Towards Intelligent Compression of Hyperspectral Imagery

by

Diego Wildenstein

BSE Electrical Engineering, Arizona State University, 2019

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2021

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Diego Wildenstein

It was defended on

April 1, 2021

and approved by

Alan D. George, PhD, Mickle Chair Professor,

Department of Electrical and Computer Engineering

Zhi-Hong Mao, PhD, Professor, Department of Electrical and Computer Engineering

Samuel Dickerson, PhD, Director, Department of Electrical and Computer Engineering

Thesis Advisor: Alan D. George, PhD, Mickle Chair Professor,

Department of Electrical and Computer Engineering

Towards Intelligent Compression of Hyperspectral Imagery

Diego Wildenstein, M.S.

University of Pittsburgh, 2021

For space applications, communications bandwidth is often contested and limited. Limited communications is particularly evident for space vehicles in low-Earth orbit, where space-based sensor platforms that collect excess amounts of data will often have difficulty communicating this data in a timely manner. Data compression is a necessary step in transferring large files, such as high resolution images, as it allows a system to make more efficient use of communications bandwidth. Sensing systems continue to evolve with increased resolutions and data rates that effectively increase the overall amount of data. Hyperspectral cameras are one type of imaging sensors that produce vast amounts of data relative to conventional camera systems. Due to the data increase, hyperspectral sensors can potentially benefit significantly from data compression. CNN-JPEG is a state-of-the-art, neural network-based compression framework. This algorithm is a lossy, end-to-end image compression system that has been adapted from previous literature to compress hyperspectral imagery collected from the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor. By applying the CNN-JPEG algorithm to hyperspectral imagery, we achieve upwards of 17× compression ratios over the original image on average for each spectral band. Additionally, CNN-JPEG compression provides over 14× compression across an entire hyperspectral dataset. This increased compression ratio comes at the cost of decreased reconstruction quality. Additionally, the amount of compression delivered by CNN-JPEG is highly dependent on the content of the image. CNN-JPEG provides a unique option for space platforms in which high compression is desired with acceptable losses in image quality.

Table of Contents

List of Tables

List of Figures

Preface

This work was supported by the NSF center for Space, High-performance, and Resilient Computing (SHREC) industry and agency members and by the IUCRC Program of the National Science Foundation under Grant No. CNS-1738783.

The author would like to thank the members of the Air Force Research Lab Space Vehicles Directorate as well as Mr. David Langerman for their continuing guidance and support of this research.

1.0 Introduction

Hyperspectral imaging sensors have wide variety of applications. Measuring along the z-axis of a hyperspectral data cube provides a spectral signature corresponding to that pixel, which can be compared to known spectral signatures to determine the chemical composition of the material contained within that pixel. Hyperspectral sensing is also a popular tool for mining and mineralogy, where spectra can be used to identify materials such as soils and ores within a scene [1]. Hyperspectral imaging is also of interest in defense apps, where the movement of objects can be tracked throughout a scene based their unique spectral signature [2, 9].

Due to the large size of hyperspectral data cubes, typically on the order of gigabytes, it is vital for spacecraft to compress these data cubes for more efficient data storage and transfer. To this end, many solutions for hyperspectral data compression exist using both lossy and lossless techniques, and each offers a unique set of performance tradeoffs. For space platforms, an ideal compression framework has three desirable characteristics: high compression ratio, high reconstruction quality, and feasibility of implementation and use on embedded platforms. Recent advances in onboard processing and space hardware have demonstrated the use of deep learning applications for spacecraft [7, 8].

Designing for remote-sensing and space-computing platforms involves a wide variety of interesting engineering challenges, including size, weight, power, and cost (SWaP-C) constraints due to hazardous environmental conditions. In particular, the space environment introduces several complications for communications. Atmospheric conditions, as well as space radiation often result in situations where communication with a ground station is only possible within limited windows [10]. Additionally, onboard applications must constantly compete for available transmission bandwidth. Consequently, the amount of data that can be communicated to a ground station or other space platform is also limited. As the resolution and data rate of remote-sensing technology increases, the amount of data that is collected by a space platform also increases. This creates an issue where the finite communications bandwidth is saturated and unable to transfer an ever-increasing amount of data

to operators on the ground. One solution for these communication issues is to make more efficient use of available bandwidth using data compression. Data compression is a proven strategy for increasing the efficiency of digital storage and communications [11, 13].

Data compression is a vital step in the processing pipeline for space platforms. As a result of the unique challenges introduced by the space environment, flight systems often impose restrictions on energy consumption. Due to these constraints, the computational capability as well as the memory available to the flight system is often limited. For platforms that produce large amounts of sensor data, these limitations produce a problem: onboard processing of large amounts of data may not be feasible, and transmission of the data to another system may be subject to interference. Additionally, transmitting data can be costly in terms of time. Data compression is a common technique that can be used to transfer data quickly and with acceptable bandwidth efficiency.

This research seeks to offer an investigation and assessment of the feasibility and accuracy of deep-learning-based compression models for hyperspectral imagery. As noted in [6], deep-learning-based compression frameworks have not been as widely studied as transform-based compression models for hyperspectral data. This research demonstrates that deep-learning models provide enhanced image compression over traditional methods with the natural trade-off of reduced reconstruction quality. The advantages of enhanced compression show that deep-learning-based compression methods are worth consideration for hyperspectral imaging systems.

2.0 Background

This section presents two broad categories of data compression and provides a brief description of the differences between them. Additionally, this chapter introduces a high level description of hyperspectral imaging and the CNN-JPEG compression system. A discussion of recent advances in hyperspectral compression and onboard processing is also included.

2.1 Data Compression

Data compression consists of two broad categories: lossy and lossless. Lossless compression schemes are able to perfectly reconstruct the data without any loss of information. This is especially useful for compression of numerical or text data, where errors or losses in precision are especially evident. However, these methods of compression are often limited in terms of how much the data can be compressed. Since the data must be reconstructed perfectly in a lossless scheme, the compression algorithm must preserve enough information to recreate the original data. Consequently, the compression ratios offered by lossless schemes are not as high as other methods [11]. Lossless compression is best employed in apps where precision and accuracy of the data is paramount over memory constraints.

In lossy compression schemes, quantization schemes are employed alongside other techniques to reduce the number of bits required to represent the data [13]. However, a reduction in representative bits often results in reduced reconstruction quality. After quantization, the original data cannot be perfectly reassembled during decompression. Lossy methods are generally able to achieve higher compression ratios than lossless schemes, with the drawback of a loss in quality. Lossy compression is typically employed in image, audio, and video compression. In these forms of media, losses in precision and quality are typically less perceptible than losses in text data. Consequently, the benefits of using lossy compression schemes over lossless for image and video data are many. Since videos and images can reach very large sizes, they are able to benefit from the higher compression ratios offered by lossy schemes.

Hyperspectral images can often reach upwards of several gigabytes in size; applying lossy image compression to these images provides a much larger reduction in file size. Additionally, the quantization can be tuned to provide higher or lower degrees of compression based on the needs of the application.

