hyperspectral image. With a given band's property of being continuous, we can represent the reflectance values in the band as a temporal sequence for use in a Long Short Term Memory Autoencoder network (LSTM-AE) [13]. We also propose a modification of the LSTM-AE network with the Bidirectional Long Short Term Memory Autoencoder network (BLSTM-AE) [14]. Both networks compress each pixel's bands independently, preserving the original image's spatial dimension. We then explore the level of generality that the independence from spatial correlation we ascribe to the bands provides the

LSTM-AE and BLSTM-AE. This is accomplished through a process of training on a subset of pixels from multiple images to train a single model. We also experimented with taking a trained model and compressing an unseen image to determine the robustness of the models.

With modern band selection methods achieving a selection rate as low as 6 bands [15], we hypothesize that our autoencoding method will be able to perform similarly to cutting edge compression methods with respect to the compression rate and reconstruction error. We also hypothesize that by removing the dependence of spatial information with our compression method, we will be able to generalize the autoencoder model to allow for a model to compress a wider variety of images without having to retrain. With the generalized model, we also hypothesize that a trained model would also be able to successfully compress images where the spatial patterns have not been observed before without significant reductions in performance.

II. Related Work

Alongside the rapid increase in research into HSI analysis, compression methodology has been advancing rapidly in both lossy and lossless compression. With lossy compression, there has been a wide array of approaches to de-correlate the spatial and spectral information. These can range from dimensionality reduction, wavelet transformation, and learning based compression which include neural networks and regression based learning [16], [17].

Wavelet transformation is the process of projecting a wave into smaller representations called wavelets. One of the most common wavelet transforms to compress HSI is the discrete wavelet transform (DWT), which is one of the bases of the JPEG-2000 compression standard [18]. With a spectral decorrelation process, JPEG-2000 can also successfully compress HSI [19].

Dimensionality reduction tasks look to represent the hyperspectral image in a smaller dimension. The typical process for these images is to de-correlate the dimensions and represent the image with a reduced representation of the de-correlated dimensions. An example of this process is the implementation of PCA to be used in conjunction with a JPEG compressor [20]. Non-iterative factorized tensor decomposition has also been successful in further compressing the images from the PCA algorithm [8].

While de-correlating both the spectral and spatial dimensions separately is an effective compression strategy, machine learning and deep learning processes have been integrated to improve the compression rates as well as the reconstruction error. One such example is a DWT-SVM proposed by Zikiou et al. [17] where the support vector machine is used to reduce the most important spectral bands further. A Convolutional Neural Network Autoencoder (CNN-AE) has been proposed to compress the images directly [7]. The learning of a deep belief network has been shown to analyze the data successfully to provide the optimal coding parameter for the lossless Golomn-Rice coder compression algorithm [21]. Neural networks have also been yielding significant success in learning the spectral information for classification purposes such as LSTM and Bidirectional LSTM networks [9], [11].

III. Methodology

In this section, we provide an overview on autoencoders, LSTMs, and Bidirectional LSTMs. These fundamental models will be employed in our approach to compress a hyperspectral image's spectral axis.

A. Autoencoder

With the size of HSI images, the *curse of dimensionality* can make extracting useful information from the image difficult. To mitigate its impact, dimensionality reduction processes usually occur on an image before a classification algorithm is implemented. To address this issue, an autoencoder allows for the representation of non-linear relations to extract features from an input space [22]. Given layers $h_{encoder}$ and $h_{decoder}$, input $\mathbf{x}$, encoded $\mathbf{y}$, and reconstructed $\mathbf{x}'$; an autoencoder consists of two parts, an encoder $\mathbf{X} \xrightarrow{h_{encoder}} \mathbf{Y}$ and a decoder $\mathbf{Y} \xrightarrow{h_{decoder}} \mathbf{X}'$. The encoder maps the high dimensional input $\mathbf{x}$ to a lower dimension vector $\mathbf{y}$ by passing the input data through the encoder layers. Then the decoder takes the lower dimension vector $\mathbf{y}$ and maps it to the original dimensional space through the decoder layers to get a reconstructed value $\mathbf{x}'$, which may incur some loss during the encoding and decoding process. This loss can be measured by the function $||\mathbf{x} - \mathbf{x}'||^2$, or mean squared reconstruction error (MSRE).

Autoencoders are trained as unsupervised networks, so they only require a subset of a dataset to train on [23]. The training process minimizes the loss obtained during the reconstruction process. For multi-layer AEs, they can either be trained as single feedforward networks or using what is referred to as "layerwise unsupervised pre-training" [24].

B. Long Short Term Memory Network (LSTM)

A recurrent neural network (RNN) is an artificial neural network where the network flow is represented with feedback connections. These feedback connections allow previous outputs to be used as inputs for any given timestamp, thus enabling temporal sequences to be modeled. To make a decision, the network not only considers the current input, but also the previously received inputs. For a given time step t and a node x with output y, in one type of network, the input for the node x is not only $x(t)$, but also $y(t-1)$. With this structure, RNNs can learn a variety of data sequences [25].

These networks are often trained using a backpropagation method called "backpropagation through time" [26], but the approach can lead to the vanishing gradient problem, which arises if a sequence is too long, leading to prior time steps eventually get left behind [27]. A modification on the RNN architecture was proposed by Hochreiter and Schmidhuber to overcome the disadvantages of the RNN called the Long Short Term Memory (LSTM) [28]. In an LSTM, each node in the hidden layer is replaced with a memory cell.

In each cell, there is a memory state m_t, the hidden state of the cell h_t, and the input value of the sequence x_t. There are three regulatory gates: the input gate, the output gate, and the forget gate, with the representations $\mathbf{i}_t$, $\mathbf{o}_t$, $\mathbf{f}_t$ respectfully. Given the previous cell's memory state $\mathbf{m}_{t-1}$, hidden state $\mathbf{h}_{t-1}$, and input vector $\mathbf{x}_t$, the values of each of the gates is calculated with the following equations:

$$\begin{aligned}
\mathbf{i}_t &= \text{sigm}(\mathbf{W}_i \odot [\mathbf{x}_t, \mathbf{h}_{t-1}]) \odot \tanh(\mathbf{W}_m \odot [\mathbf{x}_t, \mathbf{h}_{t-1}]) \\
\mathbf{o}_t &= \tanh(\mathbf{i}_t + \mathbf{f}_t) \odot \text{sigm}(\mathbf{W}_o \odot ([\mathbf{x}_t, \mathbf{h}_{t-1}])) \\
\mathbf{f}_t &= \mathbf{m}_{t-1} \odot \text{sigm}(\mathbf{W}_f \odot [\mathbf{x}_t, \mathbf{h}_{t-1}]) \\
\mathbf{m}_t &= \mathbf{i}_t + \mathbf{f}_t
\end{aligned}$$

The Bidirectional LSTM (BLSTM) is a modification of the LSTM network that includes a second independent hidden layer for each LSTM node that processes the input sequence in reverse order from the standard hidden layer. The outputs of the standard LSTM layer and the bidirectional layer converge into the output node for the sequence. BLSTMs have been shown to be able to outperform LSTMs on tasks such as speech recognition [29], but they also require the training of an extra layer for each node.

IV. HSI Compression Model

The model presented in this work to compress the spectral dimension of a hyperspectral image is a combination of the autoencoder structure and the LSTM. We also present a modification where we incorporate a bidirectional LSTM. To achieve a sequential context representation, we define each pixel as a sequence of reflectance values.

The input value x will be a pixel with n reflectance bands represented by the sequence $x_i = [x_i^1, x_i^2, ..., x_i^{n-1}, x_i^n]$. The compressed representation of a given pixel will be represented with the value y and the reconstructed pixel sequence will be represented by x'.

A. LSTM-Autoencoder

The LSTM-Autoencoder combines the feature learning of an autoencoder with the sequential context representation of an LSTM. This is accomplished by having each layer of the autoencoder contain a series of single layer LSTM models. Both the encoder and decoder contain the same number of LSTM nodes.

The autoencoder structure is trained using the Adam optimizer introduced by Kingma [30]. The Adam optimizer is a stochastic gradient descent method that employs an adaptive learning rate strategy and calculates an exponential moving average of the gradient and the squared gradients. These properties allow Adam to require fewer tunable inputs while still being able to handle noisy and some non-convex stochastic gradient descent problems [30], [31]]. Each LSTM layer is trained by backpropagation through time as defined by Mozer [26]. We used mean squared reconstruction error (MSRE) for the loss function.

Given an input sequence x, the autoencoder extracts the features from the sequence. The encoder is a three layered network consisting of 20 nodes of LSTM networks, 5 nodes of LSTM networks, and a single node LSTM network. The intermediate layers pass the full sequence to the next layer while the final encoding layer outputs the n^{th} sequence position of each node. In summary, the encoding layers take a pixel sequence x and output a 16 bit unsigned float within a range of [0,1].

The decoder takes the reduced representation and reconstruct the original sequence. In addition to the LSTM layers, a Repeat Vector Layer and a Time Distributed Layer are utilized to resize the sequences. The Repeat Vector layer takes the compressed value and duplicates the input values to build the original sequence length. The LSTM layers reconstruct the original sequence values of the pixel. Time Distributed Layers are dense layers that are designed to preserve the input sequences while the sequences are being processed, which then condense the nodes into the original spectral sequence of the pixel.

TABLE I
SUMMARY OF THE 5 SELECTED IMAGES DATACUBE SIZE.