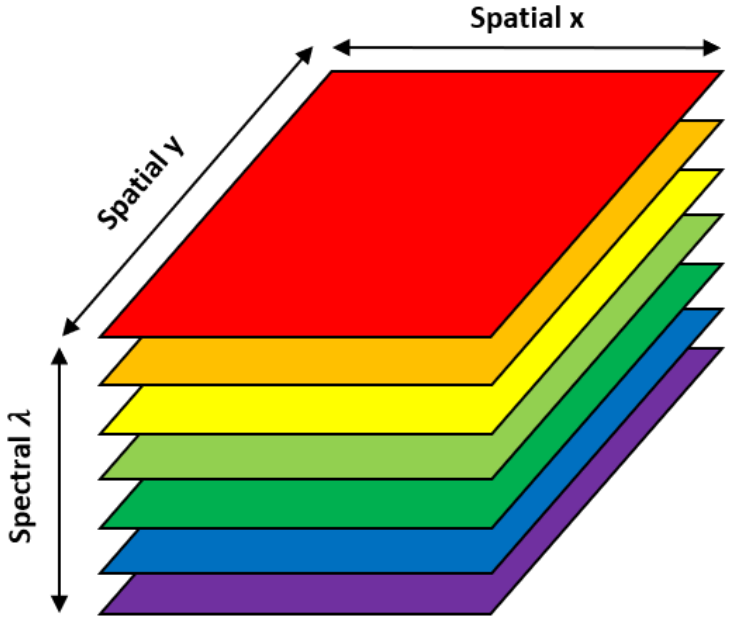

Figure 1: Structure of a hyperspectral data cube

2.2 Hyperspectral Imagery

Hyperspectral imagery differs from conventional imagery in that it captures many more spectral bands than an RGB camera. Often this is manifested as an image containing a much wider range of wavelengths than visible light. Hyperspectral sensors often collect data from near-ultraviolet through the infrared spectrum, typically wavelenghts from around 400 nanometers to 2500 nanometers. By capturing a wider band of the electromagnetic spectrum, hyperspectral sensors capture much more spectral information. This increase in spectral information brings new possibilities for imaging applications but requires much

more storage to represent that information. Since hyperspectral images include more than the visible spectrum alone, they are often represented as an image cube, where x and y coordinates refer to the spatial location of a pixel, and the λ coordinate provides the spectral content of that pixel. Fig. 1 provides an example representation of a hyperspectral image cube. In this example, each plane corresponds to a capture of the scene taken at a specific wavelength, known as a spectral band. These spectral bands are then stacked according to increasing wavelength. Conventional RGB images contain three color bands, corresponding to red, green, and blue wavelengths. Hyperspectral images often include several additional color bands as well as bands from the infrared and near-ultraviolet spectrum.

For this research, we employ image data gathered by the NASA Airborne Visible Infrared Imaging Spectrometer (AVIRIS) sensor. This sensor captures spectral radiance from 400 nm to 2500 nm, which is arranged in 224 distinct spectral bands. With 224 spectral bands, hyperspectral images contain nearly 75× as much data as a more traditional three-channel RGB image with the same resolution. The AVIRIS sensor is mounted on a high-altitude vehicle and captures spectral radiance in a whisk broom scanning method as the vehicle moves across the target area. Together, this data forms a highly dimensional composite image with dimensions equal to the sensor swath width by the flight length by 224 spectral bands.

2.3 Adaptive Compression with CNN-JPEG

Many popular compression techniques rely on some form of transform encoding to compress an image. Standard JPEG encoding employs a sequence of downsampling and transformation mapping, a discrete cosine transform, and quantization [20]. To reconstruct the image, the sequence is performed in inverse order. The JPEG transform is data-agnostic and will function in the same way independent of the data that is being processed. This feature allows JPEG to be applied universally to all forms of imagery. However, for this same reason, JPEG encoding is rarely the optimal mode of compression for every use case [15]. Recently, NN-based compression techniques have become the subject of much research

[3, 14, 17]. The technique in many NN-based frameworks is to take the initial stage of downsampling and mapping in a JPEG transform and replace it with a CNN that can optimize this mapping for the expected form of imagery. CNN-based adaptive compression algorithms are most applicable when there is some knowledge of the expected image content. Strategies using CNNs seek to minimize both the number of bytes required to store an image and the reconstruction loss of the compressed image.

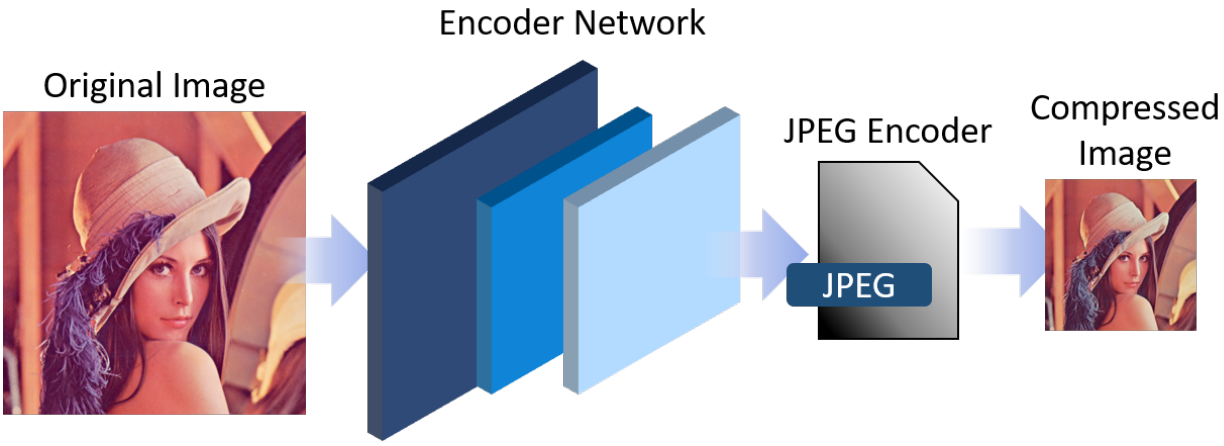

Figure 2: The CNN-JPEG encoder pipeline

The CNN-JPEG system is adapted from [17] and consists of three major components: the encoder network, a compression codec, and the decoder network. The CNN-JPEG encoding scheme is visualized in Fig. 2. The encoder network is a 3-layer neural network that reduces the spatial dimensions of the original image while preserving structural information. The resulting output is a compact representation of the original image with fewer high frequency components. This image is then encoded using a JPEG codec to achieve a fully compressed image. With a reduced amount of high frequency components, the JPEG algorithm is able to more efficiently compress the compact representation of the image. For image reconstruction, the compressed representation is first decoded using the JPEG codec. Next, it is then processed by the 20-layer decoder network. The decoder network consists of a series of convolution and activation layers followed by convolution and batch normalization layers. Using bicubic interploation, the decoder upsamples the image to the dimensions of the original image. The decoder network also predicts a residual image which is added to

the upscaled image to restore visual fidelity. The CNN-JPEG system has been demonstrated as a compression framework for onboard compression of satellite imagery [8], which makes CNN-JPEG an ideal system for this investigation.

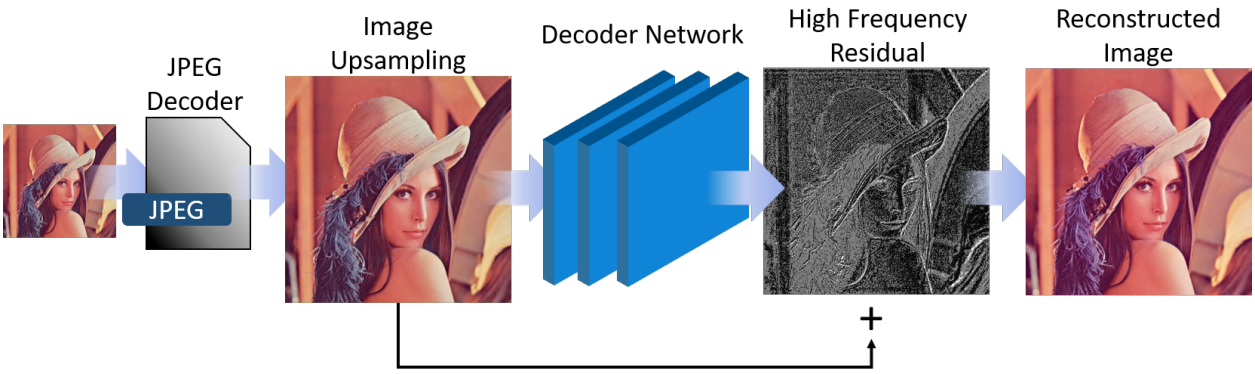

Figure 3: The CNN-JPEG decoder pipeline

Fig. 3 illustrates the CNN-JPEG image reconstruction process. The decoder network in this process contains 20 layers. The first of these layers is a convolutional layer followed by a Rectified Linear Unit (ReLU) activation function. Each subsequent layer then consists of a convolution, batch normalization, and ReLU activation in series. The 20th layer of the reconstruction CNN is a simple convolutional layer. Passing an image through the reconstruction network produces a residual image containing the high frequency components of the original, uncompressed image. This residual is added to the upsampled image to produce the fully reconstructed image. Since the encoder network reduces the amount of high frequency information in the image, the JPEG codec can efficiently compress the reduced image. During reconstruction, adding the decoder residual to the image restores the high frequency components of the image. In this way, the CNN-JPEG system is able to more efficiently compress images containing high variance and detailed content. This allows the algorithm to achieve high compression ratios while maintaining reconstruction quality by preserving detail within the image.

2.4 Related Research

Hyperspectral imaging is a vital component of several remote sensing applications. One of the largest uses of hyperspectral sensors is for Earth observation applications. The hyperspectral imager for the coastal ocean described by Corson et al. in [4] is a spaceborne hyperspectral sensor designed to study ocean and coastal environments. By measuring the wavelength and intensity of light scattered by seawater, it is possible to determine the composition and density of regions of the ocean at a given depth. Additionally, Earth observation hyperspectral sensing is also employed in a wide variety of geological applications. Van der Meer et al. provides a description of several applications in [19], including the identification of valuable minerals and mapping of hydrothermal systems. The spectral signature of an observed region can be compared to a database of known spectral signatures. This allows for the identification of certain ores or other materials within a scene. Likewise, geothermal and hydrothermal activity can be identified by the presence of volcanic materials.

Hyperspectral remote sensing is also of interest for defense applications. Nguyen et al. presents a tracking system based on hyperspectral video sensors in [12]. Using the spectral signature produced by specific objects within the scene, the movement of those objects can identified by following that spectral signature between each frame of the video. Additionally, since hyperspectral sensors often capture wavelengths outside of the visible spectrum, it is possible to identify objects even in low visibility conditions or dynamic lighting within the scene. However, hyperspectral video systems produce massive amounts of data compared to conventional video. Since each frame is itself a hyperspectral image cube, hyperspectral video requires a large amount of storage relative to visible spectrum media. As a result, hyperspectral video is often limited to low framerates and low spatial resolution.

Compression of hyperspectral image data is a widely researched topic. The characteristic 3D structure of the hyperspectral data cube presents some interesting design challenges. Compression methods can leverage spatial locality, spectral locality, or both depending on the desired effect. The needs of hyperspectral apps may call for different approaches to compression. Many hyperspectral techniques have been catalogued by Dua et al. in [6]. According to this classification, machine-learning-based compression frameworks have not

been studied as thoroughly as transform-based or prediction-based algorithms. This research notes that compression via neural-network (NN) inference is capable of operating in real time, which is of great benefit to time-sensitive systems. However, NN-based compression frameworks can come at the cost of more difficult implementation and the need for carefully curated training data.

Transform-based algorithms are commonly used for hyperspectral compression. The algorithm proposed by Tang et al. in [16] employs a lossy three-dimensional discrete wavelet transform known as 3DSPECK. This is an expansion of a 2D compression algorithm to operate on a 3D dataset. By using a discrete wavelet transform, 3DSPECK is able to better exploit both spatial and spectral locality. As a result, this algorithm is able to achieve a high degree of compression with minimal reconstruction loss. 3DSPECK compresses data using a non-scanning snapshot approach. The hyperspectral image is not partitioned based on spectral or spatial bands; rather, 3DSPECK processes the entire hyperspectral image at once to take full advantage of any data locality.

Lossy compression techniques are a proven option for hyperspectral data. Du et al. describes a lossy algorithm based on JPEG2000 encoding with principal component analysis (PCA) [5]. In this case, PCA is employed for decorrelation of spectral bands, as well as dimensionality reduction. This spectral decorrelation is able to exploit spectral locality by removing redundancies along the spectral axis. PCA also reduces the dimensionality of the data to a set of principal components, which leads to fewer values which need to be stored, thus providing enhanced compression. Notably, the use of the JPEG2000 transform is able to provide a larger signal-to-noise ratio over discrete wavelet transform methods. Lossy compression leveraging machine learning is not without precedent. Valsesia et al. has proposed onboard hyperspectral compression with reconstruction performed by using a convolutional neural network (CNN) [18]. This method is based off of an image reconstruction CNN that attempts to minimize the mean squared error between the reconstructed image and the original image. Such an adaptive reconstruction approach is able to provide a reconstruction quality that is competitive with lossless and near-lossless compression standards. Additionally, CNN-based image-compression frameworks have been demonstrated on orbital platforms [8]. For visible band satellite imagery, adaptive lossy compression schemes provide

a much larger reduction in file size with minimal loss of reconstruction quality compared to standard JPEG codecs. Training of a CNN-based compression framework can performed at a ground station while only the inference of the network is done onboard, thus saving computational resources. The usage of this framework on existing space platforms demonstrates that deep learning-based compression techniques are viable for space flight.

3.0 Approach

This research presents an assessment of hyperspectral compression with the CNN-JPEG algorithm. This chapter documents the image preprocessing steps necessary to apply CNN-JPEG to a highly-dimensional dataset. Addtionally, this chapter describes the training process of the CNN-JPEG system and the parameters used to generate the hyperspectral compression model.

3.1 Aviris Sensor Preprocessing

In this research, we assess the performance of the CNN-JPEG adaptive compression algorithm on hyperspectral imagery. To apply the CNN-JPEG algorithm to a hyperspectral data cube, an additional layer of preprocessing must added to the compression pipeline. This additional layer deconstructs the image cube via spectral scanning; slices of the image cube are taken perpendicular to the spectral axis, forming a stack of images. This stack contains as many images as there are bands in the original data cube, and each image within the stack is a monochromatic snapshot of the scene at a particular wavelength.