Dataset	Rows	Columns	Bands
Indian Pines	145	145	220
Pavia Center	1096	1096	102
KSC	512	614	176
Salinas	512	217	224
Botswana	1476	256	145

B. Bidirectional LSTM-Autoencoder

The Bidirectional LTSM Autoencoder is a modification of the LTSM Autoencoder that includes an extra reversed sequence representation for each LSTM node. This allows the BLSTM to extract the features of the sequence, not only from a left to right encoding, but a right to left encoding as well. The structure of the network does not change fundamentally from the LSTM Autoencoder, but the size of the layers increases due to the capturing of the standard LTSM output as well as the independent reversed LSTM Layers.

V. Evaluation and Analysis

This section provides experimental results of the LSTM-based compression algorithms, along with a discussion of the results. The following subsections will provide a detailed description of the datasets used and the results for a series of varying experiments to examine the capabilities of the method.

A. Datasets

The LSTM-AE algorithms were evaluated on five different hyperspectral datasets: Indian Pines (acquired by Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor), Pavia Center (acquired by Reflective Optics System Imaging Spectrometer (ROSIS) sensor), Kennedy Space Center (KSC) (acquired by AVIRIS), Salinas scene (acquired by AVIRIS), and Botswana (acquired by the Hyperion sensor). The Hyperion and AVIRIS sensor bands both have a resolution of 10nm with AVIRIS ranging from 400nm-2400nm and Hyperion

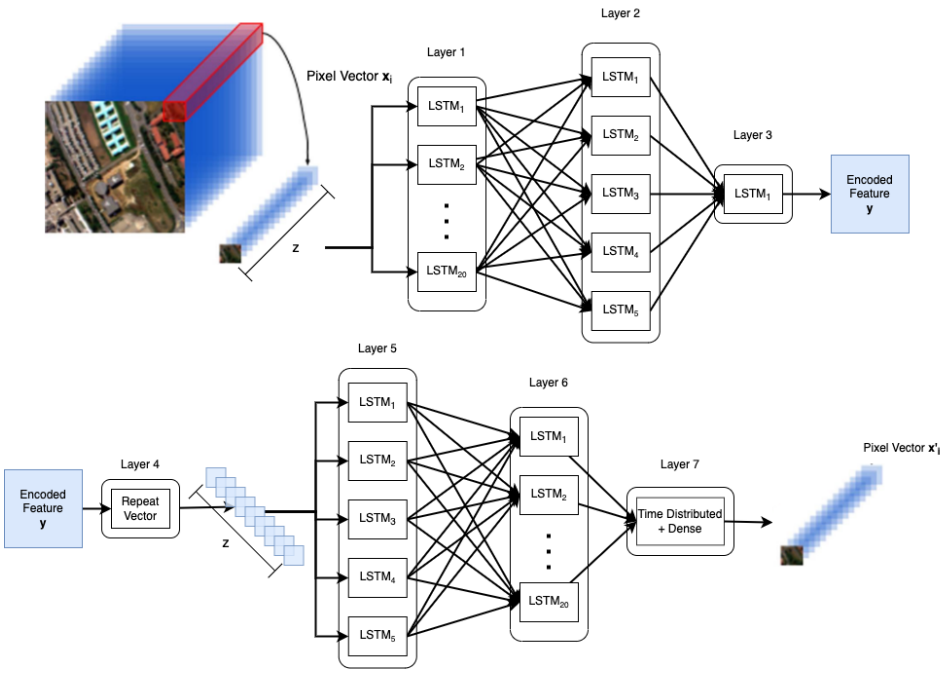

Fig. 1. Diagram of the compression model.

ranging from 400-2500nm. The ROSIS sensor has a 5nm resolution ranging from 430-960nm. Each dataset is available from [32]. Each of the images is represented visually in Figure 2 with further details given in Table I.

1) Indian Pines: The image was captured from a test site in North-western Indiana by the AVIRIS sensor. The 145×145 pixel image captured 224 bands of spectral reflectance; however, 24 bands were removed that covered the region of water absorption. The image contains a mixture of agriculture, forest or other natural perennial vegetation, and a small number of pixels contain developed elements. This image was chosen for its diverse vegetation coverage.

2) Pavia Center: Pavia Center was one of two scenes acquired by the ROSIS sensor during a flight over Pavia in northern Italy. Pavia Center captures the city itself in a 1096×1096 pixel image with 102 spectral bands. The image was preprocessed to remove zones that contain no information, which reduced the image into an approximately 1100×650 pixel image. Pavia Center was chosen because most of its pixels contain developed features with some water and mixed plant elements.

3) Kennedy Space Center: KSC was captured by the AVIRIS sensor, which captured the Merrit Island National Wildlife Refuge, the NASA Shuttle Landing Facility, and elements of the nearby town of Titusville. The 512×614 pixel image was taken with 224 captured spectral bands; however, 48 bands were removed due to them capturing water absorption and containing low signal-to-noise ratios. This image was chosen because of the split between the heavily developed Titusville and the sparse Merrit Island National Wildlife Refuge. In addition, the selection of this image allowed us to examine how well the models will be able to handle the removed spectral information.