Fig. 4 illustrates various methods of decomposing a hyperspectral image. This research primarily investigates compression via spectral scanning. Decomposing the original dataset using spectral scanning provides several advantages. Among these is that by transforming the image cube into a series of 2D images, we are able to apply the CNN-JPEG algorithm without drastically augmenting the size of the network to operate on hyperspectral data. Spaceborne apps are also required to run on embedded systems, which may be limited in processing capability as a result of power constraints. Thus, we wish to limit the complexity and size of the CNN to reduce resource usage and execution time. Additionally, spectral scanning provides a more direct structural representation of the scene, which can make it easier to assess the reconstruction quality through visual inspection.

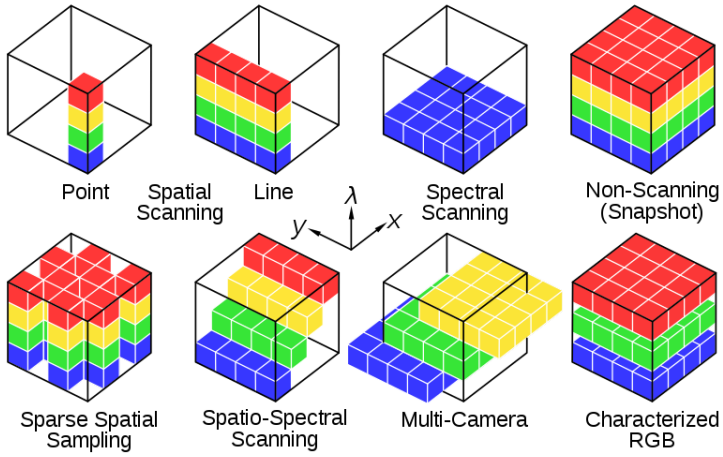

Figure 4: Hyperspectral scanning methods

After preprocessing, the CNN-JPEG algorithm was trained using component bands from four separate AVIRIS flights. With 224 bands each, these four hyperspectral data cubes provide 896 total spectral bands, with swaths ranging from approximately 750 pixels to upwards of several thousand pixels depending on flight length. The name of each flight as well as the dimensions of the hyperspectral data cube captured are shown in Table 1. Hyperspectral flights were chosen on the basis that they contain a variety of discernible features as well as have a relatively linear flight path. Linear flight path is an important consideration for training the compression model, as AVIRIS flights containing any turns or deviations are zero-padded to fill out a rectangular image. Graphically, this may result in large portions of the data cube appearing as entirely black, which have adverse effects on training the CNN-JPEG algorithm. Since a random patch extractor is utilized to augment the amount of training data available, including hyperspectral images with a nonlinear flight path would result in many patches of pure black, which is not meaningful data that can be used for training. Likewise, we also choose flight data with a variety of features to train the CNN on different environments for better generalization. Different environments provide a variety of features within the scene, which helps to avoid overfitting the network for a particular enviroment or singular image.

Table 1: AVIRIS Hyperspectral Flight Data

Flight Designation	**Image Dimensions**	**Area (km^2)**	**Size (GB)**
f110523t01p00r11	$747 \times 3525 \times 224$	665.69	1.34
f131205t01p00r12	$775 \times 6533 \times 224$	1263.9	2.54
f190821t01p00r11	$749 \times 2128 \times 224$	408.03	0.81
f080611t01p00r07	$753 \times 1924 \times 224$	418.48	0.74

Fig. 5 illustrates a composite image of the hyperspectral data cube used as the test set. In this representation, the top of the data cube is a false color representation of the scene which allows for visual identification of features within the scene. The sides of the image cube correspond to the spectral radiance along the edges of the scene. Moving down the sides of the image cube is equivalent to moving along the spectral axis λ. The spatial dimensions of the data cube are determined by the swath of the hyperspectral sensor. Also represented in this image are three spectral bands captured at wavelengths of 665 nm, 540 nm, and 472 nm. These wavelengths correspond to the visual spectrum colors red, green, and blue respectively. For the purposes of demonstration, the bands within this figure have been tinted according to their wavelength. Generally, the spectral bands extracted from the hyperspectral data cube are 2D grayscale images captured at a given wavelength, and the color value of each pixel corresponds to the spectral radiance intensity at that wavelength. The AVIRIS sensor produces 224 of these spectral bands, captured at evenly-spaced wavelengths ranging from 400 nm to 2500 nm.

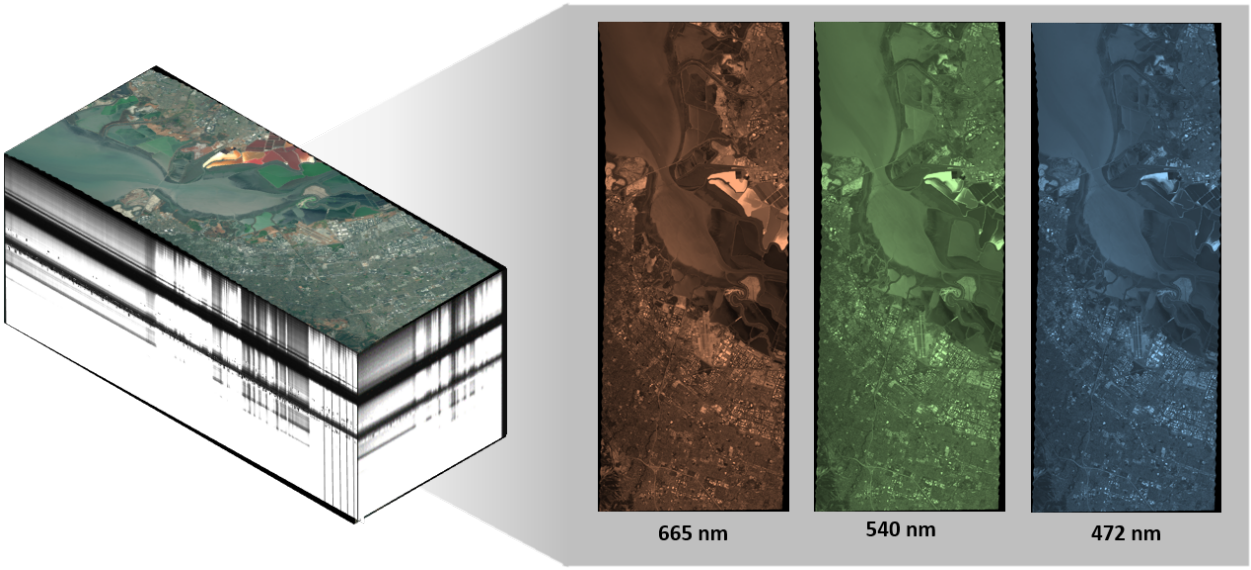

Figure 5: Composite image of a hyperspectral data cube with spectral bands

3.2 CNN-JPEG Setup

3.2.1 Training

For the purposes of training CNN-JPEG, the first three hyperspectral flights comprised the training dataset, while the fourth hyperspectral image formed the test set. Each data cube is decomposed into a series of 2D, grayscale images which can be processed faster and more easily than a dataset with large dimensionality. In this way, we also avoid a major redesign of the CNN to process 3D image cubes. In order to further augment the amount of training data available, a patch extractor is applied to the training data. The patch extractor takes 200,000 randomly selected 40×40 pixel patches of each image and applies a random rotation to avoid overfitting the network. The network was trained on this data for 50 epochs using the adam optimizer for both the encoder network and the decoder network.