4) Salinas: Salinas is a scene captured by the AVIRIS sensor during a flight over the Salinas Valley in California. The 512×217 pixel image captures 224 bands of spectral reflectance. This image had 20 bands removed manually from the original image due to capturing water absorption. Salinas contains a wide array of vegetable crops as well as vineyard fields. This image was chosen because of the unique crop types, including vineyards.

5) Botswana: Botswana is a scene captured by the NASA EO-1 Hyperion imager. The 1476×256 image captures 242 bands of the Okavango Delta in Botswana. The original image was preprocessed by the UT Center for Space Research, which removed 97 bands to reduce the number of water absorption features. This image was selected both because the image consists of both swamps and drier woodlands. Additionally, the image being captured by a satellite mounted imager will allow experimentation with the differences in the airborne mounted sensors and the satellite mounted sensors.

B. Metrics

The evaluation metrics that were used to evaluate the methods include mean squared reconstruction error ($MSRE$), compression rate (CR), and peak signal noise ratio ($PSNR$). Mean squared reconstruction error is defined as the differences between the original pixel and the reconstruction of that pixel after compression given by the following:

$$MSRE = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \hat{\mathbf{x}}_i||^2$$

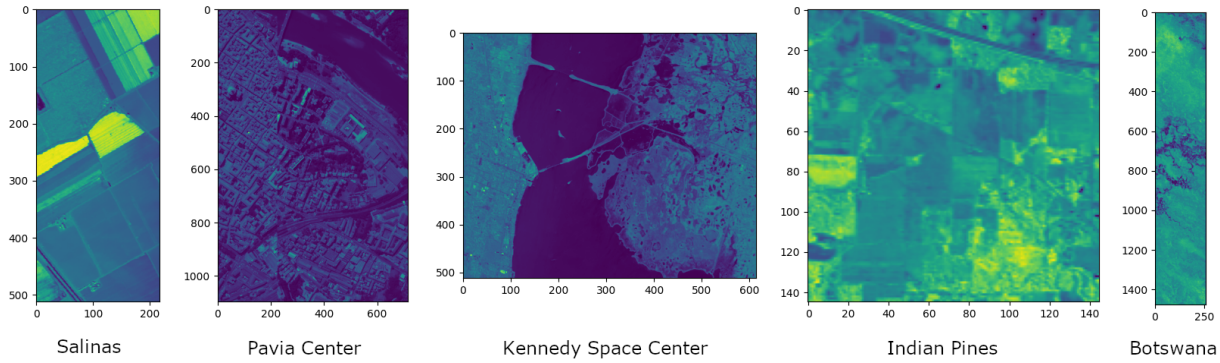

Fig. 2. Visual representation of each image's contents at the 7^{th} band.

Compression rate is defined as the ratio of the total number of the compressed image bits to the original image bits given by the following:

$$CR = \frac{\text{Size of compressed image}}{\text{Size of original image}}$$

Peak signal noise ratio (PSNR) measures the ratio of maximum power of the signal to the noise affecting the signal. This is used to represent the quality of the compression process.

$$PSNR = 10 \cdot \log_{10} \left(\frac{\text{Maximum possible pixel value}^2}{\text{MSRE}} \right)$$

The objective of compression is to minimize both the compression rate and the reconstruction error rates, thus maximizing $PSNR$.

C. Parameter Tuning

The LSTM-AE had several hyperparameters that were tuned to balance the CR and PSNR. The tuning process involved applying a grid search to select the parameters for network size, depth, learning rate, the method of selecting the random samples, dropout rates, batch size, and number of training epochs.

The tuning process found a consistent set of parameters for both the LSTM and Bidirectional LSTM autoencoders. For the network setup of LSTM, the network defined in Figure 1 was the most robust and consistent across each of the images. The BLSTM modification is applied to the LSTM structure. The optimal training process used an Adam Optimizer with a learning rate of 0.0009, 120 epochs, and a dropout rate of 0.1 to be able to quickly stabilize towards an optimal set of parameters. We experimented with a wide range of train, test, and validation splits of the pixels. Starting from an initial split of 0.7, 0.15, and 0.15, we were able to reduce the train, test, and validation split to 0.05, 0.05, and 0.9 without a significant reduction in PSNR.

D. Single Image Compression

This experiment tests the traditional approach to compressing HSI where a new model is trained for each new image. For this experiment, an entire image was used for training with approximately 25% of the pixels reserved as a validation set. The models were retrained on a randomly resampled set of pixels a total of 5 times to validate the results of the small training set.

The proposed algorithms were implemented along with state of the art techniques such as DWT-SVM [17], NFTD+PCA [8] with 16 principal components due to the large number of bands in the dataset, and a Convolutional Neural Network Autoencoder (CNN-AE) [7].