3.2.2 Compression

Compression of each band is performed via inference on the test set. As discussed, the CNN-JPEG pipeline consists of a three-layer encoder network followed by a JPEG transform and a 20-layer decoder network. Compressed images are extracted after the JPEG encoding and before the decoder pipeline. These compressed images have reduced spatial dimensions compared to the original image. The size of the compressed image is a tunable, user-defined parameter; for this research, the dimensions are reduced by exactly half of the original dimensions. Passing the compressed images through the decoder pipeline then produces the reconstructed band with image dimensions equal to that of the original.

To evaluate the performance of CNN-JPEG as a technique for hyperspectral image compression, two metrics are of interest: the compression ratio and the image reconstruction quality. Compression ratio is the ratio of the size of an original image to that of the compressed image. Compression ratio provides a straightforward metric of how much reduction in file size a compression algorithm can provide. A higher compression ratio corresponds to a higher reduction in file size. Peak Signal-to-Noise Ratio (PSNR) is a commonly employed measurement of reconstruction quality for lossy image compression techniques. PSNR is the ratio of the maximum power of a signal to the power of the noise present in that signal. The PSNR of a compressed image can be calculated via mean squared error (MSE), which for an $m \times n$ image I and compressed image K is defined in Eq. 1 and Eq. 2.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{1}$$

Using the mean square error, the PSNR between the compressed image and the original can then be calculated as follows:

$$PSNR = 10 \log_{10} \left(\frac{max(I)^2}{\sqrt{MSE}} \right) \tag{2}$$

With these metrics, the performance of CNN-JPEG for hyperspectral image compression can be assessed. To compare the performance to a baseline, we also compress the image stack using a standard JPEG transform on each band.

4.0 Results

This chapter presents the results of applying the CNN-JPEG algorithm to hyperspectral imagery. Compression ratio and reconstruction quality are measured per spectral band in the test set for both CNN-JPEG compression as well as standard JPEG compression. This chapter also includes a visualization and brief analysis of each compression type.

4.1 Compression Strength

The average per-band compression ratio of the CNN-JPEG compression compared the standard JPEG is shown in Fig. 6. Overall, the CNN-JPEG compressed bands produced a compression ratio of 17.04×. Comparatively, the fully reconstructed bands provided a compression ratio of only 1.59× over the original uncompressed bands. The baseline standard JPEG compression provided a compression ratio of 3.87× on average.

Compared to the standard JPEG transform using a quality factor of 90, CNN-JPEG is able to provide a much higher average per-band compression ratio. Additionally, since a spectral scanning technique was used to decompose the image cube, the compression provided by CNN-JPEG does not take full advantage of spectral locality. Each spectral band is compressed via CNN-JPEG independently of the other compressed bands. As a result, this compression technique cannot take advantage of any similarity between the current band and surrounding bands. A full image cube snapshot approach could provide even higher compression than a spectral or spatial scanning approach.

Of note is the much larger standard deviation present in the CNN-JPEG compressed representation due to the spectral scanning method. By employing this method of decomposing the hyperspectral data cube, we obtain a set of images corresponding to a certain wavelength. In many bands, the spectral content can be relatively uniform over a wide region. As a result, many of the pixels in that band will have a similar value, which can then be more efficiently compressed by the JPEG encoder. On the other hand, in a band with

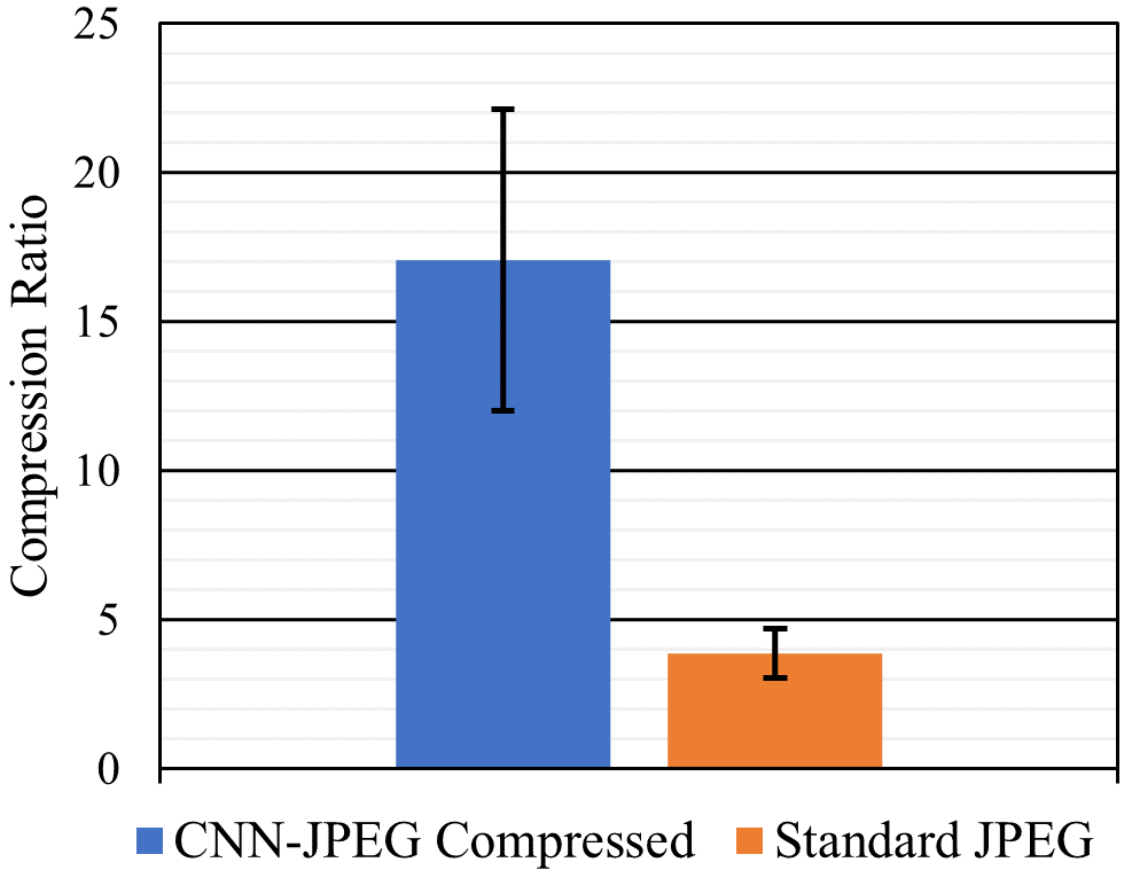

Figure 6: Average per-band compression ratio

widely varied spectral content, the JPEG encoder cannot as effectively compress the data as there is less similarity between pixel values. CNN-JPEG offsets this issue by learning a residual feature map of the data; in other words, CNN-JPEG is able learn the high-frequency features and components of an image while reducing the image dimensions. Since many of the bands in the hyperspectral image have uniform spectral content, those bands lack an abundance of well-defined features in the image, leading to the majority of the image being low-frequency content. In these images, there are fewer high-frequency components, thus limiting the amount of high-frequency data that the encoder network can reduce. As a result the effectiveness of the encoder network is restricted and the compression ratio is lowered for that band. Conversely, the encoder network is able to learn and reduce more features in bands containing a large amount of contrast and high-frequency data. For this reason, the compression ratio provided by CNN-JPEG is highly variant based on the content of each band.