As shown in Table II, the $PSNR$ ratio performance of the LSTM-AE and BLSTM-AE models were consistently over a 15% increase in $PSNR$ than the CNN-AE, with the largest increasing being 60% in the Botswana image. Furthermore, these reconstruction rates were not at a significant loss in CR when compared to the NFTD+PCA and CNN-AE, whereas the DWT-SVM was consistently the lowest compression rate as demonsstrated by Table III.

E. Reduced Training Set

This experiment also tests the traditional approach to compressing HSI where a new model is trained for each new image. The difference for this experiment is that the subset of pixels for training were sampled uniformly at random, rather than using the entire image. Since the input layer of the network and the connections to the first layer are set dynamically, both the LSTM and BLSTM autoencoders do not have to be resized manually for each image. The models were retrained on a randomly resampled set of pixels a total of 5 times to validate the results of the small training set.

Through a grid search optimization process, we reduced the training set to be 5% of the image's pixels, our validation set to be 5% of the pixels, and leaving 90% of the pixels as a test set. Table IV demonstrates the compression performance of the 5% training set. In Tables IV and Table II, we can see that when comparing the whole image to the reduced dataset, there was little change in performance between the images for LSTM-AE except in the case of the Pavia Center image, which saw an 8% increase in $PSNR$. Given Pavia Center's sparseness in classes, the increase may be explained best as overfitting in Experiment 1 for the unclassified pixels. In the case of

TABLE II
COMPARISON BETWEEN EACH OF THE METHODS FOR EACH OF THE IMAGES WITH RESULTS FROM THE FULL SINGLE IMAGE COMPRESSION. THE BOLD VALUES INDICATE HIGHEST AVERAGE PERFORMANCE.

	LSTM-AE		**BLSTM-AE**		**3D DWT +SVR**		**NFTD +PCA**		**CNN-AE**	
Dataset	$PSNR$	$MSRE$	$PSNR$	$MSRE$	$PSNR$	$MSRE$	$PSNR$	$MSRE$	$PSNR$	$MSRE$
Indian Pines	62.07	0.0020	**64.84**	**0.0015**	42.37	0.0145	49.92	0.0068	54.85	0.0041
Pavia Center	61.24	0.0022	**72.73**	**0.0007**	43.01	0.0136	46.78	0.0093	55.17	0.0040
KSC	78.70	0.0004	**78.80**	**0.0004**	42.84	0.0138	48.16	0.0081	54.15	0.0044
Salinas	65.40	0.0014	**66.73**	**0.0013**	41.32	0.0161	46.78	0.0093	54.68	0.0042
Botswana	89.53	0.0001	**91.40**	**0.0001**	44.92	0.0112	49.48	0.0071	57.29	0.0033

TABLE III
COMPARISON OF THE COMPRESSION RATES BETWEEN EACH OF THE METHODS FOR EACH OF THE IMAGES. THE BOLD VALUES INDICATE HIGHEST AVERAGE PERFORMANCE.

	LSTM -AE	**BLSTM -AE**	**3D DWT +SVR**	**NFTD +PCA**	**CNNs -AE**
Dataset	CR	CR	CR	CR	CR
Indian Pines	**0.0045**	0.0090	0.037	0.0077	0.0078
Pavia Center	0.0098	0.0195	0.039	0.0073	**0.0063**
KSC	0.0057	0.0114	0.038	0.0043	**0.0027**
Salinas	0.0045	0.0090	0.046	0.0037	**0.0022**
Botswana	0.0069	0.0138	0.043	0.0052	**0.0032**

TABLE IV
COMPARISON RESULTS BETWEEN LSTM-AE AND BLSTM-AE FOR EACH OF THE IMAGES WITH RESULTS FROM THE REDUCED TRAINING SET. THE BOLD VALUES INDICATE HIGHEST AVERAGE PERFORMANCE.

	LSTM-AE		**BLSTM-AE**	
Dataset	$PSNR$	$MSRE$	$PSNR$	$MSRE$
Indian Pines	63.27	**0.0018**	62.25	0.0020
Pavia Center	66.38	0.0013	72.70	**0.0007**
KSC	78.56	0.0004	78.69	**0.00038**
Salinas	67.28	**0.0012**	66.73	0.0013
Botswana	92.20	**9.90e-5**	89.63	0.0001

BLSTM-AE, Experiment 2 saw a reduction from Experiment 1 across all of the images. This may be due to the BLSTM being a network twice as large as the LSTM-AE. Thus, the significantly increased training set size was more beneficial for the BLSTM-AE model's performance than for the LSTM-AE model, but a reduced training set size was more beneficial for the LSTM-AE's reconstruction error.

F. Multi-Image Compression

This experiment considered the network's performance when training on a set of pixels sampled from multiple hyperspectral images. Since this is combining the pixels from multiple images, the pixels were padded to be of uniform size (i.e., 224 bands) for training the model. Since the LSTM structure only sees the pixels as a sequence of integers, this avoids requiring the selected images to contain the same spectral resolution and range. The result of the training process leaves us with one trained model that is then used to compress each of the images. This process allows us to save overall training time since a single model with the same dimensions but a larger training set will take significantly less time to train than models trained on individual images. The training process was resampled 5 times and averaged for validation.

TABLE V
COMPARISON BETWEEN LSTM-AE AND BLSTM-AE FOR EACH OF THE IMAGES WITH RESULTS FROM THE MULTI-IMAGE COMPRESSION EXPERIMENT. THE BOLD VALUES INDICATE HIGHEST AVERAGE PERFORMANCE.

	LSTM-AE		**BLSTM-AE**	
Dataset	$PSNR$	$MSRE$	$PSNR$	$MSRE$
Indian Pines	84.64	0.0002	**86.19**	**0.0002**
Pavia Center	91.78	0.0001	**93.35**	**8.83e-5**
KSC	78.89	0.0004	**79.13**	**0.0004**
Salinas	88.28	0.0001	**88.44**	**0.0001**
Botswana	90.19	0.0001	**93.01**	**9.13e-5**

As shown in Tables II, IV, and V, the $PSNR$ improved significantly for most of the image with Indian Pines, Pavia Center, and Salinas, with an over 33% increase over the full image compression $PSNR$. Unlike in the reduced set, the BLSTM-AE was consistently an improvement over the LSTM-AE, even if not by a statistically significant amount.

The KSC and Botswana models both have approximately the same $PSNR$ values. This may be due to the loss of a large number of bands captured by the sensors that were removed. M Graña stated that 21% of the original bands in KSC and 40% of the original bands in Botswana were removed [32]. Due to the large number of bands removed, there may not have been enough unique spectral information to be learned from other datasets. This also demonstrates the relationship between spectral information and the model's ability to minimize $PSNR$.

G. Generalized Compression

This experiment looked at the network's ability to apply a trained compression model to an unseen image. This experiment was trained and tested in a similar manner as the generalization experiment, but it employed a "leave one out" strategy where the set of training images only contained 4 out of the 5 images. The image that was not used to train the model was instead used entirely as the test data for the trained network. Each of the models was trained using the "leave one image out" strategy and resampled 5 times to validate the results.

TABLE VI
COMPARISON BETWEEN LSTM-AE AND BLSTM-AE FOR EACH OF THE IMAGES WITH RESULTS FROM THE GENERALIZED COMPRESSION EXPERIMENT. THE BOLD VALUES INDICATE HIGHEST AVERAGE PERFORMANCE.

	LSTM-AE		BLSTM-AE	
Dataset	$PSNR$	$MSRE$	$PSNR$	$MSRE$
Indian Pines	79.90	0.0003	**82.92**	**0.0003**
Pavia Center	86.95	0.0002	**89.17**	**0.0001**
KSC	78.16	0.0004	**78.85**	**0.0004**
Salinas	87.29	0.0002	**88.33**	**0.0001**
Botswana	85.17	0.0002	**87.92**	**0.0002**

TABLE VII
THE MFLOPS REQUIRED FOR EACH NETWORK TO PERFORM COMPRESSION ON EACH IMAGE.

Models	LSTM-AE	BLSTM-AE	CNN-AE
Indian Pines	0.14542	0.29084	29.98272
Pavia Center	0.067422	0.134844	29.98272
KSC	0.116336	0.232672	29.98272
Salinas	0.148064	0.296128	29.98272
Botswana	0.095845	0.19169	29.98272

As with the generalization experiment, this experiment had more than a 25% increase in $PSNR$ over Experiment 1 for the same compression rate with Indian Pines, Pavia Center, and Salinas. There was no significant change between any of the experiments for the KSC and Botswana images. The BLSTM-AE had a slight increase over the LSTM-AE network, but the increase is not significant.

When compared to the generalization experiment in Table V, Table VI shows that every $PSNR$ performed slightly worse with a difference of less then 10%. This demonstrates that the LSTM-AE structure may be robust enough not only to be able to be reused on new scenes, but also to be used across multiple sensors. This is demonstrated most clearly by the Botswana runs since Botswana was the only dataset captured by the Hyperion satellite based sensor. This also demonstrates an independence on spectral resolution since the ROSIS sensor has a 5nm resolution while all other sensors have a 10 nm resolution.

H. Computational Cost

To compare the runtime performance, we report the number of training parameters required and we calculated the number of mega floating-point operations needed (MFLOPs) to compress each pixel for LSTM-AE and CNN-AE. Table VII shows the number of MFLOPS for both of the Deep Neural Network compressors. The number of MFLOPS is related to the time complexity of processing the model. Table VIII shows the number of parameters. The number of parameters is related to the complexity of the model and the amount of data needed to train the model effectively. It is clear that the LSTM-AE model requires significantly less resources to compress the images than the CNN-AE model does. This is the case because the CNN-AE model is a 12 layer model with the number of convolutional filters per layer being containing 128 filters, 64 filters, or 32 filters whereas the LSTM-AE model is a five layer model with a total of 51 nodes in the network.