Fig. 7 shows the PSNR of CNN-JPEG and the standard JPEG transform. On average, the CNN-JPEG reconstruction of each band has a lower PSNR than that of the standard JPEG by about 10 dB. The standard deviation of reconstruction quality across all bands in the image stack is consistent between both CNN-JPEG and the traditional JPEG transform. This loss in quality over the traditional JPEG transform is a result of the increased presence of visual artifacts when using CNN-JPEG. These artifacts are a result of overflow in the encoder network. Each band is quantized to an 8-bit grayscale image with the highest possible value representing pure white. As the image is passed through each layer in the encoder network, any activations in the pure white regions of an image will multiply the color value causing instabilities. Since pure white is already at the maximum possible color value, overflow occurs on activation, thereby resetting the color value to pure black. The author hypothesizes that this effect can be mitigated by adjusting the activation functions to better normalize the image.

The compression ratio of the hyperspectral data cube as whole obtained by both compression techniques is shown in Fig 8. Standard JPEG provides a compression ratio of 3.54×. The CNN-JPEG system achieves a file size reduction of 14.79× when applied across each band in the hyperspectral image, which is an improvement over standard JPEG compres-

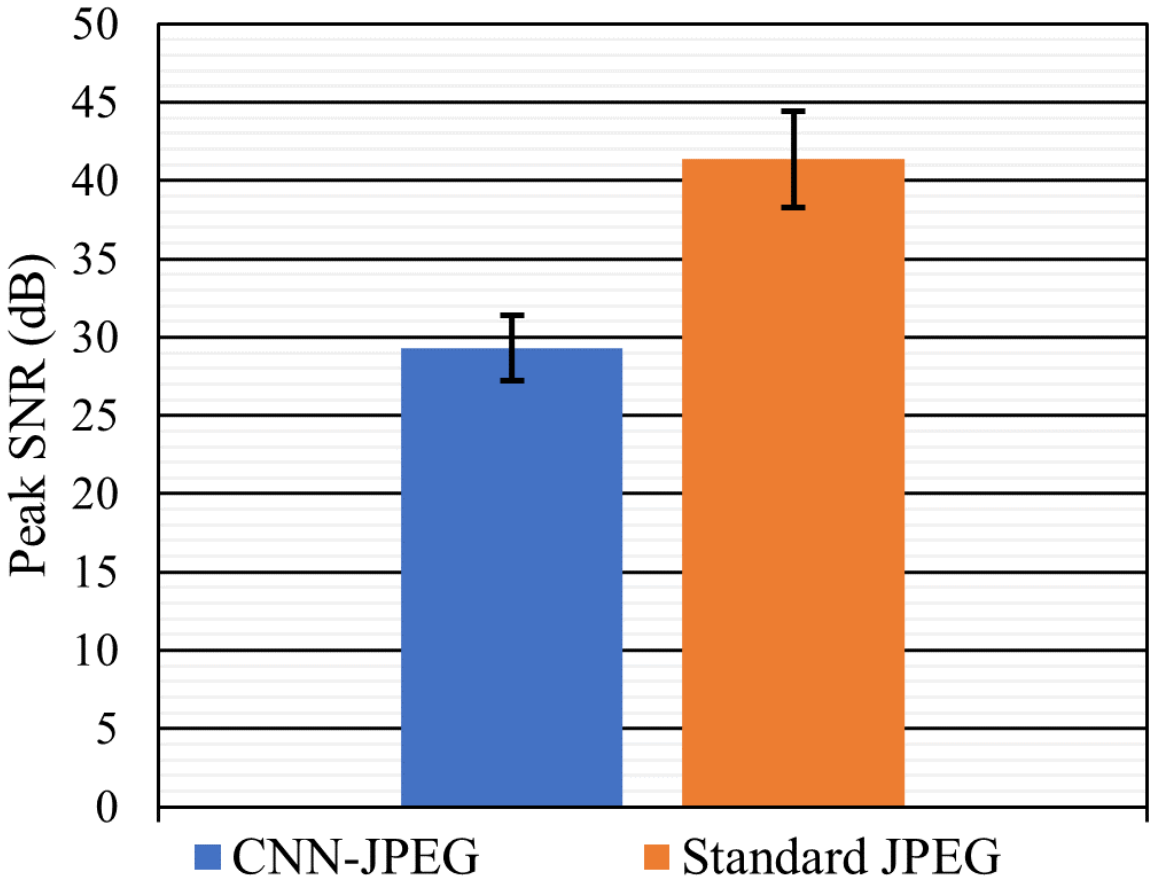

Figure 7: Comparison of average per-band PSNR

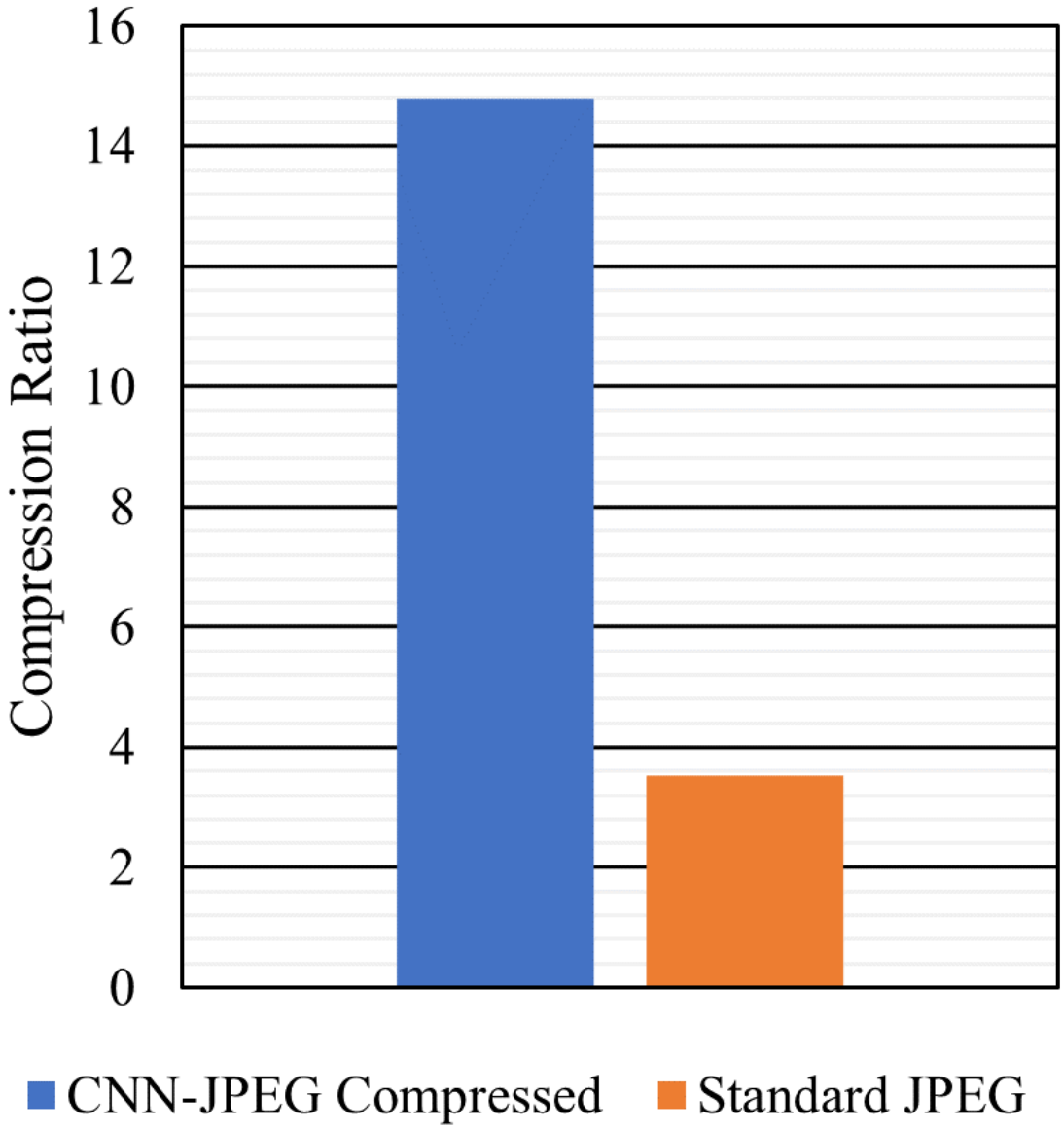

Figure 8: Full image cube compression ratio

sion by 4.18×. As such, the CNN-JPEG system offers greatly enhanced compression ratio compared to traditional image compression techniques. Additionally, the high compression ratio produced by CNN-JPEG is comparable to those of wavelet and transform based hyperspectral compression methods. The full image compression ratio can be further improved by additional tuning of the compression model parameters. Due to their increased compression power, deep learning-based compression frameworks are viable solutions for the compression of hyperspectral imagery.

4.2 Reconstruction Quality

Additionally, CNN-JPEG was unable to generalize given the high variability between bands in a hyperspectral image. The objective of CNN-JPEG is to minimize the MSE between the original and reconstructed image; the decoder network can achieve this through optimization of the residual. Bands with uniform spectral content often have a lower MSE compared to bands with more varied, high-frequency data. Since a majority of bands in a hyperspectral image tend to be low frequency content, many of the bands with more high-frequency content produced more varied gradients. This variance manifests as noise during training, and as a result CNN-JPEG was not fully optimized for the best reconstruction quality across all bands. Consequently, the average reconstruction quality of CNN-JPEG is lower than that of a standard JPEG without further optimization.