TABLE VIII
THE NUMBER OF PARAMETERS THAT ARE UTILIZED BY EACH NETWORK.

Models	LSTM-AE	BLSTM-AE	CNN-AE
Parameters	4,549	9,078	594,976

I. Statistical Analysis

With the low number of samples, we applied a permutation test to determine how significant the results we captured were in comparison to the best performing existing approach on each image. Since there was very little difference between the LSTM-AE and BLSTM-AE algorithms performance, we performed the permutation tests on the model that performed the worst for CR and $PSNR$ for our comparisons. Comparing the CR of the BLSTM to the existing methods, we calculated a p-value of 0.255. For the following experiments, we performed the permutation tests on the LSTM-AE due to it generally having a higher error rate than BLSTM. Experiments 1 and 2 achieved a p-value of 0.065 and 0.066 while Experiments 3 and 4 achieved a p-value of 0.062 and 0.060. Given the small sample size, these p-values suggest that there may be a sufficient difference between the existing methods and the LSTM-AE and BLSTM-AE algorithms proposed, even if the significance threshold has not been met.

VI. CONCLUSION

The LSTM-AE framework demonstrated a large improvement when compared to state of the art methods with regards to reconstruction error. It was also demonstrated that the improvement in $PSNR$ did not compromise the compression level of the method. The framework also demonstrated a level of robustness to be utilized across drastically differing scenes as well as differing sensors and platforms. This may also suggest that the requirement to retrain on new datasets may not be required like it is in methods such as the CNN-AE method [7]. While the BLSTM-AE did demonstrate a large improvement when compared to state of the art methods, the increase in complexity was not necessarily beneficial when compared to the LSTM-AE's reconstruction error performance.

A. Limitations

The primary limitation with a spectral based autoencoder is that the compression rates are based on the size of the original bands as well as the size of the central bottleneck layer. This limitation of the compression ratio can prevent the model from being competitive with methods that compress both spatial and spectral information for images that contain a smaller number of bands but a larger spatial dimension size.

Another limitation on the method is the removal of spatial information. The removal assumes that the level of spectral information will be diverse enough in images that the compressor can distinguish each of the bands. If this is not the

case for an image, then the spectral based strategy may not be as effective of a compressing tool.

B. Future Work

The consistent improvement for Experiments 3 and 4 and the lower p-scores suggest that the LSTM-AE and BLSTM-AE models may be able to be more robust across a wider selection of hyperspectral images and sensor types. The slight performance boost in BLSTM-AE over LSTM-AE also suggests that the BLSTM-AE may be more robust and require retraining less often. For future research, the addition of a spacial compression process can be examined for problems that require a further reduction in compression rates. Another interesting direction for future research is the inclusion of alternative layers to supplement the LSTM layers, such as the Dense layer or a 1D Convolutional layer, to improve robustness across a variety of images. In particular, images that contain low levels of spectral correlation or invariance. This may improve reconstruction error rates for images that contain low correlation between the bands.

Autoencoders also have the ability to be adapted for feature extraction problems. One direction of future research is the evaluation of the extracted features from the compressed images to train a classifier. This research could determine if a compressed model can perform classification tasks and can reduce the complexity of classifying hyperspectral images.

REFERENCES

[1] Y. Liu, H. Pu, and D.-W. Sun, "Hyperspectral imaging technique for evaluating food quality and safety during various processes: A review of recent applications," *Trends in Food Science & Technology*, vol. 69, pp. 25–35, Nov. 2017.

[2] A. Mancini, E. Frontoni, and P. Zingaretti, "A multi/hyper-spectral imaging system for land use/land cover using unmanned aerial systems," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2016, pp. 1148–1155.

[3] X. Ye, K. Sakai, M. Manago, S.-i. Asada, and A. Sasao, "Prediction of citrus yield from airborne hyperspectral imagery," *Precision Agriculture*, vol. 8, no. 3, pp. 111–125, Jun. 2007.

[4] Y. Barrios, A. Rodríguez, A. Sánchez, A. Pérez, S. López, A. Otero, E. de la Torre, and R. Sarmiento, "Lossy Hyperspectral Image Compression on a Reconfigurable and Fault-Tolerant FPGA-Based Adaptive Computing Platform," *Electronics*, vol. 9, no. 10, p. 1576, Oct. 2020.

[5] W. Fu, S. Li, L. Fang, and J. A. Benediktsson, "Adaptive Spectral–Spatial Compression of Hyperspectral Image With Sparse Representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 671–682, Feb. 2017.

[6] Q. Du and J. E. Fowler, "Low-Complexity Principal Component Analysis for Hyperspectral Image Compression," *The International Journal of High Performance Computing Applications*, vol. 22, no. 4, pp. 438–448, Nov. 2008.