Fig. 9 provides a visualization of CNN-JPEG compression on a single band of the data cube. This band corresponds to a wavelength of 665 nm. A standard JPEG compression of the same band is also provided for comparison. Ideally, the reconstructed image should appear exactly the same as the original image with no perceptible difference between the two. On first glance, the CNN-JPEG reconstruction is visually similar to the original uncompressed image. Upon closer inspection, the CNN-JPEG reconstruction exhibits unique artifacts in the extreme values of the image. The highest and lowest values are replaced by a binary distinctive noise pattern as a result of instabilities in the network driving the pixel value either very high or very low. Given that these instabilities only occur in the brightest

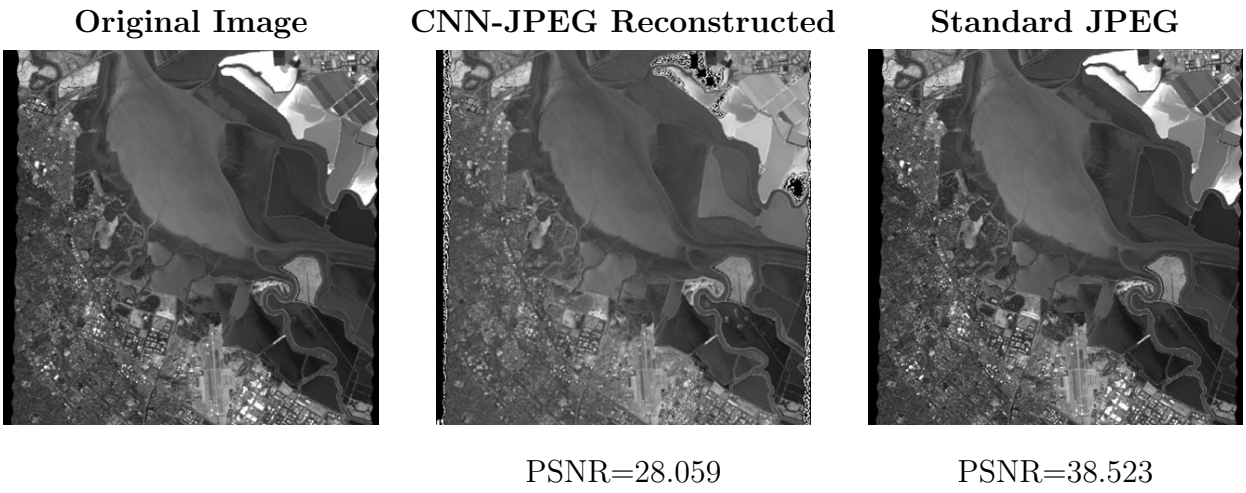

Figure 9: Visual comparison of CNN-JPEG reconstruction with original and standard JPEG compressed images.

white or darkest black regions of the reconstructed image, the reconstructed image suffers a loss in effective dynamic range over the original image. This loss of dynamic range is reflected in the PSNR of the reconstructed bands, which on average are lower than that of the standard JPEG transform. Additionally, depending on the hyperspectral application, this may have an adverse effect on accuracy, and, as a result, may be an unsatisfactory solution. Additional post processing of the image could be utilized to correct these instabilities.

5.0 Discussion

Applying CNN-JPEG to hyperspectral imagery yields increased compression ratio over a conventional JPEG transform when using a spectral scanning method. The use of a spectral scanning method offers several tradeoffs in comparison with a spatial scanning or non-scanning snapshot approach. Perhaps the most evident benefit of the spectral scanning method of decomposition is that the data can be interpreted visually with ease. Spectral scanning decomposes the hyperspectral image into a series of swaths, each at a different spectral wavelength. Since each of these swaths can be represented as a grayscale image, the scene as well as any features within it are maintained. As a result this method, the data can be represented directly. This is beneficial as it allows for quick and simple estimates of visual fidelity. Additionally, this approach does not require any modification to the size of the encoder and decoder networks. Modifying the CNN-JPEG structure to operate on highly dimensional data drastically increases the number of parameters of the CNN, which increases hardware load on embedded systems. Since this research is focused on ease of implementation of CNN-JPEG for hyperspectral imagery, the spectral scanning method is employed.

However, by decomposing the dataset in this way, CNN-JPEG is unable to exploit any spectral data locality. Since a spectral scanning method partitions the spectral axis into many individual bands, continuity between pixels along the spectral axis is broken and consecutive similar pixels cannot be efficiently compressed. Thus, the compression ratio offered by CNN-JPEG is not fully optimized. Compression ratio can be improved through modification of the encoder and decoder networks to operate on sets of bands or the full image cube. This would allow the algorithm to take full advantage of spectral locality as well as spatial locality. If the size of the encoder and decoder networks is expanded to process full hyperspectral images, similarities between bands can be more efficiently compressed.

Compression with CNN-JPEG produces high-frequency noise patterns during image reconstruction. These patterns result in reduced PSNR of the reconstructed image over the original image, and typically occur in the extreme values of the image. These noise patterns

are the result of overflow in the activation layers of the encoder. When each band passes through the activation layers, each pixel of the image is multiplied by a weight corresponding to the activation function used. In this implementation, the encoder network utilizes ReLU activation, which does not saturate for increasingly positive values. In bands that contain pure white regions with maximum color value, ReLU activation adds a bias term that is often greater than zero. The result is a color value greater than the maximum possible color value for a grayscale image. Consequently, overflow occurs and the color value of those pixels is reset to zero, representing black. As a result, bands at the output of the encoder exhibit black distortion in areas that were bright white in the original image. Overflow effects in the system can be avoided by tuning the parameters of the activation functions within the encoder network. Tuning of activation parameters can be performed by further training of the encoder network. Additional training can fine tune the weights of the activation layers, lessening the chance of overflow. Additionally, overflow effects can also be mitigated by changing the activation functions used in the encoder. Using a saturating activation function, such as a sigmoid activation, prevents positive overflow by capping the maximum value that the activation layer can output. As a result, white regions stay within the boundaries of possible color value. However, modifying the activation functions has a tradeoff in that training of the network may take longer to converge.

Fig. 10 shows a single compressed band at the output of the JPEG codec, corresponding to a wavelength of 665 nm. The brightest white areas of the original image are distorted to a pure black as a result of the instabilities in the encoder network. During reconstruction, the CNN-JPEG system adds a residual to the image to restore the high frequency components. This residual sharpens the features of the image by reintroducing high frequency content into the image. This operation increases the visual fidelity across most regions of the image. However, adding the high frequency residual produces noise-like patterns in the darkest regions of the band. These structures are visualized in Fig. 11. These artifacts may interfere with hyperspectral imaging applications depending on the specifics of that application. As a result, CNN-JPEG compression for hyperspectral should be used in applications that favor high compression ratio over image precision. Noise patterns in the reconstructed image can be reduced with further processing after decoding. Given that compression instabilities

occur in pure white areas of the original image, it is possible to predict and identify the regions containing visual noise as a result of the CNN-JPEG encoder. Thus, additional post processing can be employed to correct the instabilities and lessen the presence of the CNN-JPEG compression artifacts.

6.0 Conclusions

Space platforms are subject to limited and contested communications bandwidth. For platforms that produce large amounts of data, as is the case with hyperspectral sensors, these limitations on communications impose strict requirements on the ability to downlink data. To remedy this, we opt to make more efficient use of available bandwidth using lossy compression. By applying CNN-JPEG to large datasets such as those created by hyperspectral sensors, we can achieve compression ratios of 17× on average. This is a significant improvement in compression over standard JPEG transforms. However, this enhanced capability for compression comes at the cost of reduced image reconstruction quality, as well as the introduction of image artifacts.

Depending on the application, the increased compression offered by CNN-JPEG is greatly beneficial to first-pass apps that favor transmission throughput over precision. In apps that are highly dependent on precision, the loss in quality of CNN-JPEG is not be acceptable. For this reason, the loss in image quality must be considered an acceptable price for enhanced compression. CNN-JPEG is best employed for quick-look hyperspectral imaging apps that do not require precise quality. Future investigations of this topic include a study of the effects of CNN-JPEG compression on the performance of imaging apps and comparisons with other hyperspectral compression techniques.

Appendix Supplemental Results

Figure 10: CNN-JPEG compressed spectral band without reconstruction

Figure 11: CNN-JPEG compressed spectral band after reconstruction

Bibliography

[1] Sabine Chabrillat, Alexander F.H Goetz, Lisa Krosley, and Harold W Olsen. Use of hyperspectral images in the identification and mapping of expansive clay soils and the role of spatial resolution. *Remote Sensing of Environment*, 82(2):431–445, 2002.

[2] Chein-I Chang and Shao-Shan Chiang. Anomaly detection and classification for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 40(6):1314–1325, 2002.

[3] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto. Deep convolutional autoencoder-based lossy image compression. In *2018 Picture Coding Symposium (PCS)*, pages 253–257, 2018.

[4] M. R. Corson, D. R. Korwan, R. L. Lucke, W. A. Snyder, and C. O. Davis. The hyperspectral imager for the coastal ocean (hico) on the international space station. In *IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium*, volume 4, pages IV – 101–IV – 104, 2008.

[5] Q. Du and J. E. Fowler. Hyperspectral image compression using jpeg2000 and principal component analysis. *IEEE Geoscience and Remote Sensing Letters*, 4(2):201–205, 2007.

[6] Yaman Dua, Vinod Kumar, and Ravi Shankar Singh. Comprehensive review of hyperspectral image compression algorithms. *Optical Engineering*, 59(9):1 – 39, 2020.

[7] A. D. George and C. M. Wilson. Onboard processing with hybrid and reconfigurable computing on small satellites. *Proceedings of the IEEE*, 106(3):458–470, 2018.

[8] J. Goodwill, D. Wilson, S. Sabogal, A. D. George, and C. Wilson. Adaptively lossy image compression for onboard processing. In *2020 IEEE Aerospace Conference*, pages 1–15, 2020.

[9] G. Healey and D. Slater. Models and methods for automated material identification in hyperspectral imagery acquired under unknown illumination and atmospheric conditions. *IEEE Transactions on Geoscience and Remote Sensing*, 37(6):2706–2717, 1999.

[10] L. J. Ippolito. Radio propagation for space communications systems. *Proceedings of the IEEE*, 69(6):697–727, 1981.

[11] Matthew Klimesh. Low-complexity adaptive lossless compression of hyperspectral imagery. In Roger W. Heymann, Charles C. Wang, and Timothy J. Schmit, editors, *Satellite Data Compression, Communications, and Archiving II*, volume 6300, pages 201 – 209. International Society for Optics and Photonics, SPIE, 2006.

[12] Hien Nguyen, A. Banerjee, Philippe Burlina, Joshua Broadwater, and Rama Chellappa. *Tracking and Identification via Object Reflectance Using a Hyperspectral Video Camera*, pages 201–219. 05 2011.

[13] Mamun Bin Ibne Reaz, Masuma Akter, and Faisal Mohd-Yasin. Image compression system for mobile communication: Advancement in the recent years. *Journal of Circuits, Systems, and Computers*, 15:777–815, 10 2006.

[14] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. 05 2017.

[15] Diego Santa-Cruz, Raphaël Grosbois, and Touradj Ebrahimi. Jpeg 2000 performance evaluation and assessment. *Signal Processing: Image Communication*, 17(1):113–130, 2002. JPEG 2000.

[16] Xiaoli Tang, William A Pearlman, and James W Modestino. Hyperspectral image compression using three-dimensional wavelet coding. In *Image and Video Communications and Processing 2003*, volume 5022, pages 1037–1047. International Society for Optics and Photonics, 2003.

[17] W. Tao, F. Jiang, S. Zhang, J. Ren, W. Shi, W. Zuo, X. Guo, and D. Zhao. An end-to-end compression framework based on convolutional neural networks. In *2017 Data Compression Conference (DCC)*, pages 463–463, 2017.

[18] D. Valsesia and E. Magli. High-throughput onboard hyperspectral image compression with ground-based cnn reconstruction. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12):9544–9553, 2019.

[19] Freek D. van der Meer, Harald M.A. van der Werff, Frank J.A. van Ruitenbeek, Chris A. Hecker, Wim H. Bakker, Marleen F. Noomen, Mark van der Meijde, E. John M. Carranza, J. Boudewijn de Smeth, and Tsehaie Woldai. Multi- and hyperspectral geologic remote sensing: A review. *International Journal of Applied Earth Observation and Geoinformation*, 14(1):112–128, 2012.

[20] G. K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.