[7] Y. Dua, R. S. Singh, K. Parwani, S. Lunagariya, and V. Kumar, "Convolution Neural Network based lossy compression of hyperspectral images," *Signal Processing: Image Communication*, vol. 95, p. 116255, Jul. 2021.

[8] J. S and S. S, "Hybrid hyperspectral image compression technique for non-iterative factorized tensor decomposition and principal component analysis: application for NASA's AVIRIS data," *Computational Geosciences*, vol. 23, no. 5, pp. 969–979, Oct. 2019.

[9] P. Zhou, J. Han, G. Cheng, and B. Zhang, "Learning Compact and Discriminative Stacked Autoencoder for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4823–4833, Jul. 2019.

[10] F. Zhou, R. Hang, Q. Liu, and X. Yuan, "Hyperspectral image classification using spectral-spatial LSTMs," *Neurocomputing*, vol. 328, pp. 39–47, Feb. 2019.

[11] Q. Liu, F. Zhou, R. Hang, and X. Yuan, "Bidirectional-Convolutional LSTM Based Spectral-Spatial Feature Learning for Hyperspectral Image Classification," *Remote Sensing*, vol. 9, no. 12, p. 1330, Dec. 2017.

[12] Y. Xu, B. Du, L. Zhang, and F. Zhang, "A Band Grouping Based LSTM Algorithm for Hyperspectral Image Classification," in *Computer Vision*, ser. Communications in Computer and Information Science, J. Yang, Q. Hu, M.-M. Cheng, L. Wang, Q. Liu, X. Bai, and D. Meng, Eds. Singapore: Springer, 2017, pp. 421–432.

[13] J. Tu, H. Liu, F. Meng, M. Liu, and R. Ding, "Spatial-Temporal Data Augmentation Based on LSTM Autoencoder Network for Skeleton-Based Human Action Recognition," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 3478–3482.

[14] E. Marchi, F. Vesperini, S. Squartini, and B. Schuller, "Deep Recurrent Neural Network-Based Autoencoders for Acoustic Novelty Detection," *Computational Intelligence and Neuroscience*, vol. 2017, Jan. 2017.

[15] G. Morales, J. W. Sheppard, R. D. Logan, and J. A. Shaw, "Hyperspectral Dimensionality Reduction Based on Inter-Band Redundancy Analysis and Greedy Spectral Selection," *Remote Sensing*, vol. 13, no. 18, p. 3649, Jan. 2021.

[16] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "A new deep convolutional neural network for fast hyperspectral image classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 145, pp. 120–147, Nov. 2018.

[17] N. Zikiou, M. Lahdir, and D. Helbert, "Support vector regression-based 3D-wavelet texture learning for hyperspectral image compression," *The Visual Computer*, vol. 36, no. 7, pp. 1473–1490, Jul. 2020.

[18] ISO/IEC:15444-1, "JPEG 2000 image coding systems," International Organization for Standardization, London, United Kingdom, Standard, 2019.

[19] H. Lee, N. Younan, and R. King, "Hyperspectral image cube compression combining JPEG-2000 and spectral decorrelation," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. 6, Jun. 2002, pp. 3317–3319 vol.6.

[20] Q. Du and J. E. Fowler, "Hyperspectral Image Compression Using JPEG2000 and Principal Component Analysis," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 2, pp. 201–205, Apr. 2007.

[21] Z. Jiang, W. D. Pan, and H. Shen, "Universal Golomb–Rice Coding Parameter Estimation Using Deep Belief Networks for Hyperspectral Image Compression," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 10, pp. 3830–3840, Oct. 2018.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, Nov. 2016.

[23] T. V. Nguyen, R. K. W. Wong, and C. Hegde, "On the Dynamics of Gradient Descent for Autoencoders," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, Apr. 2019, pp. 2858–2867.

[24] D. Erhan, Y. Bengio, A. Courville, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, pp. 625—-660, 2010.

[25] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning," *arXiv:1506.00019 [cs]*, Oct. 2015.

[26] M. Mozer, "A Focused Backpropagation Algorithm for Temporal Pattern Recognition," *Complex Syst.*, 1989.

[27] O. Yildirim, "A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification," *Computers in Biology and Medicine*, vol. 96, pp. 189–202, May 2018.

[28] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[29] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition," in *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, ser. Lecture Notes in Computer Science, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrożny, Eds. Berlin, Heidelberg: Springer, 2005, pp. 799–804.

[30] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017.

[31] L. Hodgkinson and M. Mahoney, "Multiplicative Noise and Heavy Tails in Stochastic Optimization," in *International Conference on Machine Learning*. PMLR, Jul. 2021, pp. 4262–4274.

[32] M. Graña, M.A. Veganzons, and B. Ayerdi, "Hyperspectral Remote Sensing Scenes," May 2011.